CONSTRUCTIVE APPROXIMATION AND LEARNING BY GREEDY ALGORITHMS

Dino Oglic

Bonn 2018

CONSTRUCTIVE APPROXIMATION AND LEARNING BY GREEDY ALGORITHMS

Dissertation

zur Erlangung des Doktorgrades (Dr. ret. nat.)

der Mathematisch-Naturwissenschaftlichen Fakultät

der Rheinischen Friedrich–Wilhelms–Universität Bonn

vorgelegt von

Dino Oglic

aus

Stolac, Bosnien und Herzegowina

Bonn 2018

Angefertigt mit Genehmigung
der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Thomas Gärtner
2. Gutachter: Prof. Dr. Stefan Wrobel

Tag der Promotion: 29. 06. 2018
Erscheinungsjahr: 2018

Contents

Declaration

I, Dino Oglic, confirm that this work is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at the point of their use. A full list of the references employed has been included.

Acknowledgments

I want to use this opportunity to express my gratitude for the support received from many individuals and two institutions while working on the materials for this thesis.

First and foremost, I would like to express my sincere gratitude to Prof. Thomas Gärtner who guided me through my doctoral studies. He was always available for a discussion on any research topic of interest and ready to give an advice or a novel insight into the problem. I am grateful to him for giving me the freedom to follow my own research interests and encouraging me to pursue my own ideas. All the work presented in this thesis is a result of a collaboration with Thomas who dedicated a lot of time to provide thorough and in depth comments for numerous drafts of the published manuscripts comprising the main part of the thesis. He secured financial support for my research and numerous trips to conferences. Prof. Stefan Wrobel also kindly agreed to act as the second examiner of this thesis.

Sabine Burch helped me with the administrative procedures and the visa application process shortly after Thomas offered me a research assistant position in his group at the University of Bonn. She also helped me find my first accommodation in Bonn which would have been a very difficult task given my knowledge of German at that time.

My first paper was a result of a collaboration with Daniel Paurat who also helped me a lot immediately after arriving to Bonn. I would have probably gotten myself into troubles with the tax/bank/phone and numerous other applications without his help.

I shared an office at the Fraunhofer Institute with Roman Garnett who was at the time doing a post-doc with Thomas. During this time, I started to appreciate Bayesian methods and would like to thank him for this and interesting discussions we had on Gaussian processes.

I would also like to express my gratitude to other members of the CAML group for talks and discussions at our weekly meetings: Tamas Horvath, Olana Missura, Michael Kamp, Katrin Ullrich, Pascal Welke, and Mario Boley. Michael Kamp has also invested time proof-reading some of the materials presented in this thesis and I am very grateful to him for this.

My passion for mathematics was developed during primary and high school education with the help of Prof. Hasan Jamak and Zlatko Čerkez. I want to thank them here for their time and efforts invested in developing my mathematical skills and analytical reasoning.

Most importantly, I would like to thank my family for their support, especially to *Stoico*. Despite all the fun with machine learning, you are the best part of my life.

During my doctoral studies I have worked as a research assistant at the University of Bonn and the University of Nottingham. These two positions were funded by the German Science Foundation (grant number, GA 1615/1-1) and the University of Nottingham.

Abstract

This thesis develops several kernel-based greedy algorithms for different machine learning problems and analyzes their theoretical and empirical properties. Greedy approaches have been extensively used in the past for tackling problems in combinatorial optimization where finding even a feasible solution can be a computationally hard problem (i.e., not solvable in polynomial time). A key feature of greedy algorithms is that a solution is constructed recursively from the smallest constituent parts. In each step of the constructive process a component is added to the partial solution from the previous step and, thus, the size of the optimization problem is reduced. The selected components are given by optimization problems that are simpler and easier to solve than the original problem. As such schemes are typically fast at constructing a solution they can be very effective on complex optimization problems where finding an optimal/good solution has a high computational cost. Moreover, greedy solutions are rather intuitive and the schemes themselves are simple to design and easy to implement. There is a large class of problems for which greedy schemes generate an optimal solution or a good approximation of the optimum.

In the first part of the thesis, we develop two deterministic greedy algorithms for optimization problems in which a solution is given by a set of functions mapping an instance space to the space of reals. The first of the two approaches facilitates data understanding through interactive visualization by providing means for experts to incorporate their domain knowledge into otherwise static kernel principal component analysis. This is achieved by greedily constructing embedding directions that maximize the variance at data points (unexplained by the previously constructed embedding directions) while adhering to specified domain knowledge constraints. The second deterministic greedy approach is a supervised feature construction method capable of addressing the problem of kernel choice. The goal of the approach is to construct a feature representation for which a set of linear hypotheses is of sufficient capacity—large enough to contain a satisfactory solution to the considered problem and small enough to allow good generalization from a small number of training examples. The approach mimics functional gradient descent and constructs features by fitting squared error residuals. We show that the constructive process is consistent and provide conditions under which it converges to the optimal solution.

In the second part of the thesis, we investigate two problems for which deterministic greedy schemes can fail to find an optimal solution or a good approximation of the optimum. This happens as a result of making a sequence of choices which take into account only the immediate reward without considering the consequences onto future decisions. To address this shortcoming of deterministic greedy schemes, we propose two efficient randomized greedy algorithms which are guaranteed to find effective solutions to the corresponding problems. In the first of the two approaches, we provide a mean to scale kernel methods to problems with millions of instances. An approach, frequently used in practice, for this type of problems is the Nyström method for low-rank approximation of kernel matrices. A crucial step in this method is the choice of landmarks which determine the quality of the approximation. We tackle this problem with a randomized greedy algorithm based on the K-means++ cluster seeding scheme and provide a theoretical and empirical study of its effectiveness. In the second problem for which a deterministic strategy can fail to find a good solution, the goal is to find a set of objects from a structured space that are likely to exhibit an unknown target property. This discrete optimization problem is of significant interest to cyclic discovery processes such as de novo drug design. We propose to address it with an adaptive Metropolis–Hastings approach that samples candidates from the posterior distribution of structures conditioned on them having the target property. The proposed constructive scheme defines a consistent random process and our empirical evaluation demonstrates its effectiveness across several different application domains.

CHAPTER 1

Introduction

Machine learning is the study of methods for programming computers to learn (Dietterich, 2003). Over the past two decades it has become a core building block of intelligent systems capable of learning from experience and adapting to their environment. On a daily basis, machine learning algorithms provide search results, recommendations about movies and shopping items, traffic predictions, optimal navigation routes, automatic language translations, and similar services to hundreds of millions of people around the world (Dietterich and Horvitz, 2015). The development of these systems was accompanied by the technological advancement and increase in the computational power and storage capacities of computing devices. As a result of the latter, the amount of data available for analysis has increased significantly and the current trend indicates that this will continue in the years to come. It is, thus, reasonable to expect that data analysis and machine learning will become one of the driving forces of the technological progress and advancement in many fields of science (Dietterich and Horvitz, 2015; Smola and Vishwanathan, 2010).

A central problem in machine learning is that of estimating dependences and extracting law-like relationships from empirical data (Vapnik, 1982). In the past two decades, a class of theoretically well founded machine learning algorithms known as kernel methods has tackled this problem with success across many different application areas (Schölkopf and Smola, 2002). Following this line of research, the thesis develops several kernel-based greedy algorithms for different machine learning problems and analyzes their theoretical and empirical properties. A characteristic common to all investigated problems is that a solution is given by a set of atoms represented as functions or instances. We propose to tackle these problems with deterministic and/or randomized greedy algorithms. In particular, the first part of the thesis develops two deterministic greedy algorithms for optimization problems in which a solution is given by a set of functions mapping an instance space to the space of reals. The second part of the thesis, on the other hand, focuses on two discrete optimization problems in which a deterministic greedy strategy can fail to find an optimal solution or a satisfactory approximation of the optimum. Common to these two discrete optimization problems is that a solution is given by a set of instances with an a priori specified cardinality.

The first of the two deterministic greedy approaches facilitates data understanding through interactive visualization by providing means for experts to incorporate their domain knowledge into otherwise static kernel principal component analysis (Schölkopf et al., 1999).

This is achieved by greedily constructing embedding directions that maximize the variance at data points (unexplained by the previously constructed embedding directions) while adhering to specified domain knowledge constraints. The second deterministic greedy approach is a supervised feature construction method capable of addressing the problem of kernel choice. The goal of the approach is to construct a feature representation for which a set of linear hypotheses is of sufficient capacity—large enough to contain a satisfactory solution to the considered problem and small enough to allow good generalization from a small number of training examples. The approach mimics functional gradient descent and constructs features by fitting squared error residuals. We show that the constructive process is consistent and provide conditions under which it converges to the optimal solution.

In the first of the two randomized greedy approaches, we provide a mean to scale kernel methods to problems with millions of instances. An approach, frequently used in practice, for this type of problems is the Nyström method (Williams and Seeger, 2001) for low-rank approximation of kernel matrices. A crucial step in this method is the choice of landmarks which determine the quality of the approximation. We tackle this discrete optimization problem with a randomized greedy approach based on the K-means++ cluster seeding scheme (Arthur and Vassilvitskii, 2007) and provide a theoretical and empirical study of its effectiveness. In the second problem for which a deterministic strategy can fail to find a good solution, the goal is to find a set of objects from a structured space that are likely to exhibit an unknown target property. This discrete optimization problem is of significant interest to cyclic discovery processes such as *de novo* drug design (Schneider and Fechner, 2005). We propose to address it with an adaptive Metropolis–Hastings approach that samples candidates from the posterior distribution of structures conditioned on them having the target property. The proposed scheme defines a consistent random process and our empirical evidence demonstrates its effectiveness across several different application domains.

As the developed approaches are designed for different learning tasks, in the remainder of the chapter we provide a high-level overview of supervised, semi-supervised, and transductive learning. Precise mathematical definitions of all the required technical terms will be provided in the main part of the thesis, so that the chapters with developed approaches can be read independently. Having reviewed relevant learning tasks, we give an informal introduction to kernel methods that are at the core of the developed greedy approaches. Following this, we outline a greedy approach to machine learning and provide a high-level description of components constituting a greedy algorithm. This chapter concludes with a summary of the contributions and an overview of the remaining chapters.

1.1 Learning Tasks

In this section, we provide a brief review of machine learning tasks investigated in this thesis. Formal and more precise descriptions will be given in chapters that apply to these tasks.

1.1.1 Supervised Learning

Let us begin by introducing supervised learning through the simplest machine learning task – classification of objects. Assume we are required to write a program that is capable of determining whether an email belongs to a *spam* or *non-spam* class of emails. The main problem in this task is to develop an algorithm that assigns a correct class label (i.e., spam or non-spam) to an object/instance (i.e., email) based on a set of already classified training examples. In this particular case, a training example is a pair consisting of an email and its associated class label (e.g., 1 for spam and 0 for non-spam emails). The machine learning task in which an algorithm is required to extract a dependency/rule from given labeled training data is called *supervised learning*. As the algorithm is required to work for any possible email, it needs to generalize well and provide accurate predictions on unseen instances. The generalization ability and quality of an algorithm in supervised learning is evaluated by applying the learned predictor to test examples, unavailable to the algorithm during training.

In the presented example, supervised learning was introduced with a task where the labels are constrained to binary numbers. The term supervised learning, however, is not limited to such tasks and refers also to tasks with categorical, real-valued, and structured labels (e.g., graphs). A supervised learning problem with objects/instances associated to real-valued labels is known as *regression*.

1.1.2 Semi-Supervised and Transductive Learning

In *semi-supervised* and *transductive learning* tasks, in addition to having labeled training data, a program required to extract a dependence from data has also a set of unlabeled instances at its disposal. Often, it is the case that labels are expensive to obtain and there is only a small number of labeled instances together with a large number of unlabeled ones. While in semi-supervised tasks a learning algorithm is required to generalize to all possible instances, in transductive tasks an algorithm is only required to output a prediction on instances available to it during training (i.e., labeled and unlabeled instances provided as input to the algorithm). In these two machine learning tasks, the additional unlabeled data can help with learning if it contains information useful for the inference of an unknown target dependence being inferred from training examples. Precise assumptions often imposed on unlabeled data will be given in chapters specific to these tasks.

Beside the described standard forms of supervision that involve labeled instances, there are cases in which only partial supervision is provided to semi-supervised and transductive algorithms. An example of partial supervision is a pairwise constraint that specifies whether a pair of instances belongs to the same class or not. A pair of instances from the same class is usually referred to as a must-link constraint and, otherwise, a cannot-link constraint.

1.2 Kernel Methods

In all the reviewed learning tasks, machine learning algorithms are required to generalize from training examples to unseen instances. In order to achieve this, beside information in labeled training data algorithms need a notion of similarity on the available instances. Often, this information is provided to learning algorithms through a similarity function that quantifies the relationship between any two instances from the instance space. In this way, similarity functions add an additional structure to learning algorithms that is required for generalization. The choice of a similarity function determines the generalization properties of learning algorithms and is a core question in machine learning.

A *kernel function* is a special type of similarity function that is symmetric and positive definite. It is defined by a mapping that embeds instances into an inner product space. For any two instances, the kernel function computes their similarity as the inner product between their mappings. This type of similarity is beneficial for the study of learning algorithms because it allows the study of convergence properties using techniques from analysis and understanding of the algorithms on an intuitive level using analytical geometry.

Kernel methods represent a class of learning algorithms in which the relation between instances is defined using kernel functions. In learning algorithms from this class, for a given kernel function an optimal predictor is often given as a solution to a convex optimization problem. These convex optimization problems usually consist of two terms: *i*) a cost function that takes into account the deviations between predictions and labels at training instances, and *ii*) a regularization term that takes into account the complexity of inferred functional dependence and promotes the choice of smooth hypotheses. The mathematical properties of the space defined by a kernel function allow one to express an optimal prediction at any particular instance with a linear combination of kernel function values between that and all training instances. As a result of this, the optimization problems in kernel methods can be expressed solely in terms of kernel values between pairs of instances. In literature, the latter is known as the *kernel trick*. An important consequence of the kernel trick is the fact that explicit non-linear maps that define kernel functions do not need to be stored and highly expressive kernel functions corresponding to infinite dimensional inner product spaces can be seamlessly used with developed kernel methods.

The most popular approaches from this class of learning algorithms are support vector machines (Vapnik, 1982; Schölkopf and Smola, 2002), kernel ridge regression (Schölkopf and Smola, 2002), and kernel principal component analysis (Schölkopf et al., 1999). The first two approaches were initially developed for supervised learning tasks, and later extended to semi-supervised and transductive learning (e.g., see Chapelle et al., 2006). Kernel principal component analysis is a non-linear dimensionality reduction technique that projects instances to a lower-dimensional space while retaining as much as possible of the variation present in the dataset. As a result of the kernel trick, the inputs to these algorithms consist of a kernel matrix with entries corresponding to kernel values between pairs of instances and (for tasks with supervision) labels assigned to a subset of the available instances. Thus, these algorithms are highly flexible and for any particular algorithm an identical implementation can be used in combination with different kernel functions.

1.3 Greedy Algorithms

The focus of this thesis is on learning problems in which a solution is given by a set of atoms which can be represented as functions or instances. Greedy approaches have been extensively used in the past for tackling a large class of related problems in combinatorial optimization where atoms are elements of a discrete space (e.g., see Chapter 16 in Cormen et al., 2009). A deterministic greedy algorithm constructs a solution to a problem by making a sequence of choices which take into account only the immediate reward without considering the consequences onto future decisions. Thus, at each decision point of this constructive process a deterministic greedy algorithm adds an atom (i.e., a smallest constituent part/component) to the partial solution derived at the previous decision point. The selected atom at a decision point is given by an optimization problem that is computationally simpler and easier to solve than the original optimization problem. There are many problems for which deterministic greedy strategies generate an optimal solution or a good approximation of the optimum (Cormen et al., 2009). Such algorithms are especially effective in situations where an estimate of an optimal solution is needed quickly. Typically, it is not an easy task to show that a greedy algorithm constructs an optimal solution to an optimization problem. However, there are two properties of the optimization problems which can aid in designing deterministic greedy algorithms (Cormen et al., 2009): *i*) a greedy choice property which ensures that a globally optimal solution can be derived by making a locally optimal (i.e., greedy) choice, and *ii*) an optimal substructure property which establishes that an optimal solution to a problem contains within it optimal solutions to subproblems (i.e., subsets of a set of atoms comprising an optimal solution are optimal solutions to the corresponding subproblems). While deterministic greedy strategies are effective and efficient for a large class of optimization problems there are many problems in which such strategies can fail to generate an optimal solution or a good approximation of the optimum. This typically happens as a result of considering only the immediate benefit at each decision point during the construction of the greedy solution. To address this shortcoming of deterministic greedy strategies, we investigate the effects of randomization at decision points of such a constructive process. More specifically, we focus on strategies in which an atom is added to a partial solution by sampling proportional to a suitable measure of improvement over that solution. We leave the specifics of the developed approaches to their respective chapters, and provide here a high-level overview of the main components characteristic to a greedy algorithm.

A greedy algorithm can be characterized by the following four components:

i) A *set of atoms* defining the smallest constituent parts of a solution. In Chapter 2, the atoms are functions from a reproducing kernel Hilbert space and a solution (i.e., a data visualization) is a set of such functions. In Chapter 3, the atoms are ridge-wave functions that comprise a dictionary of features and a solution for the problem investigated in this chapter is a set of such features. In Chapter 4, the atoms are instances provided as input to the algorithm and a solution is a subset of the instances with an a priori specified cardinality. In Chapter 5, the atoms are elements of an instance space that can be accessed via a sampler and a solution is a set of instances with desired properties.

ii) A *solution space* is a set of all feasible solutions to a considered problem. For the greedy algorithms considered in this thesis, the respective solution spaces consist of data visualizations (Chapter 2), feature representations (Chapter 3), subsets of the available instances with a fixed cardinality (Chapter 4), and sets of structured objects with an a priori specified cardinality (Chapter 5).

iii) A *selection function* that chooses the next atom to be added to the current partial solution. For deterministic greedy approaches (Chapters 2 and 3), the selection function always chooses an atom that offers the most rewarding immediate benefit. For randomized greedy approaches, investigated in Chapters 4 and 5, the selection function is randomized based on a theoretical consideration of the respective problems.

iv) An *evaluation function* capable of evaluating the quality of intermediate/partial and complete solutions with respect to the optimization objective of a considered problem.

Having described what characterizes a greedy algorithm, we proceed to the next section where we give an outline of the thesis that summarizes our main contributions.

1.4 Outline of the Thesis

We provide here an outline of this thesis consisting of two main parts: *i*) deterministic greedy approaches (Chapters 2 and 3), and *ii*) randomized greedy approaches (Chapters 4 and 5). The outline is given by covering the investigated problems and our contributions by chapters.

Chapter 2

Data understanding is an iterative process in which domain experts combine their knowledge with the data at hand to explore and confirm hypotheses about the data. One important set of tools for exploring hypotheses about data are visualizations. Typically, traditional unsupervised dimensionality reduction algorithms are used to generate visualizations. These tools allow for interaction, i.e., exploring different visualizations, only by means of manipulating some technical parameters of the algorithm. Therefore, instead of being able to intuitively interact with visualizations, domain experts have to learn and argue about these technical parameters. To address this shortcoming of unsupervised algorithms for data visualization, we propose a greedy approach that enables experts to incorporate their domain knowledge into the otherwise static kernel principal component analysis algorithm. The developed approach is not limited to data lying in a Euclidean space (e.g., it can generate visualizations of data represented with vectors, graphs, strings etc.) and allows for an intuitive interaction with data visualizations. More specifically, the approach allows domain experts to explore hypotheses and discover structure in datasets by: *i*) selecting a small number of control points and moving them across the projection space, *ii*) specifying whether pairs of points should be placed close together or far apart, and *iii*) providing class labels for a small number of data points. Each embedding direction in a visualization generated by the proposed approach can be expressed as a non-convex quadratic optimization problem over a hypersphere of constant radius. A globally optimal solution for this problem can be found in polynomial time using the algorithm presented in this chapter. To facilitate direct feedback from domain experts, i.e., updating the whole embedding with a sufficiently high frame-rate during interaction, we reduce the computational complexity further by incremental up- and down-dating. Our empirical evaluation demonstrates the flexibility and utility of the approach.

Chapter 3

A key aspect of kernel methods is the choice of kernel function that defines a notion of similarity between instances. This choice is crucial for the effectiveness of kernel-based learning algorithms and it is important to select a kernel function such that it expresses the properties of data relevant to a dependence being learned. As such properties of the data are not provided as input to a learning algorithm, it needs to learn to select a good kernel for the problem at hand. To address this shortcoming of kernel methods, we develop an effective method for supervised feature construction. The main goal of the approach is to construct a feature representation for which a set of linear hypotheses is of sufficient capacity—large enough to contain a satisfactory solution to the considered problem and small enough to allow good generalization from a small number of training examples. We provide conditions under which this goal can be achieved with a greedy procedure that constructs features by fitting squared error residuals. The proposed constructive process is consistent and can output a rich set of features. More specifically, it allows for learning a feature representation that corresponds to an approximation of a positive definite kernel from the class of stationary kernels. The effectiveness of the approach is evaluated empirically by fitting a linear ridge regression model in the constructed feature space and our empirical results indicate a superior performance of the proposed approach over the competing methods.

Chapter 4

In this chapter, we focus on the problem of scaling kernel methods to datasets with millions of instances. An approach, frequently used in practice, for this type of problems is the Nyström method for low-rank approximation of kernel matrices. A crucial step in this method is the choice of landmarks. This is a difficult combinatorial problem that directly determines the quality of the approximation. To address this problem we propose to use a randomized greedy sampling scheme developed for the seeding of K-means++ clusters. Our main contribution is the theoretical and empirical study of the effectiveness of landmarks generated with this sampling scheme. Previous empirical studies (Zhang et al., 2008; Kumar et al., 2012) also observe that the landmarks obtained using (kernel) K-means clustering define a good low-rank approximation of kernel matrices. However, the existing work does not provide a theoretical guarantee on the approximation error for this approach to landmark selection. We close this gap and provide the first bound on the approximation error of the Nyström method with kernel K-means++ samples as landmarks. Moreover, for the frequently used Gaussian kernel we provide a theoretically sound motivation for doing the Lloyd refinements of kernel K-means++ samples in the instance space. We substantiate our theoretical results empirically by comparing the approach to several state-of-the-art landmark sampling algorithms.

Chapter 5

In this chapter, we consider an active classification problem in a structured space with cardinality at least exponential in the size of its combinatorial objects. The ultimate goal in this setting is to discover structures from structurally different partitions of a fixed but unknown target class. An example of such a process is that of computer-aided *de novo* drug design. In the past 20 years several Monte Carlo search heuristics have been developed for this process. Motivated by these hand-crafted search heuristics, we devise a Metropolis–Hastings sampling scheme that samples candidates from the posterior distribution of structures conditioned on them having the target property. The Metropolis–Hastings acceptance probability for this sampling scheme is given by a probabilistic surrogate of the target property, modeled with a max-entropy conditional model. The surrogate model is updated in each iteration upon the evaluation of a selected structure. The proposed approach is consistent and our empirical results indicate that it achieves a large structural variety of discovered targets.

1.5 Bibliographical Notes

This thesis builds on the following published articles:

(Oglic et al., 2018) Dino Oglic, Steven A. Oatley, Simon J. F. Macdonald, Thomas Mcinally, Roman Garnett, Jonathan D. Hirst, Thomas Gärtner, 2018. Active Search for Computer-Aided Drug Design. *In*: Molecular Informatics, 37(1-2):1700130.

(Oglic and Gärtner, 2017) Dino Oglic and Thomas Gärtner, 2017. Nyström Method with Kernel K-means++ Samples as Landmarks. *In*: Proceedings of the Thirty-Fourth International Conference on Machine Learning (ICML 2017), volume 70 of Proceedings of Machine Learning Research, pages 2652–2660.

(Oglic et al., 2017) Dino Oglic, Roman Garnett, and Thomas Gärtner, 2017. Active Search in Intensionally Specified Structured Spaces. *In*: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017), pages 2443–2449.

(Oglic and Gärtner, 2016) Dino Oglic and Thomas Gärtner, 2016. Greedy Feature Construction. *In*: Advances in Neural Information Processing Systems 29 (NIPS 2016), pages 3945–3953.

(Oglic et al., 2014b) Dino Oglic, Daniel Paurat, and Thomas Gärtner, 2014. Interactive Knowledge-Based Kernel PCA. *In*: Machine Learning and Knowledge Discovery in Databases: European Conference (ECML PKDD 2014), pages 501–516.

(Oglic et al., 2014a) Dino Oglic, Roman Garnett, and Thomas Gärtner, 2014. Learning to Construct Novel Structures. *In*: NIPS Workshop on Discrete and Combinatorial Problems in Machine Learning (DISCML 2014).

(Paurat et al., 2013b) Daniel Paurat, Dino Oglic, and Thomas Gärtner, 2013. Supervised PCA for Interactive Data Analysis. *In*: 2nd NIPS Workshop on Spectral Learning.

Part I

Deterministic Greedy Approaches

CHAPTER 2

Knowledge-Based Kernel Principal Component Analysis

Data visualization is an important part of knowledge discovery and at the core of data understanding and exploration tasks (e.g., see Shearer, 2000). Its importance for data science has been recognized already by Tukey (1974). While knowledge discovery is inherently interactive and iterative, most data visualization methods are inherently static. Switching algorithms and changing their parameters allow for some interaction with the visualization but this interaction is rather indirect and only feasible for machine learning experts rather than domain experts. As data science and its tools are getting more and more widespread, the need arises for methods that allow domain experts to directly interact with data visualizations through intuitive domain-knowledge constraints. Motivated by this, we propose two variants of kernel principal component analysis that allow domain experts to directly interact with data visualizations and to add domain-knowledge and other constraints in an intuitive way. The proposed approach represents an extension of kernel principal component analysis to semi-supervised tasks and it can be, with a suitable choice of kernel, used for interactive visualization of data represented as graphs, strings, vectors, time-series etc. Similar to kernel principal component analysis, a projection/embedding direction (i.e., knowledge-based kernel principal component) corresponds to a function in the underlying reproducing kernel Hilbert space. For each such projection direction, we propose to find a function which (i) either maximizes the variance at data instances while having constant norm or minimizes the norm while having constant variance at data instances, (ii) is as orthogonal as possible to previously generated embedding directions, and (iii) adheres to the knowledge-based constraints as much as possible. The first two requirements are common for kernel principal component analysis and the third one ensures that an embedding direction accounts not only for the variation at data instances but also for the relationship to their labels. Our knowledge-based principal components can take into account a variety of hard and/or soft constraints, allowing flexible placement of control points in an embedding space, addition of must-link and cannot-link constraints, as well as known class labels. The goal of the proposed approach is to allow domain experts to interact with a low-dimensional embedding and choose from the many possible ones not by tuning parameters but by dragging or grouping the chosen data points in the embedding, whereby all related data points automatically and smoothly adjust their positions accordingly. As it is unrealistic to expect that domain experts will provide supervision for all available data points (possibly millions of unlabeled instances), the problem setting of our algorithms is that of semi-supervised learning. Thus, the effectiveness of an approach depends on its ability to incorporate standard semi-supervised learning assumptions and given knowledge-based constrains into the search for *good* projection directions.

We start our presentation with a brief review of kernel principal component analysis (Section 2.1) and semi-supervised learning (Section 2.2). Following this, we devise means to incorporate domain-knowledge into kernel principal component analysis and propose two knowledge-based variants of that approach (Section 2.3). Having expressed knowledge-based kernel principal components as optimization problems over a reproducing kernel Hilbert space of functions (Section 2.3.1), we show that the representer theorem (Wahba, 1990) applies to these problems (Section 2.3.2). Following this, we apply the representer theorem and transform them to the optimization of an indefinite quadratic form over a hypersphere of constant radius, and subject to an optional linear equality constraint (Section 2.3.3). The optimization of a quadratic form over a hypersphere of constant radius is, in general, a non-convex optimization problem with potentially exponentially many local optima. We analyze this problem in Section 2.4 and show that a globally optimal solution can be found in time cubic in the size of the kernel expansion. The algorithms for solving this non-convex problem rely on the work by Forsythe and Golub (1965) and Gander et al. (1989), who were the first to consider the optimization of a quadratic form over a hypersphere. In particular, Gander et al. (1989) generally suggested two approaches for solving this problem: *i*) transforming it to a quadratic and then linear eigenvalue problem or *ii*) reducing it to solving a one-dimensional secular equation. While the first approach is more elegant, the second one is numerically much more stable. Both approaches have computational complexity that is cubic in the number of instances. The runtime complexity of these algorithms prohibits us from scaling knowledge-based kernel principal component analysis to datasets with millions of instances. In order to address this shortcoming, we propose two methods for the approximation of knowledge-based kernel principal components (Section 2.5). In the first approach, we observe that the minimization of a quadratic form over a hypersphere of constant radius is equivalent to solving a linear system defined with a symmetric and positive definite matrix (Section 2.5.1). For that problem, an iterative solution is possible using the conjugate gradient descent method (e.g., see Chapter 10 in Golub and van Loan, 1996). The approach computes an approximation to an optimal solution by iteratively improving over the existing estimate of the solution and the cost of such an iteration is quadratic in the number of instances. The iterative approach is presented in Section 2.5.1 and it is guaranteed to find a good approximation to an optimal solution with a small number of iterations in cases where the matrix defining the quadratic form is well-conditioned or has a low-rank. In the second approach for scaling knowledge-based kernel principal component analysis, we propose to use a low-rank factorization of the kernel matrix in the place of the original matrix (Section 2.5.2). For example, such a factorization can be obtained using the Nyström method for low-rank approximation of kernel matrices (Nyström, 1930; Williams and Seeger, 2001). Alternatively, if the kernel function is from the class of stationary kernels, it is possible to approximate the kernel matrix with a low-rank factorization defined by the corresponding random Fourier features (Rahimi and Recht, 2008a). These techniques are described in Section 2.5.2, together with transformations of the corresponding optimization problems that allow us to efficiently compute the low-rank approximations of knowledge-based kernel principal components.

In order to allow a direct interaction with the embedding, i.e., updating the whole embedding with a sufficiently high frame-rate, the cubic complexity of the presented solvers (Forsythe and Golub, 1965; Gander et al., 1989) is not sufficient. To overcome this, we observe that in an interactive setup it is hardly ever the case that the optimization problem has to be solved from scratch. Instead, consecutive optimization problems will be strongly related and indeed we show (in Section 2.6) that consecutive solutions differ only in rank-one updates which allows for much more fluent and natural interaction. However, even quadratic computational complexity of such interactions is not sufficient for large scale datasets with millions of instances. To efficiently interact with visualizations of such datasets, we propose to combine low-rank approximations of kernel matrices with rank-one updates of the consecutive problems.

To generate an informative visualization using the proposed approach a number of hyperparameters needs to be fine-tuned. These parameters can, for instance, specify the confidence in specific type of knowledge-based constraints or provide additional flexibility when it comes to the choice of kernel function. In Section 2.7, we consider the problem of selecting a good set of hyperparameters automatically without putting any effort on domain experts. We achieve this goal by deriving closed form hyperparameter gradients for different validation objective functions. The derived gradients can then be used with minimization procedures such as the limited memory Broyden–Fletcher–Goldfarb–Shanno method (L-BFGS-B) to determine a good set of hyperparameters. The hyperparameter optimization is, in general, posed as a non-convex problem and does not yield a globally optimal solution.

Having presented our approach, we provide a discussion of the alternative approaches to interactive data visualization (Section 2.8) and present the results of our empirical study (Section 2.9). The main focus of the study is on demonstrating the flexibility and usability of the proposed approach in an interactive knowledge discovery setting. To achieve this, we first show that small perturbations of the location of control points only lead to small perturbations of the embedding of all points. This directly implies that it is possible to smoothly change the embedding without sudden and unexpected jumps (large changes) of the visualization. We then show that by appropriate placement of control points, knowledge-based kernel principal component analysis can mimic other embeddings. In particular, we consider the sum of two different kernels and observe that by placing a few control points, the 2D kernel principal component analysis embedding of either of the original kernels can be reasonably well recovered. In addition, we investigate the amount of information retained in low-dimensional embeddings. More specifically, we take the benchmark datasets for semi-supervised classification prepared by Chapelle et al. (2006) and assess the predictive performance of the first knowledge-based kernel principal component in relation to other approaches. Last but not least, we show that it is possible to discover structures within the dataset that do not necessarily exhibit the highest correlation with variance and, thus, remain hidden in the plain kernel principal component analysis embedding.

2.1 Kernel Principal Component Analysis

Dimensionality reduction is the process of reducing the dimensionality of a dataset consisting of a large number of possibly interrelated variables, while retaining as much as possible of the information present in the dataset (Jolliffe, 1986). Principal component analysis (PCA) is a linear dimensionality reduction technique that transforms a dataset to a new low-dimensional representation with mutually independent features, while retaining most of the variation present in the dataset. The earliest formulations of the technique can be traced back to Pearson (1901) and Hotelling (1933). The name itself originates from the work of Hotelling (1933) in which he considered the problem of finding mutually uncorrelated features that determine the input data representation. Motivated by the ongoing research in psychology, Hotelling (1933) called such uncorrelated features *components*. The components are called *principal* because they are obtained by successively trying to retain as much as possible of the previously unexplained variation present in the dataset. In addition to dimensionality reduction, this technique can also be used for exploratory analysis or visualization of datasets (Jolliffe, 1986; Jeong et al., 2009). For example, a data visualization can be generated by projecting instances to a space spanned by the first two principal components. Having generated a data visualization, domain experts can then use it to gain different insights into properties of the dataset such as the difficulty of a learning task, separation of instances into clusters and/or classes, or detection of outliers. Kernel principal component analysis (Schölkopf et al., 1999) is an extension of principal component analysis in which the relation between input features and principal components is no longer linear. In contrast to principal component analysis, this kernel method is also not restricted to Euclidean spaces and can be used with data represented as graphs, strings, time-series, relations etc. As kernel principal component analysis is at the core of our approach for interactive data visualization (described in Section 2.3), we provide a brief review of it in the remainder of this section.

2.1.1 Definition of Kernel Principal Components

This section reviews two principles for obtaining (kernel) principal components. The first principle originates from the work by Hotelling (1933) and the second one is due to Pearson (1901). For kernel principal component analysis with a centered kernel function these two principles are equivalent. In the remainder of this section, we review these two principles and discuss their similarities and differences. We start our review by introducing the relevant terms that will be used throughout this chapter. Following this, we formulate two optimization problems which can be used to derive the first principal component in a reproducing kernel Hilbert space of functions mapping an instance space to the space of reals. The section concludes with a scheme for the iterative computation of the remaining principal components.

Let us assume that n instances $\{x_1, x_2, \ldots, x_n\}$ are sampled from a Borel probability measure defined on an instance space $\mathcal{X}$ (not necessarily Euclidean). Let $r \ll d$ denote the dimensionality of the transformed dataset and $\mathcal{H}$ a reproducing kernel Hilbert space with kernel $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Without loss of generality, we can assume that the columns of the data matrix $X \in \mathbb{R}^{d \times n}$ are centered instances (i.e., $\sum_{i=1}^{n} x_i = 0$). While in principal component analysis a component f is a linear combination of input features, i.e., $f = \alpha^\top x$ and $\alpha \in \mathbb{R}^d$, in kernel principal component analysis it is an element of the Hilbert space of functions defined on $\mathcal{X}$. As we will see in the next section, the representer theorem (Wahba, 1990; Schölkopf et al., 2001; Dinuzzo and Schölkopf, 2012) allows us to write a kernel principal component as $f = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$, where $\alpha_i \in \mathbb{R}$ and $i = 1, 2, \ldots, n$. We now describe how kernel principal components can be computed from a set of instances.

On the one hand, following the approach by Hotelling (1933), the first principal component can be obtained by maximizing the component variance at the given instances. As collinear components are equivalent, principal components can be without loss of generality restricted to functions having unit norm in $\mathcal{H}$. Thus, the first principal component is given as a solution of the following optimization problem (Hotelling, 1933)

$$f_1 = \underset{\|f\|_{\mathcal{H}}=1}{\operatorname{argmax}} \sum_{i=1}^{n} \left(f(x_i) - \frac{1}{n} \sum_{j=1}^{n} f\left(x_j\right) \right)^2 . \tag{2.1}$$

On the other hand, building on the work of Pearson (1901), the first principal component can be derived by finding a functional direction that minimizes the squared residual errors between images of instances in $\mathcal{H}$ and their projections over that functional direction. More formally, the first principal component can be computed as (Pearson, 1901)

$$f_1 = \underset{\|f\|_{\mathcal{H}}=1}{\operatorname{argmin}} \sum_{i=1}^{n} \|k(x_i, \cdot) - \langle k(x_i, \cdot), f\rangle f\|_{\mathcal{H}}^2 . \tag{2.2}$$

The optimization objective from problem (2.2) can be simplified using the reproducing property of the kernel, i.e.,

$$\begin{aligned}\sum_{i=1}^{n} \|k(x_i, \cdot) - \langle k(x_i, \cdot), f\rangle f\|_{\mathcal{H}}^2 = & \sum_{i=1}^{n} k(x_i, x_i) - 2f(x_i)^2 + f(x_i)^2 \|f\|_{\mathcal{H}}^2 = \\ & \sum_{i=1}^{n} k(x_i, x_i) - f(x_i)^2 .\end{aligned}$$

Eliminating the constant terms from the transformed objective, the optimization problem from Eq. (2.2) becomes

$$f_1 = \underset{\|f\|_{\mathcal{H}}=1}{\operatorname{argmax}} \sum_{i=1}^{n} f(x_i)^2 . \tag{2.3}$$

For centered kernel functions, the optimization problems from Eq. (2.1) and (2.2) are equivalent. We say that a kernel function is centered if it can be written as

$$k\left(x_i, x_j\right) = \left\langle \tilde{k}(x_i, \cdot) - \frac{1}{n}\sum_{l=1}^{n} \tilde{k}(x_l, \cdot), \tilde{k}(x_j, \cdot) - \frac{1}{n}\sum_{l=1}^{n} \tilde{k}(x_l, \cdot) \right\rangle ,$$

where $\tilde{k} \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is also a kernel function. To see that the two objectives are equivalent, first observe that from the definition of centered kernel function it follows that

$$\frac{1}{n}\sum_{i=1}^{n} k(x_i, \cdot) = \frac{1}{n}\sum_{i=1}^{n} \tilde{k}(x_i, \cdot) - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{l=1}^{n} \tilde{k}(x_l, \cdot) = 0 .$$

Thus, we can rewrite the optimization problem from Eq. (2.1) to match that in Eq. (2.3), i.e.,

$$f_1 = \underset{\|f\|_{\mathcal{H}}=1}{\operatorname{argmax}} \sum_{i=1}^{n} \left\langle f, k(x_i, \cdot) - \frac{1}{n}\sum_{l=1}^{n} k(x_l, \cdot) \right\rangle^2 = \underset{\|f\|_{\mathcal{H}}=1}{\operatorname{argmax}} \sum_{i=1}^{n} f(x_i)^2 .$$

For simplicity of our derivations, in the remainder of the section we, without loss of generality, assume that the kernel function is centered and that the optimization problem in Eq. (2.3) defines the first principal component. To obtain successive principal components, additional constraints need to be imposed on the objective in problem (2.3) so that the principal components are not correlated. Thus, to compute the s-th principal component ($s > 1$), the following optimization problem needs to be solved (Schölkopf et al., 1999)

$$\begin{aligned} f_s \; = \; & \underset{f \in \mathcal{H}}{\operatorname{argmax}} \quad \sum_{i=1}^{n} f(x_i)^2 \\ s.t. \quad & \|f\|_{\mathcal{H}} = 1 \; \wedge \; \langle f, f_i \rangle = 0 \text{ for } i = 1, 2, \ldots, s-1 . \end{aligned} \tag{2.4}$$

In Section 2.1.2, we show that the representer theorem (Wahba, 1990) applies to the optimization problems from Eq. (2.3) and (2.4). Building on this result, we then demonstrate how to solve these problems and compute the kernel principal components (Section 2.1.3).

2.1.2 Representer Theorem

Having formulated optimization problems for the computation of the first principal component, we proceed to show that the representer theorem (Wahba, 1990) applies to these problems. If the kernel function is centered, we can rewrite problems (2.1) and (2.2) as

$$\max_{\|f\|_{\mathcal{H}}=1} \sum_{i=1}^{n} f(x_i)^2 = \max_{f \in \mathcal{H}} \frac{\sum_{i=1}^{n} f(x_i)^2}{\|f\|_{\mathcal{H}}^2} = \min_{f \in \mathcal{H}} \frac{\|f\|_{\mathcal{H}}^2}{\sum_{i=1}^{n} f(x_i)^2} = \min_{\sum_{i=1}^{n} f(x_i)^2 = 1} \|f\|_{\mathcal{H}}^2 \ . \tag{2.5}$$

The first equality follows from the fact that in the second optimization problem, for an optimal solution $f \in \mathcal{H}$, the identical value of the objective is attained at $c \cdot f \in \mathcal{H}$ with $c \in \mathbb{R} \setminus \{0\}$. The same argument is used to derive the last equality in Eq. (2.5). A component $f \in \mathcal{H}$ and this implies that it can be expressed as $f = u + v$, where $u \in \mathcal{H}_X = \operatorname{span}(\{k(x, \cdot) \mid x \in X\})$ and $v \perp \mathcal{H}_X$. Plugging this representation of a component into optimization problem (2.5), we deduce that the first principal component satisfies

$$f_1 = \operatorname*{argmin}_{\sum_{i=1}^{n} f(x_i)^2} \|f\|_{\mathcal{H}}^2 = \operatorname*{argmin}_{\sum_{i=1}^{n} u(x_i)^2 = 1} \|u\|_{\mathcal{H}}^2 + \|v\|_{\mathcal{H}}^2 = \operatorname*{argmin}_{\sum_{i=1}^{n} u(x_i)^2 = 1} \|u\|_{\mathcal{H}}^2 \ .$$

Hence, the first principal component can be expressed as a linear combination of evaluation functionals defined by data instances, i.e., $f_1 = \sum_{i=1}^{n} \alpha_{1,i} k(x_i, \cdot)$ with $\alpha_{1,i} \in \mathbb{R}$ and $i = 1, 2, \cdots, n$. Having shown this, we have derived a version of the representer theorem for the problem of finding the first principal component in kernel principal component analysis.

To derive the representer theorem for the s-th principal component (with $s > 1$), let us consider the following optimization problem

$$\begin{aligned} f_s \ = \ & \operatorname*{argmax}_{f \in \mathcal{H}} \quad \frac{\sum_{i=1}^{n} f(x_i)^2}{\|f\|_{\mathcal{H}}^2} \\ s.t. \quad & \quad \langle f, f_i \rangle = 0 \text{ for } i = 1, 2, \ldots, s-1 \ . \end{aligned} \tag{2.6}$$

On the one hand, for an optimal solution $f \in \mathcal{H}$ of problem (2.6), the optimal value of the optimization objective is also attained at $c \cdot f \in \mathcal{H}$ with $c \in \mathbb{R} \setminus \{0\}$. Thus, the optimization problems from Eq. (2.4) and (2.6) are equivalent. On the other hand, the latter optimization problem is equivalent to

$$\begin{aligned} f_s \ = \ & \operatorname*{argmin}_{f \in \mathcal{H}} \quad \|f\|_{\mathcal{H}}^2 \\ s.t. \quad & \quad \sum_{i=1}^{n} f(x_i)^2 = 1 \ \wedge \ \langle f, f_i \rangle = 0 \text{ for } i = 1, 2, \ldots, s-1 \ . \end{aligned} \tag{2.7}$$

Let us now consider the case with $s = 2$. Similar to the reasoning above, $f \in \mathcal{H}$ can be expressed as $f = u + v$ with $u \in \mathcal{H}_X$ and $v \perp \mathcal{H}_X$. From the fact that the first principal component $f_1 \in \mathcal{H}_X$ it follows that $\langle f, f_1 \rangle = \langle u, f_1 \rangle$. On the other hand, from the reproducing property of the kernel it follows that $\sum_{i=1}^{n} f(x_i)^2 = \sum_{i=1}^{n} u(x_i)^2$. Thus, the constraints from problem (2.7) are independent of v and the minimum value of the optimization objective in Eq. (2.7) is attained at $f_2 \in \mathcal{H}_X$. Having shown this, we have demonstrated that the representer theorem applies to the problem for the computation of the second principal component. For $s > 2$, the claim follows by reasoning along the lines of the case $s = 2$.

2.1.3 Derivation of Kernel Principal Components

The fact that the representer theorem applies to the optimization problems defining principal components allows us to exploit the kernel trick and transform these problems into optimization problems over a Euclidean space. For that, let K denote the kernel matrix with entries $K_{ij} = k\left(x_i, x_j\right)$ and columns K_i $(1 \leq i, j \leq n)$. From the representer theorem it follows that a principal component can be expressed as $f_s = \sum_{i=1}^{n} \alpha_{s,i} k\left(x_i, \cdot\right)$, where $\alpha_{s,i} \in \mathbb{R}$, $1 \leq s \leq r$, and $1 \leq i \leq n$. Now, using this representation and the reproducing property of the kernel we get

$$\|f\|_{\mathcal{H}}^2 = \left\langle \sum_{i=1}^{n} \alpha_i k(x_i, \cdot), \sum_{j=1}^{n} \alpha_j k(x_j, \cdot) \right\rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k\left(x_i, x_j\right) = \alpha^\top K \alpha \ ,$$

$$\sum_{i=1}^{n} f\left(x_i\right)^2 = \sum_{i=1}^{n} \left\langle \sum_{j=1}^{n} \alpha_j k(x_j, \cdot), k(x_i, \cdot) \right\rangle^2 = \sum_{i=1}^{n} \left\| K_i^\top \alpha \right\|_2^2 = \alpha^\top \left(\sum_{i=1}^{n} K_i K_i^\top \right) \alpha = \alpha^\top K^2 \alpha \ ,$$

$$\langle f, f_s \rangle = \left\langle \sum_{j=1}^{n} \alpha_j k(x_j, \cdot), \sum_{l=1}^{n} \alpha_{s,l} k(x_l, \cdot) \right\rangle = \sum_{j=1}^{n} \sum_{l=1}^{n} \alpha_j \alpha_{s,l} k\left(x_j, x_l\right) = \alpha^\top K \alpha_s \ .$$

Let $K = UDU^\top$ be an eigendecomposition of the symmetric and positive definite kernel matrix K. Then, if we denote with $K^{1/2} = UD^{1/2}U^\top$ and $\gamma = K^{1/2}\alpha$, the optimization problem for the first principal component can be written as

$$\max_{\alpha \in \mathbb{R}^d} \frac{\alpha^\top K^2 \alpha}{\alpha^\top K \alpha} = \max_{\gamma \in \mathbb{R}^d} \frac{\gamma^\top K \gamma}{\gamma^\top \gamma} \ .$$

The latter optimization problem is known as the Rayleigh–Ritz quotient (e.g., see Lütkepohl, 1997) and an optimal solution is obtained when γ is the eigenvector corresponding to the largest eigenvalue of K. Assuming that the eigenvalues $\{\lambda_i\}_{i=1}^{n}$ in diagonal matrix D are sorted in descending order, the result implies that $\gamma^* = u_1$, where u_1 is the first column in matrix U and an eigenvector corresponding to the eigenvalue λ_1. Hence, we deduce that

$$\alpha_1 = K^{-1/2}\gamma^* = UD^{-1/2}U^\top u_1 = \frac{1}{\sqrt{\lambda_1}} u_1 \quad \land \quad f_1 = \sum_{i=1}^{n} \alpha_{1,i} k\left(x_i, \cdot\right) \ .$$

For the s-th principal component we have that the optimization problem from Eq. (2.4) can be transformed into the matrix form as

$$\begin{aligned} \max_{\alpha \in \mathbb{R}^d} \quad & \alpha^\top K^2 \alpha \\ s.t. \quad & \alpha^\top K \alpha = 1 \ \land \ \alpha^\top K \alpha_i = 0 \text{ for } i = 1, 2, \ldots, s-1 \ . \end{aligned} \tag{2.8}$$

To solve this problem, we first form the Lagrangian function

$$\mathcal{L}(\alpha, \mu) = \alpha^\top K^2 \alpha - \mu_0 \left(\alpha^\top K \alpha - 1 \right) - \sum_{i=1}^{s-1} \mu_i \alpha^\top K \alpha_i \ , \tag{2.9}$$

where μ is a vector of Lagrange multipliers and μ_0 is a scalar Lagrange multiplier. Setting the gradient of the Lagrangian to zero we obtain the stationary constraints

$$\begin{aligned} 2K^2\alpha &= 2\mu_0 K\alpha + \sum_{i=1}^{s-1} \mu_i K \alpha_i \ , \\ \alpha^\top K \alpha &= 1 \ \land \ \alpha^\top K \alpha_i = 0 \text{ for } 1 \leq i \leq s-1 \ . \end{aligned}$$

Let us now focus on the case with $s = 2$. As shown above, the first principal component $f_1 = \sum_{j=1}^{n} \alpha_{1,j} k(x_j, \cdot)$ is given by $\alpha_1 = {}^{u_1}/\sqrt{\lambda_1}$. From here it then follows that the corresponding orthogonality constraint, $\alpha^\top K \alpha_1 = 0$, can be transformed into

$$\alpha^\top U D U^\top u_1 = 0 \implies \alpha^\top u_1 = 0 \; .$$

Thus, after multiplying the first stationary constraint by $\alpha_1^\top$ from the left, we deduce that

$$\mu_1 = 2\alpha_1^\top K^2 \alpha = \frac{2}{\sqrt{\lambda_1}} u_1^\top U D^2 U^\top \alpha = 2\lambda_1^{3/2} u_1^\top \alpha = 0 \; .$$

Plugging this into the first stationary constraint and multiplying it by $\alpha^\top$ from the left, we conclude that $\mu_0 = \alpha^\top K^2 \alpha$. From here, after factoring out the kernel matrix from the first stationary constraint, we obtain that

$$K\alpha = \mu_0 \alpha \; .$$

Thus, μ_0 is an eigenvalue of K and the optimal value of the optimization objective in Eq. (2.8). From this result and the fact that $\alpha \perp u_1$ it follows that α is collinear with the eigenvector u_2 corresponding to the second largest eigenvalue λ_2. The fact that $\alpha^\top K \alpha = 1$ now implies that the second principal component, $f_2 = \sum_{j=1}^{n} \alpha_{2,j} k(x_j, \cdot)$, is given by $\alpha_2 = {}^{u_2}/\sqrt{\lambda_2}$.

Having demonstrated how to derive the kernel principal components for $s = 1$ and $s = 2$, we conclude our description of kernel principal component analysis. The remaining principal components (for $s > 2$) can be derived by reasoning analogously as in the case $s = 2$.

2.2 Semi-Supervised Learning

This section provides an overview of semi-supervised learning and assumptions specific to this learning task. The definitions and terminology used throughout the section follow along the lines of the edited volume 'Semi-supervised learning' by Chapelle et al. (2006).

2.2.1 Problem Setting

Supervised learning is a learning task in which the goal is to find a mapping from an instance space $\mathcal{X}$ to a space of labels $\mathcal{Y}$ based on a training sample $\mathbf{z} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of n examples sampled independently from a Borel probability measure ρ defined on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. The task can be evaluated on test examples sampled independently from ρ and unavailable to the learning algorithm during training. Semi-supervised learning is a class of supervised learning tasks where the algorithm required to extract a functional dependence from training data, in addition to having a training sample $\mathbf{z} \in \mathcal{Z}^n$, has also a set of unlabeled instances at its disposal. More formally, in semi-supervised learning the training data consists of a training sample $\mathbf{z} \in \mathcal{Z}^n$ and a set $X' = \{x_{n+1}, \ldots, x_{n+n'}\}$ of n' unlabeled instances that are sampled independently from the marginal probability measure $\rho_{\mathcal{X}}$ defined on $\mathcal{X}$.

A related learning task, sometimes confused with semi-supervised learning, is that of transductive learning. In contrast to semi-supervised learning, the goal in such tasks is to predict correct labels on unlabeled instances $X' = \{x_{n+1}, \ldots, x_{n+n'}\}$. Typically, semi-supervised learning algorithms are evaluated in transductive setting with test samples consisting only of unlabeled instances available during training (i.e., a subset of X') and their labels.

2.2.2 When Can Unlabeled Data Aid in Learning?

In general, unlabeled instances do not necessarily aid in semi-supervised learning tasks. Moreover, there are cases when additional unlabeled instances negatively affect the predictive performance of learning algorithms (e.g., see Chapter 4 in Chapelle et al., 2006). For unlabeled data to be useful in a semi-supervised learning task it needs to contain information relevant to the inference of a target concept. More formally (Chapelle et al., 2006), the knowledge on $\rho_{\mathcal{X}}$ extracted from unlabeled instances has to carry information that is useful in the inference of $\rho(y \mid x)$. Thus, for semi-supervised learning to work certain assumptions on the data distribution will need to hold. In their edited volume on semi-supervised learning, Chapelle et al. (2006) formulate three standard assumptions of semi-supervised learning: *i*) smoothness assumption, *ii*) cluster assumption, and *iii*) manifold assumption. At least one of these assumptions on the data distribution will need to be satisfied for unlabeled data to aid in learning. In the remainder of the section, similar to Chapelle et al. (2006), we cover each of these three assumptions by focusing on the problem of classification.

> **Smoothness assumption:** If two instances x_1 and x_2 from a high-density region of $\rho_{\mathcal{X}}$ are close, then so should be the corresponding outputs y_1 and y_2.

This is an adaptation of a standard smoothness assumption for supervised learning where it is assumed that if two instances are close in the instance space then so should be the corresponding outputs y_1 and y_2. Such assumptions are required to be able to generalize from training data to unseen instances. In contrast to the smoothness assumption for supervised learning, the smoothness assumption for semi-supervised learning depends on the marginal distribution of instances and this is precisely the source of additional information that allows improvement in predictive performance of learning algorithms as a result of taking into account the unlabeled instances in addition to labeled training examples.

> **Cluster assumption:** If instances are in the same cluster, then they are likely to of the same class.

A cluster is often defined as a set of instances that can be connected by short curves which traverse only high-density regions of an instance space (Chapelle et al., 2006). Thus, for classification problems the cluster assumption is equivalent to the semi-supervised smoothness assumption. The motivation for the cluster assumption comes from datasets in which each class tends to form a cluster and in those cases unlabeled data can aid in determining boundaries of clusters (i.e., curves encompassing sets of instances) which correspond to decision boundaries separating the classes. As the boundary of a cluster cannot pass through a high-density region of the instance space, the assumption implies that the boundary lies in a low-density region. Here it is also important to note that the assumption does not state that clusters are compact structures consisting only of instances of the same class, but that frequently instances from the same class are observed close together in a high-density region of the instance space.

> **Manifold assumption:** The instances lie (roughly) on a low-dimensional manifold.

A manifold is a topological space that is locally Euclidean, i.e., around every point, there is a neighborhood that is topologically the same as the open unit ball in a Euclidean space (Rowland, 2017). To illustrate it consider the Earth which is roughly spherical in shape but in a small neighborhood it looks flat and not round. Such small neighborhoods can be accurately represented by planes (e.g., geographical maps) unlike the Earth itself. In general, any object that is *nearly* flat on small scales is a manifold. The manifold assumption for semi-supervised learning can be seen as a link between the smoothness assumptions for supervised and semi-supervised learning. In particular, a manifold can be seen as an approximation to a high-density region of the instance space and in this case the semi-supervised smoothness assumption is identical to the supervised smoothness assumption restricted to the data on the manifold. When the manifold assumption is satisfied, additional unlabeled instances can aid in approximating the manifold boundaries and allow embedding of data from a possibly high-dimensional input space to a low-dimensional space of the manifold. In this way, learning algorithms can overcome problems faced in high-dimensional spaces where exponentially many samples are needed for consistent estimation due to the fact that volume grows exponentially with the dimension of the problem.

2.3 Knowledge-Based Kernel Principal Component Analysis

Having reviewed the prerequisites, kernel principal component analysis (Section 2.1) and semi-supervised learning (Section 2.2), we now propose an extension of the former in which the principal components adhere to a set of knowledge-based constraints while still maximizing the variance at data instances. We start our presentation by introducing means for incorporating domain knowledge into kernel principal component analysis (Section 2.3.1). Following this, we show that the representer theorem (Wahba, 1990) applies to the optimization problems for the computation of knowledge-based kernel principal components (Section 2.3.2), which are defined over a reproducing kernel Hilbert space of functions. We then apply the representer theorem to these optimization problems and express the knowledge-based kernel principal components via optimization problems over a Euclidean space (Section 2.3.3).

2.3.1 Definition of Knowledge-Based Kernel Principal Components

We first give a formal definition of our knowledge-based constraints and then introduce two variants of semi-supervised kernel principal component analysis capable of incorporating these constraints into the process of finding functional directions with maximum-variance at data instances. In doing so, we retain the notation introduced in the previous sections.

2.3.1.1 Knowledge-Based Constraints

In this section, we formulate three types of constraints allowing experts to incorporate their domain knowledge into kernel principal component analysis. The first constraint allows one to specify an approximate or explicit value of a kernel principal component at a particular instance. In a data visualization generated by knowledge-based kernel principal component analysis such constraints express the placements of instances in a one/two/three dimensional space of the visualization. The second constraint allows for weak supervision via a pair of instances that should be close/far from each other in the projection space. The third constraint is characteristic to classification tasks and allows one to group instances by their class. Henceforth, we refer to points for which a constraint is specified as control points.

The placement of control points along a principal component can be incorporated into kernel principal component analysis as a soft constraint with

$$\Omega(f, \mathcal{A}_s) = \frac{1}{|\mathcal{A}_s|} \sum_{(x, y_s) \in \mathcal{A}_s} \|f(x) - y_s\|^2 ,$$

where y_s is the placement coordinate 'along' the s-th projection axis corresponding to an instance x and $\mathcal{A}_s$ denotes a set of such placements. An alternative way is to treat the placements as hard constraints (Paurat et al., 2013b) and incorporate them into a linear operator defined on the s-th functional direction as

$$f(x) = y_s \text{ for all } (x, y_s) \in \mathcal{A}_s \ .$$

Note that soft constraints allow some displacement which can lead to better visualizations if noise is to be expected in the positions of the control points.

Domain knowledge can also be expressed in terms of similarity between points and such knowledge-based constraints can, for instance, be defined by pairing points which should or should not be placed close to each other. Squared distances between projections of paired points are then minimized for must-link pairs and maximized for cannot-link pairs, i.e.,

$$\Omega(f, \mathcal{B}_s) = \frac{1}{|\mathcal{B}_s|} \sum_{(i,l,y_{il}) \in \mathcal{B}_s} y_{il} \left(f(x_i) - f(x_l)\right)^2 \ ,$$

where $y_{il} = +1$ for a must-link and $y_{il} = -1$ for a cannot-link constraint, and $\mathcal{B}_s$ denotes a set of such pairwise constraints.

Beside pairwise constraints and explicit placements of control points, domain knowledge can be incorporated into kernel principal component analysis by providing class labels for a small number of instances. In particular, a soft constraint corresponding to this type of supervision can be defined as

$$\Omega(f, \mathcal{C}_s) = \sum_{(x,y) \in \mathcal{C}_s} \sum_{i \in \text{k-NN}(x)} \frac{w_i}{\sum_{j \in \text{k-NN}(x)} w_j} y f(x_i) + \sum_{(x,*) \notin \mathcal{C}_s} \sum_{i \in \text{k-NN}(x)} \frac{w_i}{\sum_{j \in \text{k-NN}(x)} w_j} f(x) f(x_i) \ ,$$

where w_i reflects the similarity between instances x and x_i, $y = \pm 1$, $\mathcal{C}_s$ denotes a set of classification constraints, and $\text{k-NN}(x)$ denotes a set with the arguments of the k nearest unlabeled neighbors of an instance x. The motivation for this soft constraint comes from the consideration of an upper bound on the leave-one-out error of the k-NN classifier given in Joachims (2003). More specifically, a similarity weighted k-NN classifier makes a leave-one-out mistake on example (x, y) when

$$\delta(x) = \sum_{i \in \text{k-NN}(x)} y y_i \frac{w_i}{\sum_{j \in \text{k-NN}(x)} w_j} \leq 0 \ .$$

From here it then follows that an upper bound on the zero-one leave-one-out error is given by (Joachims, 2003)

$$\mathcal{L}_{\text{k-NN}}(X, Y) \leq \frac{1}{n} \sum_{i=1}^{n} \left(1 - \delta(x_i)\right) \ .$$

Having formally defined our knowledge-based constraints, we proceed to next section in which we define knowledge-based kernel principal components.

2.3.1.2 Knowledge-Based Kernel Principal Components

We propose two variants of kernel principal component analysis which extend this unsupervised method to semi-supervised tasks via incorporation of domain-knowledge constraints described in the previous section. The two proposed approaches stem from the transformations of the optimization problems for kernel principal component analysis considered in Eq. (2.5). Similar to Section 2.1.2, in order to simplify our derivation we, without loss of generality, assume that the kernel function is centered.

In the first approach that extends kernel principal component analysis to semi-supervised tasks, we iteratively find the constant $\mathcal{H}_X$-norm (discussed subsequently) knowledge-based kernel principal components $f_1, \dots, f_r \in \mathcal{H}$ by solving the following optimization problem

$$\begin{aligned} f_s = \underset{f \in \mathcal{H}}{\operatorname{argmax}} \quad & \frac{1}{n}\sum_{i=1}^{n} f(x_i)^2 - \Omega(f,s) - \lambda_0 \sum_{s'=1}^{s-1} \langle f, f_{s'} \rangle^2 \\ \text{subject to} \quad & \|f\|_{\mathcal{H}_X} = R \ \wedge \ \Upsilon(f, \mathcal{A}_s) = 0 \ , \end{aligned} \tag{2.10}$$

where $\Omega(f,s) = \lambda_1 \Omega(f,\mathcal{A}_s) + \lambda_2 \Omega(f,\mathcal{B}_s) - \lambda_3 \Omega(f,\mathcal{C}_s)$, Υ is a linear operator over direction f defined using the hard placements of control points, and $\lambda_0, \lambda_1, \lambda_2, \lambda_3, R \in \mathbb{R}^+$ are hyperparameters of the optimization problem. Additionally, Υ can be used to express a hard orthogonality constraint over the computed directions, i.e.,

$$\langle f, f_{s'} \rangle = 0 \text{ for all } s' = 1, \dots, s-1 \ .$$

In contrast to this, the current objective expresses the orthogonality between components using a soft constraint term consisting of the sum of squared inner products between the current and already computed knowledge-based principal components $f_1, \dots, f_{s-1}$.

Alternatively, it is possible to formulate an extension of kernel principal component analysis by starting from a different formulation of it, which reverses the roles of the norm of the projection direction and its variance at data instances. More specifically, the second approach extends kernel principal component analysis to semi-supervised tasks by iteratively finding the constant variance knowledge-based kernel principal components $f_1, \dots, f_r \in \mathcal{H}$ which adhere to the domain knowledge constraints, i.e.,

$$\begin{aligned} f_s = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \quad & \|f\|_{\mathcal{H}}^2 + \Omega(f,s) + \lambda_0 \sum_{s'=1}^{s-1} \langle f, f_{s'} \rangle^2 \\ \text{subject to} \quad & \frac{1}{n}\sum_{i=1}^{n} f(x_i)^2 = R^2 \ \wedge \ \Upsilon(f, \mathcal{A}_s) = 0 \ . \end{aligned} \tag{2.11}$$

In contrast to the optimization problem in Eq. (2.10), the latter optimization problem does not restrict the norm of knowledge-based principal components to a subspace $\mathcal{H}_X \subset \mathcal{H}$. The reason for this is given in next section, where we show that the representer theorem (Wahba, 1990; Schölkopf et al., 2001; Dinuzzo and Schölkopf, 2012) applies to these two problems.

2.3.2 Representer Theorem

Let us now show that the representer theorem (Wahba, 1990; Schölkopf et al., 2001; Dinuzzo and Schölkopf, 2012) applies to problems (2.10) and (2.11). First, we show that the theorem applies to all knowledge-based constraints. In the soft constraint terms $\Omega(f,\mathcal{A}_s)$, $\Omega(f,\mathcal{B}_s)$, and $\Omega(f,\mathcal{C}_s)$ an optimizer $f \in \mathcal{H}$, $f = u + v$ with $u \in \mathcal{H}_X$ and $v \perp \mathcal{H}_X$, is defined with a data instance $x \in X$. Thus, from the reproducing property of the kernel we have that the optimizer in these constraints is an element of $\mathcal{H}_X$, i.e.,

$$f(x) = \langle f, k(x,\cdot) \rangle = \langle u + v, k(x,\cdot) \rangle = \langle u, k(x,\cdot) \rangle = u(x) \ .$$

The hard constraint term Υ is also independent of v as it holds that $f(x) = u(x)$ for all $x \in X$. Hence, the representer theorem applies to both, soft and hard, knowledge-based constraints.

Let us now show that the representer theorem holds for problem (2.10). In the computation of the first extreme variance direction, f_1, there is no soft orthogonality term in the optimization objective. Plugging the representation of an optimizer expressed in terms of the data span and its orthogonal complement, $f_1 = u_1 + v_1$ with $u_1 \in \mathcal{H}_X$ and $v_1 \perp \mathcal{H}_X$, into Eq. (2.10), we conclude (noting that the theorem applies to knowledge-base constraints) that the optimization objective is independent of v_1, and that the representer theorem applies to this case. For the computation of the s-th variance direction f_s with $s > 1$, we additionally have orthogonality terms $\langle f_s, f_{s'} \rangle = \langle u_s + v_s, f_{s'} \rangle = \langle u_s, f_{s'} \rangle$ for $s' < s$, which are also independent of v_s. Therefore, the representer theorem applies to problem (2.10) and we can express an optimizer as $f_s = \sum_i \alpha_{s,i} k(x_i, \cdot)$ with $\alpha_{s,i} \in \mathbb{R}$.

To show that the representer theorem applies to problem (2.11), first observe that the constant variance constraint, together with all knowledge-based constraints, are independent of $v_s \perp \mathcal{H}_X$. Thus, the only term that depends on v_s in problem (2.11) is the norm of the principal component in the optimization objective. Now, as the knowledge-based kernel principal component is obtained by minimizing this objective the norm is minimized too and the minimum value is attained when $v_s = 0$. Hence, the representer theorem applies also to problem (2.11) and we can express an optimizer as $f_s = \sum_i \alpha_{s,i} k(x_i, \cdot)$ with $\alpha_{s,i} \in \mathbb{R}$.

Here it is important to note that in problem (2.10), the norm constraint is defined over $\mathcal{H}_X$ instead of the whole Hilbert space. The reason for this lies in the fact that with norm defined over $\mathcal{H}$, the representer theorem no longer applies to that optimization problem. In the remainder of the section, we focus on solving the optimization problem in Eq. (2.11) and note that an optimal solution to problem (2.10) can be obtained using the same techniques. Moreover, a detailed derivation for the latter problem can be found in Oglic et al. (2014b).

2.3.3 Derivation of Knowledge-Based Kernel Principal Components

Having shown that the representer theorem applies to problem (2.11), we now transform it to an optimization problem over a Euclidean space.

As the representer theorem applies to $\Omega(f, \mathcal{A}_s)$ term, we are able to express it using the kernel matrix and coefficients defining $f = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot) \in \mathcal{H}_X$, i.e.,

$$\Omega(f, \mathcal{A}_s) = \frac{1}{|\mathcal{A}_s|} \alpha^T K A K \alpha - \frac{2}{|\mathcal{A}_s|} y_s^T A K \alpha \ ,$$

where K denotes the kernel matrix and A is a diagonal matrix such that $A_{ii} = 1$ when a label is provided for instance x_i, otherwise $A_{ii} = 0$. Alternatively, one could assign weights to particular examples by setting $A_{ii} = a_i^2$ with $a_i \in \mathbb{R}$. Similarly, we can express the hard constraint Υ as

$$\Upsilon(f, \mathcal{A}_s) = AK\alpha - Ay_s = 0 \ .$$

To obtain a compact matrix-form representation of the pairwise constraint term $\Omega(f, \mathcal{B}_s)$, we start by transforming it using the representer theorem,

$$\Omega(f, \mathcal{B}_s) = \sum_{(i,j,y_{ij}) \in \mathcal{B}_s} y_{ij} \left(f(x_i) - f(x_j) \right)^2 = \sum_{(i,j,y_{ij}) \in \mathcal{B}_s} y_{ij} \left(\left(\mathbf{e}_i - \mathbf{e}_j \right)^\top K\alpha \right)^2 =$$

$$\alpha^\top K \left(\sum_{(i,j,y_{ij}) \in \mathcal{B}_s} y_{ij} \Delta_{ij} \Delta_{ij}^\top \right) K\alpha \ ,$$

where $\mathbf{e}_i$ is the canonical basis vector with one at the i-th coordinate and $\Delta_{ij} = \mathbf{e}_i - \mathbf{e}_j$. Thus, denoting with $B = \sum_{(i,j,y_{ij}) \in \mathcal{B}_s} y_{ij} \Delta_{ij} \Delta_{ij}^\top$ the *pairwise constraint matrix* we obtain a compact representation of the corresponding soft constraint

$$\Omega\left(f, \mathcal{B}_s\right) = \frac{1}{|\mathcal{B}_s|} \alpha^\top K B K \alpha \ .$$

We note here that the pairwise constraint matrix B is in fact a Laplacian matrix of the graph given by edge weights y_{ij}, where $1 \leq i, j \leq n$.

Before we give a compact matrix-form representation of the classification constraint, we need to introduce a weighted k-NN adjacency matrix. Let $\overline{G}$ denote a weighted k-NN adjacency matrix such that

$$\overline{g}_{ij} = \begin{cases} \frac{w_{ij}}{\sum_{l \in \text{k-NN}(x_i)} w_{il}} & \text{if } j \in \text{k-NN}\left(x_i\right) \\ 0 & \text{otherwise} \ . \end{cases}$$

Similar to Joachims (2003), in order to make this k-NN relation symmetric we replace the matrix $\overline{G}$ with the adjacency matrix $G = {\overline{G}+\overline{G}^\top}/2$. We can now using the matrix G express the first summand in the classification constraint as

$$\sum_{(x_i, y_i) \in \mathcal{C}_s} \sum_{j \in \text{k-NN}(x_i)} g_{ij} y_i f\left(x_j\right) = \sum_{(x_i, y_i) \in \mathcal{C}_s} \sum_{j \in \text{k-NN}(x_i)} g_{ij} y_i \mathbf{e}_j^\top K \alpha = y^\top C G K \alpha \ ,$$

where C is a diagonal matrix such that $C_{ii} = 1$ when a label is provided for instance x_i, otherwise $C_{ii} = 0$. Similarly, for the second summand in the classification constraint we have

$$\sum_{(x_i, *) \notin \mathcal{C}_s} \sum_{j \in \text{k-NN}(x_i)} g_{ij} f\left(x_i\right) f\left(x_j\right) = \sum_{(x_i, *) \notin \mathcal{C}_s} \sum_{j \in \text{k-NN}(x_i)} g_{ij} \mathbf{e}_i^\top K \alpha \mathbf{e}_j^\top K \alpha = \alpha^\top K \left(\mathbb{I} - C\right) G K \alpha \ .$$

Hence, we can express the classification constraint in matrix-form as

$$\Omega\left(f, \mathcal{C}_s\right) = \frac{1}{\left|\mathcal{C}_s^\perp\right|} \alpha^\top K C^\perp G K \alpha + \frac{1}{|\mathcal{C}_s|} y^\top C G K \alpha \ ,$$

where $\mathcal{C}_s^\perp = \{x \in X \mid (x, *) \notin \mathcal{C}_s\}$, $|\mathcal{C}_s^\perp| = n - |\mathcal{C}_s|$, and $C^\perp = \mathbb{I} - C$.

Having expressed knowledge-based constraints in the matrix form, let us now do the same for the remaining terms in the optimization problem from Eq. (2.11). As the variance term has already been expressed in Section 2.1.3, it remains only to express the soft orthogonality term using the representer theorem. For this term, we have that it holds

$$\sum_{s'=1}^{s-1} \langle f, f_{s'} \rangle^2 = \sum_{s'=1}^{s-1} \left(\alpha^\top K \alpha_{s'}\right)^2 = \alpha^\top K \left(\sum_{s'=1}^{s-1} \alpha_{s'} \alpha_{s'}^\top\right) K \alpha \ .$$

Denoting with $H = \sum_{s'=1}^{s-1} \alpha_{s'} \alpha_{s'}^\top$, we can rewrite the optimization problem in Eq. (2.11) as

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^n} \quad & \alpha^\top K \left(K^{-1} + \frac{\lambda_1}{|\mathcal{A}_s|} A + \frac{\lambda_2}{|\mathcal{B}_s|} B - \frac{\lambda_3}{\left|\mathcal{C}_s^\perp\right|} C^\perp G + \lambda_0 H\right) K \alpha - y_s^\top \left(2 \frac{\lambda_1}{|\mathcal{A}_s|} A + \frac{\lambda_3}{|\mathcal{C}_s|} C G\right) K \alpha \\ s.t. \quad & \alpha^\top K^2 \alpha = R^2 \ \wedge \ A_{hard}\left(K \alpha - y_s\right) = 0 \ , \end{aligned}$$

where A_{hard} is a diagonal matrix specifying the hard placements of the control points over the s-th knowledge-based kernel principal component.

Let us assume now that all constraints are soft constraints (hard constrains will be addressed later) and set

$$
\begin{aligned}
& z = K\alpha \;, \\
& S = K^{-1} + \frac{\lambda_1}{|\mathcal{A}_s|}A + \frac{\lambda_2}{|\mathcal{B}_s|}B - \frac{\lambda_3}{\left|\mathcal{C}_s^{\perp}\right|}C^{\perp}G + \lambda_0 H \;, \text{ and} \\
& b = \left(\frac{\lambda_1}{|\mathcal{A}_s|}A + \frac{\lambda_3}{2\,|\mathcal{C}_s|}CG\right)y_s \;.
\end{aligned}
$$

Then, the latter optimization problem becomes

$$
\begin{aligned}
z_s = \operatorname*{argmin}_{z \in \mathbb{R}^n} \quad & z^\top S z - 2b^\top z \\
s.t. \quad & z^\top z = R^2 \;.
\end{aligned}
\tag{2.12}
$$

In case the soft orthogonality constraint is replaced with the hard one, the optimization problem from Eq. (2.11) becomes

$$
\begin{aligned}
z_s = \operatorname*{argmin}_{z \in \mathbb{R}^n} \quad & z^\top \overline{S} z - 2b^\top z \\
s.t. \quad & z^\top z = R^2 \;\wedge\; L^\top z = 0 \;,
\end{aligned}
\tag{2.13}
$$

where $L \in \mathbb{R}^{n \times (s-1)}$ with coefficient vectors $\alpha_{s'}$ corresponding to knowledge-based kernel principal components $f_{s'}$ ($1 \leq s' < s$) as columns, and $\overline{S} = K^{-1} + \frac{\lambda_1}{|\mathcal{A}_s|}A + \frac{\lambda_2}{|\mathcal{B}_s|}B - \frac{\lambda_3}{|\mathcal{C}_s^{\perp}|}C^{\perp}G$.

Let us now return to the hard constraint on the placement of control points. From our derivations, we know that this constraint can be expressed as

$$
0 = A_{hard}\,(K\alpha - y_s) = A_{hard}(z - y_s) \;.
$$

The matrix A_{hard} is a diagonal matrix with ones at the diagonal entries corresponding to labeled instances specified with the hard constraint (the remaining diagonal entries are equal to zero). This hard constraint defines a homogeneous linear system of rank given by the number of such labeled instances. As we will demonstrate in the next section, this constraint fixes a part of the solution vector z_s and, thus, reduces the rank of the optimization problem. The resulting optimization problem (e.g., see Section 2.4.1) is again given as a quadratic form over a hypersphere of constant radius.

2.4 Optimization Problem

As demonstrated in Section 2.3.3, to compute our embedding, for any combination of knowledge-based constraints, we have to solve the following optimization problem

$$
\begin{aligned}
\min_{z \in \mathbb{R}^n} \quad & z^\top S z - 2b^\top z \\
s.t. \quad & z^\top z = R^2 \;\wedge\; L^\top z = d,
\end{aligned}
\tag{2.14}
$$

where $S \in \mathbb{R}^{n \times n}$ is a symmetric matrix, $L \in \mathbb{R}^{n \times m}$ with $m \ll n$, $b \in \mathbb{R}^n$, $d \in \mathbb{R}^m$, and $R \in \mathbb{R}$. The problem in Eq. (2.14) is defined with an indefinite quadratic form over a hypersphere of constant radius and subject to an additional linear equality constraint. Thus, this optimization problem is non-convex with potentially exponentially many local optima (with respect to the rank of the optimization problem). Despite this, building on the work by Forsythe and Golub

(1965) and Gander et al. (1989) it is possible to find a globally optimal solution for this problem in time cubic in the size of the kernel expansion n. In particular, we first show (in Section 2.4.1) how to eliminate the hard linear constraint from the optimization problem in Eq. (2.14) and optimize only a quadratic form over a hypersphere of constant radius. Following this, we show (in Section 2.4.2) how to derive a globally optimal solution for that problem in a closed form. The closed form solution can be numerically unstable to compute and, in Section 2.4.3, we provide an efficient alternative with a numerically stable secular solver.

2.4.1 Eliminating Linear Constraint

For knowledge-based kernel principal component analysis with hard orthogonality or hard placement of control points, we have an additional linear equality constraint in the optimization problem. As the linear constraint is of rank $m \ll n$, it can be eliminated and the initial problem can be transformed to the minimization of a possibly indefinite quadratic form over an $(n-m)$-dimensional hypersphere of constant radius.

In order to eliminate the linear constraint, we perform a QR factorization of its matrix $L = Q\Gamma$, where $Q \in \mathbb{R}^{n\times n}$ is an orthogonal matrix,

$$\Gamma = \begin{bmatrix} \overline{\Gamma} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n\times m} ,$$

and $\overline{\Gamma} \in \mathbb{R}^{m\times m}$ is an upper-triangular matrix. Substituting

$$Q^\top z = \begin{bmatrix} \zeta^* \\ \zeta \end{bmatrix}, \text{ such that } \zeta^* \in \mathbb{R}^m \text{ and } \zeta \in \mathbb{R}^{n-m},$$

linear and hypersphere constraints from Eq. (2.14) can be transformed into

$$\begin{aligned} d = L^\top z = \Gamma^\top (Q^\top z) = \overline{\Gamma}^\top \zeta^* \qquad &\Longrightarrow \qquad \zeta^* = \left(\overline{\Gamma}^\top\right)^{-1} d \,, \\ R^2 = z^\top z = \|\zeta^*\|^2 + \zeta^\top \zeta \qquad &\Longrightarrow \qquad \zeta^\top \zeta = R^2 - \|\zeta^*\|^2 = \hat{R}^2 \,. \end{aligned}$$

As ζ^* is a constant vector, we can rewrite the objective in Eq. (2.14) as a quadratic form over an $(n-m)$-dimensional hypersphere of constant radius. More specifically, the quadratic term can be rewritten as

$$\begin{aligned} & z^\top S z = z^\top Q Q^\top S Q Q^\top z = \zeta^{*\top} S_{11} \zeta^* + 2\zeta^\top S_{12} \zeta^* + \zeta^\top S_{22} \zeta, \text{ where} \\ & Q^\top S Q = \begin{bmatrix} S_{11} & S_{12}^\top \\ S_{12} & S_{22} \end{bmatrix} \text{ with } S_{11} \in \mathbb{R}^{m\times m}, S_{12} \in \mathbb{R}^{(n-m)\times m} \text{ and } S_{22} \in \mathbb{R}^{(n-m)\times(n-m)}. \end{aligned}$$

On the other hand, the linear term can be transformed into

$$b^\top z = b^\top Q Q^\top z = b_1^\top \zeta^* + b_2^\top \zeta \,,$$

where $b_1 \in \mathbb{R}^m$ and $b_2 \in \mathbb{R}^{n-m}$ are the corresponding blocks in the vector $Q^\top b$. Denoting with $\hat{b} = b_2 - S_{12}\zeta^*$ we obtain the resulting optimization problem,

$$\begin{aligned} \min_{\zeta \in \mathbb{R}^{n-m}} \quad & \zeta^\top S_{22} \zeta - 2\hat{b}^\top \zeta \\ s.t. \quad & \zeta^\top \zeta = \hat{R}^2 \,. \end{aligned}$$

2.4.2 Optimization of a Quadratic Form over a Hypersphere

In this section, we review the works by Forsythe and Golub (1965) and Gander et al. (1989), in which two approaches are described for finding a globally optimal solution for the optimization of a quadratic form over a hypersphere of constant radius. To find an optimizer for this optimization problem (e.g., see problem 2.12), we first derive the Lagrange function

$$\mathscr{L}(z,\mu) = z^\top Sz - 2b^\top z - \mu(z^\top z - R^2)\,, \tag{2.15}$$

and set its derivatives to zero, i.e.

$$Sz = b + \mu z \quad \wedge \quad z^\top z = R^2\,. \tag{2.16}$$

Notice here that we have, in order to simplify our notation, performed the change of variable and instead of the variable ζ used in Section 2.4.1, the optimization is performed with respect to the variable z (i.e., as formulated in Eq. 2.12). As the optimization problem from Eq. (2.12) is non-convex, a solution to the system in Eq. (2.16) is only a local optimum for that problem. The following proposition, however, gives a criterion for distinguishing the global optimum of problem (2.12) from the solution set of the system in Eq. (2.16). Alternative and slightly more complex proofs for the same claim are given by Forsythe and Golub (1965) and Gander (1980). Let us now denote the optimization objective from Eq. (2.12) with $\Theta\,(z) = z^\top Sz - 2b^\top z$.

Proposition 2.1. *(Forsythe and Golub, 1965; Gander, 1980) The optimization objective* $\Theta\,(z)$ *attains the minimal value at the tuple* (z,μ) *satisfying the stationary constraints (2.16) with the smallest value of* μ*. Analogously, the maximal value of* $\Theta\,(z)$ *is attained at the stationary tuple with the largest value of the Lagrange multiplier.*

Proof. Let (z_1,μ_1) and (z_2,μ_2) be two tuples satisfying the stationary constraints (2.16) with $\mu_1 \geq \mu_2$. Plugging the two tuples into the first stationary constraint we obtain

$$Sz_1 = \mu_1 z_1 + b\,, \tag{2.17}$$

$$Sz_2 = \mu_2 z_2 + b\,. \tag{2.18}$$

Substracting (2.18) from (2.17) we have

$$Sz_1 - Sz_2 = \mu_1 z_1 - \mu_2 z_2\,. \tag{2.19}$$

Multiplying Eq. (2.19) first with $z_1^\top$ and then with $z_2^\top$ and adding the resulting two equations (having in mind that the matrix S is symmetric) we derive

$$z_1^\top Sz_1 - z_2^\top Sz_2 = (\mu_1 - \mu_2)(R^2 + z_1^\top z_2)\,. \tag{2.20}$$

On the other hand, using the Cauchy-Schwarz inequality and (2.16) we deduce

$$z_1^\top z_2 \leq \|z_1\|\|z_2\| = R^2\,. \tag{2.21}$$

Now, combining the results obtained in (2.20) and (2.21) with the initial assumption $\mu_1 \geq \mu_2$,

$$z_1^\top Sz_1 - z_2^\top Sz_2 \leq 2R^2(\mu_1 - \mu_2)\,. \tag{2.22}$$

Finally, subtracting the optimization objectives for the two tuples and using (2.17) and (2.18) multiplied by $z_1^\top$ and $z_2^\top$, respectively, we show that

$$\Theta(z_1) - \Theta(z_2) = 2R^2(\mu_1 - \mu_2) - (z_1^\top Sz_1 - z_2^\top Sz_2) \geq 0\,,$$

where the last inequality follows from (2.22). □

Hence, instead of the original optimization problem (2.12) we can solve the system with two stationary equations (2.16) and minimal μ. Gander et al. (1989) propose two methods for solving such problems. In the first approach, the problem is reduced to a quadratic eigenvalue problem and afterwards transformed into a linear eigenvalue problem. In the second approach the problem is reduced to solving a one-dimensional secular equation. The first approach is more elegant, as it allows us to compute the solution in a closed form. Namely, the solution to the problem (2.12) is given by (Gander et al., 1989)

$$z^* = (S - \mu_{\min}\mathbb{I})^{-1}\, b\ ,$$

where $\mu_{\min}$ is the smallest real eigenvalue of

$$\begin{bmatrix} S & -\mathbb{I} \\ -\frac{1}{R^2} b b^\top & S \end{bmatrix} .$$

Despite its elegance, the approach requires us to decompose a non-symmetric block matrix of dimension $2n$ and this is not a numerically stable task for every such matrix. Furthermore, the computed solution z^* highly depends on the precision up to which the optimal μ is computed and for an imprecise value the solution might not be on the hypersphere at all (for a detailed study refer to Gander et al., 1989). For this reason, we rely on the secular approach in the computation of the optimal solution. In the next section, we present an efficient algorithm (Gander et al., 1989) for the computation of the optimal Lagrange multiplier to machine precision and here we describe how to derive the secular equation required to compute the multiplier. First, the stationary constraint from Eq. (2.16) is simplified by decomposing the symmetric matrix $S = U\Sigma U^\top$ as

$$U\Sigma U^\top z = b + \mu z\ .$$

Then, the resulting equation is multiplied with the orthogonal matrix $U^\top$ from the left and transformed into

$$\Sigma\tilde{z} = \tilde{b} + \mu\tilde{z} \quad \text{with} \quad \tilde{b} = U^\top b \ \wedge \ \tilde{z} = U^\top z\ .$$

From the latter equation we compute

$$\tilde{z}_i(\mu) = \frac{\tilde{b}_i}{\sigma_i - \mu} \quad (i = 1, 2, ..., n)\ ,$$

and substitute the computed vector $\tilde{z}(\mu)$ into the second stationary constraint to form the secular equation

$$g(\mu) = \sum_{i=1}^{n} \tilde{z}_i^2(\mu) - R^2 = \sum_{i=1}^{n} \frac{\tilde{b}_i^2}{(\sigma_i - \mu)^2} - R^2 = 0\ . \tag{2.23}$$

The optimal value of parameter μ is the smallest root of the non-linear secular equation and the optimal solution to problem (2.12) is given by

$$z^* = U \cdot \tilde{z}(\mu_{\min})\ .$$

Moreover, the interval at which the root lies is known (Gander et al., 1989). In particular, it must hold $\mu_{\min} \leq \sigma_n \leq \sigma_{n-1} \leq \ldots \sigma_1 \leq \mu_{\max}$, where $\{\sigma_i\}_{i=1}^{n}$ are the eigenvalues of matrix S. To see this, suppose that $\sigma_n \neq 0$ and observe that the derivative of the secular function

$g'(\mu) > 0$ for $\mu \in (-\infty, \sigma_n)$. From here it then follows that the secular function is monotone increasing on the interval $(-\infty, \sigma_n)$. Thus, if the secular function has a root in that interval then the root is unique. That such a root exists follows from the fact that the secular function changes sign on $(-\infty, \sigma_n)$, i.e., $\lim_{\mu \to -\infty} g(\mu) = -R^2 < 0$ and $\lim_{\mu \to \sigma_n^-} g(\mu) = +\infty$.

The complexity of both approaches (secular and eigenvalue) for an r-dimensional embedding is $O(rn^3)$. The cubic term arises from the eigendecompositions required to compute the solutions to problem (2.12) for each of the r knowledge-based kernel principal components.

2.4.3 Secular Equation

We review here an effective iterative method (Gander et al., 1989) for finding the smallest/largest root of the secular equation in Eq. (2.23). An obvious choice for the root finder is the Newton method and, yet, it is not well suited for the problem. The tangent at certain points in the interval of interest crosses the x-axis outside that interval leading to incorrect solution or division by zero. An efficient root finder, then, must overcome these issues and converge very quickly. The main idea behind the efficient iterative root finder is to first approximate the secular equation with a quadratic surrogate and then update the current root estimate with the root of the surrogate function.

As the smallest root $\mu_{\min} \in (-\infty, \sigma_n)$, the secular equation (2.23) has a quadratic surrogate for only one side of the interval (Gander et al., 1989), i.e.,

$$h_t(\mu) = \frac{p_t}{(q_t - \mu)^2} - R^2 ,$$

where $p_t, q_t \in \mathbb{R}$ and $h_t(\mu)$ is a quadratic surrogate function of the secular equation $g(\mu)$. In order to determine the coefficients of the surrogate at the step t, the secular equation and its derivative are matched to the corresponding surrogate approximations at the candidate root. More formally, the following constraints are enforced on the surrogate function

$$h_t(\mu_t) = g(\mu_t) \quad \wedge \quad h_t'(\mu_t) = g'(\mu_t) ,$$

where $\{\mu_t\}_{t \geq 0}$ is a sequence of iterative approximations of $\mu_{\min}$ (defined subsequently). From this constraint on the derivate of the two functions it follows that

$$g'(\mu_t) = 2\frac{g(\mu_t) + R^2}{q_t - \mu_t} \quad \Longrightarrow \quad q_t = \mu_t + 2\frac{g(\mu_t) + R^2}{g'(\mu_t)} .$$

Now, combining the computed coefficient q_t with the constraint on the surrogate value at μ_t we obtain the second coefficient

$$p_t = 4\frac{\left(g(\mu_t) + R^2\right)^3}{g'(\mu_t)^2} .$$

Having computed the coefficients p_t and q_t the sequence $\{\mu_t\}_{t \geq 0}$ is given by

$$\frac{p_t}{(q_t - \mu_{t+1})^2} - R^2 = 0 \quad \Longrightarrow \quad \mu_{t+1} = q_t - \frac{\sqrt{p_t}}{R} = \mu_t + 2\frac{g(\mu_t) + R^2}{g'(\mu_t)}\left(1 - \frac{\sqrt{g(\mu_t) + R^2}}{R}\right) .$$

For an initial solution that satisfies $\mu_{\min} < \mu_0 < \sigma_n$, the convergence is monotonic (Bunch et al., 1978), i.e., $\mu_{\min} < \mu_{t+1} < \mu_t$ for all $t > 0$.

2.5 Large Scale Approximations

In this section, we address the problem of extracting knowledge-based kernel principal components from large scale datasets with millions of instances. The approaches described in the previous section are not suitable for such problems because of their computational complexity. In particular, both of the presented approaches have cubic runtime complexity in the size of the kernel expansion and do not scale to problems with millions of instances. We propose two approaches to overcome this shortcoming of knowledge-based kernel principal component analysis. The approaches are motivated by the fact that frequently used kernel matrices have a fast decaying spectrum and that small eigenvalues can be removed without a significant effect on the precision (Schölkopf and Smola, 2002). In the first approach (presented in Section 2.5.1), we propose to iteratively solve the optimization problem from Eq. (2.12) using the conjugate gradient descent method (Golub and van Loan, 1996). An iteration of the approach has quadratic computational complexity in the number of instances and for low-rank matrices it is possible to obtain a good approximation of the optimal solution with a small number of such iterations. In the second approach (presented in Section 2.5.2), we first find an approximate low-rank factorization of the kernel matrix and then derive knowledge-based kernel principal components with that matrix in place of the original kernel matrix. The approach has the computational complexity linear in the number of instances and can scale knowledge-based kernel principal component analysis to millions of instances.

2.5.1 Iterative Optimization of a Quadratic Form over a Hypersphere

In this section, we build on the work by Golub and van Loan (1996) an propose an iterative approach for solving the optimization problem in Eq. (2.12) that has the quadratic runtime cost per iteration (with respect to the number of instances). The approach is based on the conjugate gradient descent method (Golub and van Loan, 1996) for solving linear systems of equations defined with symmetric and positive definite matrices. First, we describe (in Section 2.5.1.1) a procedure for an approximate computation of the smallest value of the Lagrange multiplier satisfying the stationary constraints from Eq. (2.16). The procedure is based on the conjugate gradient descent method and has the quadratic runtime cost in the number of instances. For the optimal value of the Lagrange multiplier, the optimal solution to problem in Eq. (2.12) is the solution of the following linear system

$$(S - \mu_{\min}\mathbb{I})\, z = b \ . \tag{2.24}$$

As discussed in Section 2.4, the matrix S is symmetric and $\mu_{\min} < \sigma_n \leq \sigma_{n-1} \leq \cdots \leq \sigma_1$. From here it then follows that the matrix $P = (S - \mu_{\min}\mathbb{I})$ is symmetric and positive definite. Hence, we can apply the conjugate gradient descent method (Section 10.2, Golub and van Loan, 1996) to iteratively solve this system with the quadratic cost per iteration. In Section 2.5.1.2, we provide a brief review of this method and present a theoretical guarantee on the quality of the solution obtained in this way. In our review of the approach, we follow closely the exposition by Golub and van Loan (Chapter 10, 1996).

2.5.1.1 Iterative Computation of the Lagrange Multiplier

In this section, we propose a mean to approximate the optimal Lagrange multiplier (defining the linear system in Eq. 2.24) in large scale problems. In order to compute the multiplier, we first need to derive the open interval containing this root of the secular equation. As shown in Section 2.4.2, the optimal multiplier lies in the open interval determined by the smallest eigenvalue of the matrix S. To obtain the smallest eigenvalue of the matrix S, we propose to use the power iteration algorithm (Golub and van Loan, 1996) which has the quadratic runtime cost per iteration. However, as we need the smallest eigenvalue and the power iteration algorithm computes the largest one, we apply the algorithm to the matrix $-S$.

Having computed the smallest eigenvalue of the matrix S, we have determined the interval of the secular root corresponding to the optimal Lagrange multiplier. In order to compute this multiplier we form a slightly different version of the secular equation,

$$g(\mu) = z^\top (S - \mu\mathbb{I})^{-2} z - R^2 .$$

In our empirical evaluations (Section 2.9), the iterative algorithm described in Section 2.4.3 proved to be very fast and always converged in few iterations to machine precision. To apply this algorithm with the conjugate gradient descent method and without an eigendecomposition of S, we need to be able to derive the coefficients, p_t and q_t ($t > 0$), of the surrogate quadratic function (see Section 2.4.3). For this, we need to be able to evaluate the secular equation and its derivative at any iteration. The first is simple to achieve using the conjugate gradient descent algorithm from the previous section. In particular, for the derivative of the secular equation at an estimate μ_t of $\mu_{\min}$ we have

$$g'(\mu_t) = 2z_{\mu_t}^\top (S - \mu_t\mathbb{I})^{-1} z_{\mu_t} ,$$

where z_{μ_t} is the solution of the linear system $P_{\mu_t} z = b$ with $P_{\mu_t} = S - \mu_t\mathbb{I}$, obtained using the conjugate gradient descent method. Thus, by applying the conjugate gradient descent method one more time to solve the linear system $P_{\mu_t}\hat{z} = z_{\mu_t}$, one obtains the gradient of the secular equation at μ_t. The described procedure has quadratic runtime complexity stemming from the cost per iteration of the conjugate gradient descent method. Hence, for low-rank kernel matrices (or matrices with a fast decaying spectrum) we can use this approach to compute an approximation of the optimal multiplier for problem (2.12) in $\mathcal{O}(n^2)$ time.

2.5.1.2 Conjugate Gradient Descent

This section reviews the conjugate gradient descent approach (Chapter 10, Golub and van Loan, 1996) in the context of Section 2.4 and the optimization problem in Eq. (2.24). The approach is based on the observation that solving the linear system, $Pz = b$, is equivalent to minimizing the quadratic form

$$\Phi(z) = \frac{1}{2} z^\top P z - b^\top z .$$

The fact that P is a symmetric and positive definite matrix implies that the minimal value of $\Phi(z)$ is attained by setting $z = P^{-1}b$. Thus, the simplest iterative method for solving the linear system in Eq. (2.24) is the gradient descent approach. The negative gradient of the quadratic form at the step t is given by the residual at that step, i.e.,

$$r_t = b - Pz_t = -\nabla\Phi(z_t) .$$

If the residual vector is non-zero then there exists a positive constant $\tau \in \mathbb{R}^+$ such that $z_{t+1} = z_t + \tau r_t$ and $\Phi(z_{t+1}) < \Phi(z_t)$. While simple and easy to implement, the gradient descent method can be inefficient when the condition number $\kappa(P) = {}^{\sigma_1 - \mu_{\min}}/_{\sigma_n - \mu_{\min}}$ is large.

To avoid this issue, the conjugate gradient descent method minimizes the quadratic form $\Phi(z)$ along a set of linearly independent directions $\{g_i\}_{i=1}^t$ that do not necessarily correspond to residuals $\{r_i\}_{i=1}^{t}$, with $t = 1, 2, \ldots, n$. The convergence is guaranteed in at most n steps because that is the dimension of the problem and a solution can be written as a linear combination of at most n linearly independent vectors. Similar to Golub and van Loan (1996), let us first consider the choice of a direction g_t. For this purpose, let us now take (we subsequently show that this can always be done)

$$z_t = z_0 + G_{t-1}\xi + \tau g_t \ ,$$

where G_{t-1} is a matrix with columns $\{g_i\}_{i=1}^{t-1}$, $\xi \in \mathbb{R}^{t-1}$, and $\tau \in \mathbb{R}$. Then, we have that

$$\begin{aligned}\Phi(z_t) = \ & \Phi(z_0 + G_{t-1}\xi + \tau g_t) = \\ & \Phi(z_0 + G_{t-1}\xi) + \tau\xi^\top G_{t-1}^\top P g_t + \frac{\tau^2}{2} g_t^\top P g_t + \tau g_t^\top (P z_0 - b) = \\ & \Phi(z_0 + G_{t-1}\xi) + \tau\xi^\top G_{t-1}^\top P g_t + \frac{\tau^2}{2} g_t^\top P g_t - \tau g_t^\top r_0 \ .\end{aligned}$$

If $g_t \perp \mathrm{span}(\{Pg_1, \ldots, Pg_{t-1}\})$ then $\xi^\top G_{t-1}^\top P g_t = 0$ and the search for z_t splits into two independent optimization problems,

$$\begin{aligned}\min_{z \ \in \ z_0 + \mathrm{span}(\{g_1, \ldots, g_t\})} \Phi(z) = \ & \min_{\xi \in \mathbb{R}^{t-1}, \ \tau \in \mathbb{R}} \Phi(z_0 + G_{t-1}\xi + \tau g_t) = \\ & \operatorname*{argmin}_{\xi \in \mathbb{R}^{t-1}, \ \tau \in \mathbb{R}} \Phi(z_0 + G_{t-1}\xi) + \frac{\tau^2}{2} g_t^\top P g_t - \tau g_t^\top r_0 = \\ & \min_{\xi \in \mathbb{R}^{t-1}} \Phi(z_0 + G_{t-1}\xi) + \min_{\tau \in \mathbb{R}} \left(\frac{\tau^2}{2} g_t^\top P g_t - \tau g_t^\top r_0 \right) .\end{aligned}$$

From here it then follows that the solution to the first optimization problem minimizes the quadratic form over $z_0 + \mathrm{span}(\{g_1, \ldots, g_{t-1}\})$. On the other hand, the optimal solution to the second problem is $\tau_t = \frac{g_t^\top r_0}{g_t^\top P g_t}$. Moreover, the fact that $g_t \perp \mathrm{span}(\{Pg_1, \ldots, Pg_{t-1}\})$ implies

$$g_t^\top r_{t-1} = -g_t^\top (P z_{t-1} - b) = -g_t^\top (P z_0 + P G_{t-1}\xi - b) = g_t^\top r_0 \ .$$

Thus, direction g_t should be chosen so that $g_t \perp \mathrm{span}\{Pg_1, \ldots, Pg_{t-1}\}$ and $g_t^\top r_{t-1} \neq 0$. In Golub and van Loan (Section 10.2, 1996), the authors show that such conjugate directions can always be selected by setting

$$g_t = r_{t-1} + \pi_t g_{t-1} \ .$$

Multiplying the latter equation with $g_{t-1}^\top P$ from the left and using the fact that the vectors Pg_{t-1} and g_t are mutually orthogonal we obtain that

$$\pi_t = -\frac{g_{t-1}^\top P r_{t-1}}{g_{t-1}^\top P g_{t-1}} \ .$$

Hence, the conjugate gradient descent can be performed by setting

$$z_t = z_{t-1} + \tau_t g_t = z_{t-1} + \frac{g_t^\top r_0}{g_t^\top P g_t}(r_{t-1} + \pi_t g_{t-1}) = z_{t-1} + \frac{g_t^\top r_{t-1}}{g_t^\top P g_t}\left(r_{t-1} - \frac{g_{t-1}^\top P r_{t-1}}{g_{t-1}^\top P g_{t-1}} g_{t-1} \right) .$$

The conjugate gradient descent iteration in this form requires three matrix-vector multiplications. This is computationally inefficient and it can be improved by observing that

$$r_t = b - P z_t = b - P z_{t-1} - \tau_t P g_t = r_{t-1} - \tau_t P g_t \ .$$

From here it then follows that

$$\|r_{t-1}\|^2 = r_{t-1}^\top r_{t-1} = r_{t-1}^\top r_{t-2} - \tau_{t-1} r_{t-1}^\top P g_{t-1} \ .$$

Noting that $r_{t-1}^\top r_{t-2} = 0$ (e.g., see Theorem 10.2.3 in Golub and van Loan, 1996) we get

$$\|r_{t-1}\|^2 = -\tau_{t-1} r_{t-1}^\top P g_{t-1} \ .$$

On the other hand, from the definition of τ_{t-1} it follows that

$$g_{t-1}^\top r_{t-2} = g_{t-1}^\top r_0 = \tau_{t-1} g_{t-1}^\top P g_{t-1} \ .$$

The latter expression implies that we can express π_t as

$$\pi_t = \frac{\|r_{t-1}\|^2}{g_{t-1}^\top r_{t-2}} \ .$$

Hence, we can now give a conjugate gradient descent iteration that requires only one matrix-vector multiplication,

$$z_t = z_{t-1} + \frac{g_t^\top r_{t-1}}{g_t^\top P g_t} \left(r_{t-1} + \frac{\|r_{t-1}\|^2}{g_{t-1}^\top r_{t-2}} g_{t-1} \right) \ .$$

Having given an iterative solution that requires a single matrix-vector multiplication and, thus, has the quadratic runtime cost per iteration, we now review the theoretical properties of the method. First, we present a worst case bound on the approximation error of the approach expressed in terms of the number of iterations and condition number of the matrix defining the linear system in Eq. (2.24).

Theorem 2.2. *(Luenberger, 1973) Assume $P \in \mathbb{R}^{n \times n}$ is a symmetric and positive definite matrix and $b \in \mathbb{R}^n$. If the conjugate gradient descent method produces iterates $\{z_i\}$ and $\kappa = \kappa(P)$ then*

$$\|z^* - z_t\|_P \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^t \|z^* - z_0\|_P \ ,$$

where $z^* = P^{-1} b$ and $\|z\|_P^2 = z^\top P z$.

Corollary 2.3. *The approximation error of the conjugate gradient descent method satisfies*

$$\left\| z_t - P^{-1} b \right\| \leq 2\sqrt{\kappa} \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^t \left\| z_0 - P^{-1} b \right\|.$$

Proof. This corollary is formulated as a self-study problem in Golub and van Loan (Problem 10.2.8, 1996). In order to show this claim, let us first observe that

$$\left\| z_t - P^{-1} b \right\|_P^2 = \left(z_t - P^{-1} b \right)^\top P \left(z_t - P^{-1} b \right) = \left\| P^{1/2} \left(z_t - P^{-1} b \right) \right\|^2 \ .$$

For the resulting expression, using the properties of the operator norm, we obtain

$$\sqrt{\sigma_n - \mu_{\min}} \left\| z_t - P^{-1} b \right\| \leq \left\| P^{1/2} \left(z_t - P^{-1} b \right) \right\| \leq \sqrt{\sigma_1 - \mu_{\min}} \left\| z_t - P^{-1} b \right\| \ .$$

Hence, from Theorem 2.2 and the latter inequality it follows that

$$\sqrt{\sigma_n - \mu_{\min}} \left\| z_t - P^{-1} b \right\| \leq \quad \|z^* - z_t\|_P \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^t \|z^* - z_0\|_P \leq$$

$$2\sqrt{\sigma_1 - \mu_{\min}} \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^t \left\| z_0 - P^{-1} b \right\| \ .$$

The result follows after dividing the latter inequality by $\sqrt{\sigma_n - \mu_{\min}}$. □

From these two bounds, we conclude that the conjugate gradient descent method converges fast, i.e., in a small number iterations, for well-conditioned matrices. Thus, for knowledge-based kernel principal component analysis with a well-conditioned matrix P the approach can provide an efficient approximation of the optimal solution for the optimization of a quadratic form over a hypersphere of constant radius (described in Section 2.4). Beside these two results, Golub and van Loan (1996) give an upper bound on the number of required iterations for matrices that can be written as a sum of the identity and a low-rank matrix. The following theorem states that result more formally.

Theorem 2.4. *(Golub and van Loan, 1996) Assume that* $P = \mathbb{I} + \overline{P} \in \mathbb{R}^{n \times n}$ *is a symmetric and positive definite matrix and* $\operatorname{rank}\left(\overline{P}\right) = r$*. Then, the conjugate gradient descent method converges in at most* $r + 1$ *steps.*

Thus, for low-rank kernel matrices the conjugate gradient descent method can provide an effective approximation of the optimal solution defining the knowledge-based kernel principal components. Having reviewed this approach and theoretical results giving insights into its effectiveness, we proceed to the next section where we derive knowledge-based kernel principal components using an approximate low-rank factorization of a kernel matrix.

2.5.2 Low-Rank Approximations

In this section, we propose an alternative approach for the derivation of knowledge-based kernel principal components in large scale problems compared to the approach presented in Section 2.5.1. The main idea behind this approach is to substitute the full kernel matrix with an approximate low-rank factorization and adapt the techniques presented in Section 2.4 to account for the low-rank approximation. More specifically, we propose to use a matrix $\overline{K} = \Psi^\top \Psi$ with $\Psi \in \mathbb{R}^{l \times n}$ such that for all $\varepsilon > 0$ there exists $l \leq n$ so that

$$\left\| K - \Psi^\top \Psi \right\|_p < \varepsilon \ ,$$

where $\|\cdot\|_p$ denotes the Schatten p-norm of a symmetric and positive definite matrix (Weidmann, 1980). Typically, the rank of the approximation $l \ll n$ and this enables us to find the approximate knowledge-based kernel principal components using the closed form solvers in time $\mathcal{O}(l^3)$. This is a significant speed-up compared to the runtime cost of $\mathcal{O}(n^3)$ for the optimization problem in Eq. (2.11) defined with the full kernel matrix.

For the moment, suppose that the kernel matrix K can be approximated with a low-rank factorization $\Psi^\top \Psi$. Then, the optimization problem from Eq. (2.12) can be written as

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \quad & z^\top \Psi^\top \Psi \left(\left(\Psi^\top \Psi \right)^{-1} + E \right) \Psi^\top \Psi z - 2 b^\top \Psi^\top \Psi z \\ s.t. \quad & z^\top \left(\Psi^\top \Psi \right)^2 z = R^2 \ , \end{aligned} \tag{2.25}$$

where $E = S - K^{-1}$. The fact that the matrix Ψ is of rank $l \ll n$ implies that the inverse matrix $(\Psi^\top \Psi)^{-1} \in \mathbb{R}^{n \times n}$ is also of rank l. To see this, let us perform a singular value decomposition of matrix $\Psi = U \Pi V^\top$, where $U \in \mathbb{R}^{l \times l}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices, and $\Pi \in \mathbb{R}^{l \times n}$ is a diagonal matrix with at most l positive singular values. From the decomposition it follows that $\Psi^\top \Psi = V \Pi^2 V^\top$. If we denote with $\Pi_l \in \mathbb{R}^{l \times l}$ the diagonal matrix with l non-zero singular values then the inverse matrix $(\Psi^\top \Psi)^{-1} = V_l \Pi_l^{-2} V_l^\top$, where V_l denotes the right singular vectors corresponding to non-zero singular values in Π_l. The fact that the singular value matrix Π is of rank l also implies that in $(\Psi^\top \Psi)^2 = V_l \Pi_l^4 V_l^\top$ there is no dependence on the right singular vectors corresponding to zero singular values. Hence, substituting $\overline{z} = \Pi_l^2 V_l^\top z \in \mathbb{R}^l$ into Eq. (2.25) we obtain the optimization problem for the low-rank approximation of knowledge-based kernel principal components,

$$\begin{aligned} \min_{\overline{z} \in \mathbb{R}^l} \quad & \overline{z}^\top \left(\Pi_l^{-2} + V_l^\top E V_l \right) \overline{z} - 2 \left(V_l^\top b \right)^\top \overline{z} \\ s.t. \quad & \overline{z}^\top \overline{z} = R^2 \ . \end{aligned} \tag{2.26}$$

In the latter problem, $\Pi_l^{-2} + V_l^\top E V_l \in \mathbb{R}^{l \times l}$ is a symmetric matrix that can be computed in time $\mathcal{O}(l^3 + l^2 n)$, where $\mathcal{O}(l^3)$ stems from the singular value decomposition of matrix Ψ and $\mathcal{O}(l^2 n)$ from the matrix-matrix multiplications in $V_l^\top E V_l$. The latter computational cost is not cubic because the matrices comprising E are either diagonal or very sparse (e.g., see Eq. 2.12). Hence, a closed form solution for the problem in Eq. (2.26) can be computed in time $\mathcal{O}(l^3 + l^2 n)$ using the approaches from Section 2.4. As $l \ll n$ the approach can scale knowledge-based kernel principal component analysis to millions of instances.

Having described the optimization problem for the computation of low-rank approximations to knowledge-based kernel principal components, we now review two standard approaches for obtaining a good low-rank factorization of the kernel matrix. While the approach reviewed in Section 2.5.2.1 is suitable for any kernel function, the one reviewed in Section 2.5.2.2 works only for the class of stationary kernels (e.g., see Chapter 3).

2.5.2.1 Nyström Method

The section provides a brief review of the Nyström method (Nyström, 1930; Williams and Seeger, 2001) for low-rank approximation of kernel matrices. The method will be investigated in more details in Chapter 4, where an approximation bound will also be given. The presentation in this section follows closely that of Williams and Seeger (2001), where the approach was first introduced for the purpose of low-rank approximation of kernel matrices.

The Nyström method computes a low-rank approximation $\overline{K}$ of a kernel matrix K by first sampling (without replacement) l instances from X. The literature often refers to these selected instances as landmarks. If we denote with $K_{l,l}$ the block in the kernel matrix corresponding to kernel function values between the landmarks and with $K_{n,l}$ the block with kernel values between all available instances and the landmarks, then the Nyström approximation is given by

$$\overline{K} = K_{n,l} K_{l,l}^{-1} K_{l,n} \ .$$

Now, from the eigendecomposition of the symmetric and positive definite matrix $K_{l,l} = V_{l,l} \Sigma_{l,l}^2 V_{l,l}^\top$, we obtain that the low-rank approximation can be written as

$$\overline{K} = K_{n,l} V_{l,l} \Sigma_{l,l}^{-1} \left(K_{n,l} V_{l,l} \Sigma_{l,l}^{-1} \right)^\top = \Psi^\top \Psi \ ,$$

where $\Psi = \left(K_{n,l} V_{l,l} \Sigma_{l,l}^{-1} \right)^\top$. In order to express a particular instance $x_i \in X$ in this feature representation, one first needs to compute the column vector, K_i, with kernel values between that instance and landmarks. Then, the instance x_i can be represented as $K_i^\top V_{l,l} \Sigma_{l,l}^{-1}$.

The computational complexity of the Nyström method is $\mathcal{O}\left(l^3 + l^2 n\right)$ and the fact that $l \ll n$ implies that the method is capable of alleviating the cubic complexity of our approach. If we denote with V_l and Σ_l matrices with the top l eigenvectors and eigenvalues of the kernel matrix $K = V \Sigma V^\top$, then the optimal approximation of the kernel matrix (measured in the Schatten p-norm) is obtained if the landmarks can be selected such that $\overline{K} = V_l \Sigma_l V_l^\top$.

2.5.2.2 Random Fourier Features

In this section, we provide a brief overview of the random Fourier features method for the approximation of stationary kernel functions. A more detailed review of this approach is provided in Chapter 3. Before we give a low-rank approximation of the kernel matrix using random Fourier features, we define the class of shift-invariant/stationary kernel functions.

Definition 2.1. *Let $D \subset \mathbb{R}^d$ be an open set. A positive definite kernel $k\colon D \times D \to \mathbb{R}$ is called stationary or shift-invariant if there exists a function $s\colon D \to \mathbb{R}$ such that $k(x,y) = s(x-y)$, for all $x, y \in D$. The function s is said to be a function of positive type.*

Having defined the class of stationary kernels, let us now review the key theoretical result for the approximation of kernel functions using random Fourier features.

Theorem 2.5. *(Bochner, 1932) The Fourier transform of a bounded positive measure on $\mathbb{R}^d$ is a continuous function of positive type. Conversely, any function of positive type is the Fourier transform of a bounded positive measure.*

From this theorem it follows that for a stationary kernel k it holds

$$k(x,y) = s(x-y) = \int_{\mathbb{R}^d} \exp\left(-i\langle w, x-y\rangle\right) d\mu(w) \ ,$$

where μ is a positive and bounded measure. As $k(x,y)$ is a real function in both arguments, the complex part in the integral on the right hand-side is equal to zero, and we have

$$k(x,y) = 2\int \cos\left(w^\top x + b\right)\cos\left(w^\top y + b\right) d\hat{\mu}(w,b) \ ,$$

where $\hat{\mu}(w,b) = \frac{\mu(w)}{2\pi} > 0$ for all $w \in \mathbb{R}^d$ and $b \in [-\pi, \pi]$. Hence, it is possible to sample (w, b) proportional to $\hat{\mu}(w,b)$ and approximate the kernel value at (x,y) by the Monte-Carlo estimate of the integral defining the inner product between two instances. The first kernel approximation algorithm based on this idea was proposed by Rahimi and Recht (2008a). That work gives an approximation of a stationary kernel using l random Fourier features by

$$\overline{k}(x,y) = \frac{2}{l}\sum_{i=1}^{l} \cos\left(w_i^\top x + b_i\right)\cos\left(w_i^\top y + b_i\right) \ ,$$

where $\{(w_i, b_i)\}_{i=1}^{l}$ are independent samples from the probability distribution that is proportional to the measure $\hat{\mu}(w,b)$. The convergence of the approximation to the actual value of the kernel function at a given pair of instances follows from the Hoeffding's concentration inequality (e.g., see Chapter 3 for more details). Hence, if we denote with $\psi_l(x) = \operatorname{vec}\left\{\sqrt{2/l}\cos\left(w_1^\top x + b_1\right), \ldots, \sqrt{2/l}\cos\left(w_l^\top x + b_l\right)\right\}$, the approximation of the kernel function at (x,y) can be written as

$$\overline{k}(x,y) = \psi_l(x)^\top \psi_l(y) \ .$$

From here it then follows that the approximation of the kernel matrix can be written as

$$\overline{K} = \psi_l(X)^\top \psi_l(X) \ ,$$

where x_i denotes the i-th column in the data matrix X $(1 \le i \le n)$ and $\psi_l(X) \in \mathbb{R}^{l \times n}$ is the matrix with random Fourier features, $\psi_l(x_i)$, as columns.

2.6 Interactive Data Visualization

Having introduced means for incorporating domain knowledge constraints into kernel principal component analysis (Section 2.3), we now propose an efficient algorithm for interaction with a data visualization generated using the knowledge-based kernel principal components. To shape such an embedding interactively with the help of knowledge-based constraints it is required to solve the optimization problem in Eq. (2.11) at each interaction step. In Section 2.4, we have described how to solve the arising optimization problem in a closed form with runtime complexity $\mathcal{O}(rn^3)$, where r is the number of embedding directions. In this section, we show how the proposed algorithm can be adapted to enable user interaction in $\mathcal{O}(r^2n^2)$ time. For low-rank approximations (Section 2.5.2), the runtime complexity of computing a closed form solution is $\mathcal{O}(rl^3)$ with $l \ll n$, and for such data embeddings the interaction can be performed in $\mathcal{O}(r^2l^2)$ time. In order to achieve this speed-up in runtime, we express the interaction in terms of rank-one updates of the original problem (Section 2.6.1) and review a linear time algorithm for solving the arising secular equations (Section 2.6.2).

2.6.1 Efficient Formulation of Interaction

Each interaction step consists of moving a single control point. In particular, either one selects a new control point or updates the position of an existing one. To compute such an embedding interactively the algorithm needs to solve a variant of the problem in Eq. (2.12) for different interaction steps and for all r directions. Let us assume, without loss of generality, that the algorithm is required to find r knowledge-based kernel principal components defined with only soft control point placements and soft orthogonality. We denote with $S_{t,s}$ the symmetric matrix defining the quadratic term in the problem arising for a step t and a direction s. The linear term corresponding to the s-th direction in the step t is a function of a block of the kernel matrix and control point placements y_t. We denote such a linear term with $b_{t,s}$ and write the first stationary constraint for the s-th direction in the step t as

$$S_{t,s}z = \mu z + b_{t,s} \ .$$

For a knowledge-based principal component corresponding to $s > 1$ and $t > 1$, the matrix $S_{t,s}$ is a rank-one update of the matrix $S_{t,s-1}$. To see this, observe that the soft orthogonality term is the only difference between these two matrices. Otherwise, for $s = 1$ and $t > 1$ either a new control point is selected and the matrix $S_{t,s}$ is a rank-one update of the matrix $S_{t-1,s}$ or the position of an existing control point is updated and $S_{t,s} = S_{t-1,s}$. For simplicity of our derivation, we can without loss of generality assume that a new control point has been added in the step $t > 1$ and focus on finding the first projection direction. The derivation for the update of an existing control point differs only in that $S_{t,1} = S_{t-1,1}$.

As we have already solved the optimization problem for the step $t-1$ and $s = 1$, we can reuse the eigendecomposition of $S_{t-1,s} = U_{t-1,s}\Sigma_{t-1,s}U_{t-1,s}^{\top}$ and express the matrix $S_{t,s}$ as

$$S_{t,s} = S_{t-1,s} - \tau a_{t,s}a_{t,s}^{\top} = U_{t-1,s}\left(\Sigma_{t-1,s} - \tau \bar{a}_{t,s}\bar{a}_{t,s}^{\top}\right)U_{t-1,s}^{\top} \ ,$$

where $a_{t,s}$ is the rank-one update of the quadratic term from Eq. (2.12) corresponding to an addition of a control point and $\bar{a}_{t,s} = U_{t-1,s}^{\top}a_{t,s}$. Let us denote the rank-one update of the diagonal matrix $\Sigma_{t-1,s}$ as

$$\Theta_{t,s} = \Sigma_{t-1,s} - \tau \bar{a}_{t,s}\bar{a}_{t,s}^{T} \ . \tag{2.27}$$

The computational complexity of a complete eigendecomposition (e.g., see Bunch et al., 1978; Arbenz, 2012) of the matrix $\Theta_{t,s}$ is $O(n^2)$. The decomposition can be computed by solving n secular equations (see Section 2.6.2), one for each of the eigenvalues of $\Theta_{t,s}$. Rewriting the first stationary constraint using the substitution we get

$$U_{t-1,s}\Theta_{t,s}U_{t-1,s}^T z = \mu z + b_{t,s} \quad \Longrightarrow \quad \Theta_{t,s}\overline{z} = \mu\overline{z} + \overline{b}_{t,s} \ ,$$

where $\overline{z} = U_{t-1,s}^T z$ and $\overline{b}_{t,s} = U_{t-1,s}^T b_{t,s}$. Now, using the eigendecomposition $\Theta_{t,s} = V_{t,s}\Sigma'_{t,s}V_{t,s}^T$ we transform the latter problem into

$$\Sigma'_{t,s}\tilde{z}_{t,s} = \mu\tilde{z}_{t,s} + \tilde{b}_{t,s} \ , \tag{2.28}$$

where $\tilde{z}_{t,s} = V_{t,s}^T\overline{z}$ and $\tilde{b}_{t,s} = V_{t,s}^T\overline{b}_{t,s}$. The second stationary constraint combined with Eq. (2.28) gives the secular solution $\tilde{z}_{t,s}(\mu_{\text{min}})$, similar to the one from Eq. (2.23). Hence, for $t > 1$ and $s = 1$ the knowledge-based kernel principal component $z_{t,s}$ is given by

$$z_{t,s} = U_{t-1,s}V_{t,s}\tilde{z}_{t,s}(\mu_{\text{min}}) \ ,$$

where $\tilde{z}_{t,s}(\mu)$ is a vector with the i-th component given by

$$\tilde{z}_{t,s,i}(\mu) = \frac{U_{t-1,s}V_{t,s}b_{t,s}}{\sigma'_{t,s,i} - \mu} \ .$$

Here, $\sigma'_{t,s,i}$ denotes an eigenvalue of $\Theta_{t,s}$ and $i = 1, \ldots, n$.

For the interaction in a step $t > 1$ and for a component $s > 1$, the matrix $S_{t,s}$ is a rank-one update of $S_{t,s-1}$. Thus, the knowledge-based kernel principal component $z_{t,s}$ is given by

$$z_{t,s} = U_{t,s-1}V_{t,s}\tilde{z}_{t,s}(\mu_{\text{min}}) = U_{t-1,s}\left(\prod_{i=1}^{s} V_{t,i}\right)\tilde{z}_{t,s}(\mu_{\text{min}}) \ ,$$

where $\tilde{z}_{t,s}(\mu)$ is a vector with components $(1 \le i \le n)$

$$\tilde{z}_{t,s,i}(\mu) = \frac{U_{t-1,s}\left(\prod_{i=1}^{s} V_{t,i}\right)b_{t,s}}{\sigma'_{t,s,i} - \mu} \ .$$

Hence, to compute r directions at the interaction step t the algorithm needs to perform $\mathcal{O}(r^2)$ matrix-vector multiplications, each incurring a quadratic cost, together with r quadratic time decompositions of $\Theta_{t,s}$ matrices. What remains to compute the data embedding is a multiplication of direction vectors with the kernel matrix which is again of quadratic runtime complexity. Therefore, the overall complexity of an interaction step is $\mathcal{O}(r^2n^2)$. When dealing with low-rank approximations of rank $l \ll n$, the optimization problem is solved over the space of dimension l instead of n, and the computational cost of the interaction is $\mathcal{O}(r^2l^2)$.

In a similar fashion, it is possible to show that the computational cost of the interaction with other knowledge-based constraints is also quadratic.

2.6.2 Rank-One Modification of a Diagonal Matrix

In this section, we review an approach for computing an eigendecomposition of a matrix that can be written as a rank-one modification of a diagonal matrix. Hence, expressed in the notation from the previous section (for simplicity, indices s and t are omitted), we consider the problem of finding an eigendecomposition of the matrix

$$\Theta = \Sigma + \tau a a^\top \; .$$

The review follows along the lines of the works by Arbenz (2012) and Li (1993). Let us begin by assuming that the diagonal entries of the matrix Σ satisfy $\sigma_n < \sigma_{n-1} < \cdots < \sigma_1$ and that all components in the vector a are non-zero real numbers.

If (σ', v) is an eigenpair of Θ, then it holds

$$(\Sigma - \sigma' \mathbb{I})\, v = -\tau a a^\top v \; .$$

Now, if $\sigma' = \sigma_i$ for some $1 \leq i \leq n$, then either $a^\top v = 0$ or $a_i = 0$. According to our initial assumption $a_i \neq 0$ for all $1 \leq i \leq n$ and then it must hold that $a^\top v = 0$. The latter, however, implies that $(\Sigma' - \sigma_i \mathbb{I})\, v = 0$ and $v = \mathbf{e}_i$. From here, on the other hand, it follows that $a_i = a^\top v = a^\top \mathbf{e}_i = 0$. As this is in contradiction with our assumption about the vector a, it follows that $\sigma' \neq \sigma_i$ for all $1 \leq i \leq n$. Hence, the matrix $\Sigma - \sigma' \mathbb{I}$ is of full-rank and

$$v = -\tau a^\top v \, (\Sigma - \sigma' \mathbb{I})^{-1} \, a \; . \tag{2.29}$$

Multiplying this equation with $a^\top$ from the left, we deduce that

$$a^\top v \left(1 + \tau a^\top \, (\Sigma - \sigma' \mathbb{I})^{-1} \, a \right) = 0 \; .$$

As we have already established that $a^\top v \neq 0$, the latter expression gives the secular equation for the eigenproblem of rank-one modification of a diagonal matrix,

$$g(\sigma') = 1 + \tau \sum_{i=1}^{n} \frac{a_i^2}{\sigma_i - \sigma'} \; . \tag{2.30}$$

Thus, to find the eigenvalues of Θ we need to determine the roots of the secular equation from Eq. (2.30). The problem of finding such roots has been investigated in details by Bunch et al. (1978) and (Li, 1993). We review such an approach in Section 2.6.2.1 and focus now on determining the intervals of the eigenvalues and corresponding eigenvectors of Θ.

The derivative of the secular equation is given by

$$g'(\sigma) = \tau \sum_{i=1}^{n} \frac{a_i^2}{(\sigma_i - \sigma)^2} \; .$$

Thus, the secular function $g(\sigma)$ is, for $\tau > 0$, increasing on open intervals (σ_{i+1}, σ_i) for $1 \leq i \leq n-1$, as well as on the interval $(\sigma_1, +\infty)$. In other words, the interlacing property holds for the entries from the diagonal matrix Σ and the eigenvalues of Θ, i.e.,

$$\sigma_n < \sigma_n' < \sigma_{n-1} < \sigma_{n-1}' < \cdots < \sigma_1 < \sigma_1' \; .$$

For $\tau < 0$, the smallest eigenvalue $\sigma_n' \in (-\infty, \sigma_n)$ and $\sigma_i' \in (\sigma_{i+1}, \sigma_i)$ for $1 \leq i \leq n-1$. Moreover, we can observe that for $\tau > 0$ and $\hat{a} = \sqrt{\tau} a$, the secular equation can be written as

$$g(\sigma) = 1 + \sum_{i=1}^{n} \frac{\hat{a}_i^2}{\sigma_i - \sigma} \; .$$

A similar property holds for $\tau < 0$ and we can, without loss of generality, assume that $\tau = 1$.

Having formulated the secular equation and determined the intervals with secular roots, we now show how to compute the corresponding eigenvectors. From Eq. (2.29), it follows that an eigenvector v corresponding to the eigenvalue σ' is collinear with the vector $(\Sigma - \sigma'\mathbb{I})^{-1} a$. Hence, once the eigenvalues are computed the eigenvectors are given by ($1 \leq i \leq n$)

$$v_i = \frac{\left(\Sigma - \sigma_i'\mathbb{I}\right)^{-1} a}{\left\|\left(\Sigma - \sigma_i'\mathbb{I}\right)^{-1} a\right\|} \; .$$

This method for the computation of eigenvectors can, however, be numerically unstable when a secular root is close to one of the interval endpoints (e.g., the case with $\sigma_i' \in (\sigma_i, \sigma_{i-1})$ and σ_i' very close to either σ_i or σ_{i-1}). An efficient and more numerically stable approach for computing the eigenvectors can be found in Gu and Eisenstat (1994).

2.6.2.1 Secular Equation of a Rank-One Modification of a Diagonal Matrix

The secular equation of a rank-one update modification of a diagonal matrix differs from the secular equation investigated in Section 2.4.3 in that the denominators $\sigma_i - \sigma'$ are not squared. Moreover, the secular roots now correspond to eigenvalues and all of them need to be computed. This implies that a secular root finding procedure needs to be informed of the fact that the interval endpoints are, in general, finite real numbers. Let us now review an approach (Bunch et al., 1978; Li, 1993; Arbenz, 2012) for finding the secular root from the interval (σ_m, σ_{m-1}), with $1 \leq m \leq n$ and $\sigma_0 = +\infty$. First, the non-constant terms from the secular function g are split into two functions

$$\psi_1(\xi) = \sum_{i=m}^{n} \frac{a_i^2}{\sigma_i - \xi} > 0 \quad \wedge \quad \psi_2(\xi) = \sum_{i=1}^{m-1} \frac{a_i^2}{\sigma_i - \xi} < 0 \; .$$

Then, each function ψ_i ($i = 1, 2$) is approximated by a quadratic surrogate function

$$h_{i,t}(\xi) = p_{i,t} + \frac{q_{i,t}}{\sigma_{m+1-i} - \xi} \; ,$$

where $p_{i,t}, q_{i,t} \in \mathbb{R}$ are constants. These constants are computed such that (Li, 1993)

$$h_{i,t}(\xi_t) = \psi_i(\xi_t) \quad \wedge \quad h_{i,t}'(\xi_t) = \psi_i'(\xi_t) \; ,$$

where $\{\xi_t\}_{t \geq 0} \subset (\sigma_m, \sigma_{m-1})$ is a sequence of iterative approximations of the secular root from the interval (σ_m, σ_{m-1}). Thus, for the quadratic surrogate with $i = 1$ we obtain that

$$q_{1,t} = \psi_1'(\xi_t)(\sigma_m - \xi_t)^2 \quad \wedge \quad p_{1,t} = \psi_1(\xi_t) - \psi_1'(\xi_t)(\sigma_m - \xi_t) \; .$$

Similarly, for the surrogate with $i = 2$ we have that

$$q_{2,t} = \psi_2'(\xi_t)(\sigma_{m-1} - \xi_t)^2 \quad \wedge \quad p_{2,t} = \psi_2(\xi_t) - \psi_2'(\xi_t)(\sigma_{m-1} - \xi_t) \; .$$

Now, combining the surrogates for the two terms we obtain a quadratic surrogate for the secular function in Eq. (2.30),

$$h_t(\xi) = p_t + \frac{q_{1,t}}{\sigma_m - \xi} + \frac{q_{2,t}}{\sigma_{m-1} - \xi} \; , \tag{2.31}$$

where $p_t = p_{1,t} + p_{2,t}$. The next candidate for the root ξ_{t+1} is then given as a root of the quadratic surrogate function $h_t(\xi_{t+1}) = 0$. While the quadratic function $h_t(\xi)$ has two roots, only one of them lies in the interval (σ_m, σ_{m-1}). To see this, first observe that $\lim_{\xi \to \sigma_m} h_t(\xi) = -\infty$, $\lim_{\xi \to \sigma_{m-1}} h_t(\xi) = +\infty$, and $h_t'(\xi) > 0$ for $\xi \in (\sigma_m, \sigma_{m-1})$. Thus, there can only be one root in that interval because $h_t(\xi)$ is an increasing function with different signs on the left and the right endpoint of the interval.

Bunch et al. (1978) have shown that the presented secular root finder converges to the desired root quadratically (i.e., the accuracy gets doubled at each iteration). In our empirical evaluations of the approach, the convergence usually happens after $5 - 10$ iterations. The computational complexity of finding all secular roots is $\mathcal{O}(n^2)$. As the secular roots correspond to the eigenvalues of Θ, the computational complexity of finding all eigenvalues of a rank-one modification of a diagonal matrix is also $\mathcal{O}(n^2)$. Here, we note that different roots of the secular function in Eq. (2.30) belong to distinct intervals and can be, therefore, computed in parallel. This enables an efficient GPU implementation of the secular solver resulting in a significant speed-up to the presented algorithm. Consequently, with a GPU implementation of the secular solver it is possible to increase the interaction frame rate and improve scalability.

2.6.2.2 Deflation

In this section, we address the cases not covered by the assumptions leading to the secular function in Eq. (2.30). In particular, we review an approach (Arbenz, 2012) for dealing with rank-one modification vectors that contain some zero-valued entries and diagonal matrices with multiple occurrences of a non-zero eigenvalue.

Assume that $a_i = 0$ for some $1 \leq i \leq n$. Then, $(\sigma_i, \mathbf{e}_i)$ is an eigenpair for the matrix Θ because it holds that $\Theta \mathbf{e}_i = \sigma_i \mathbf{e}_i$. As each zero-valued entry in the vector a determines one eigenpair for the matrix Θ, we can omit these components from our eigenproblem and focus only on the components with non-zero values. In this way, the general case in which zero-valued entries in a are possible is transformed to the already considered case with the vector a having all non-zero entries.

Having addressed the zero-valued entries in a, assume now that $\sigma_i = \sigma_j$ for $i \neq j$ and $1 \leq i, j \leq n$. For the two corresponding entries in vector a it is possible to define the Givens rotation (e.g., see Golub and van Loan, 1996), G_{ij}, so that the vector $\hat{a} = G_{ij}^\top a$ is given by

$$\hat{a}_k = \begin{cases} a_k \,, & k \neq i \wedge k \neq j \\ \sqrt{a_i^2 + a_j^2} \,, & k = i \\ 0 \,, & k = j \,. \end{cases}$$

On the other hand, as the matrix Σ is diagonal it also holds that $G_{ij}^\top \Sigma G_{ij} = \Sigma$. This property implies that we can work with the transformed matrix

$$\hat{\Theta} = G^\top \Theta G = \Sigma + G^\top a \left(G^\top a \right)^\top = \begin{bmatrix} \Sigma_1 + a_1 a_1^\top & 0 \\ 0 & \Sigma_2 \end{bmatrix} ,$$

where the vector a_1 consists of non-zero entries, Σ_1 is a diagonal matrix with distinct diagonal entries, and G is the product of the Givens rotation matrices (one for each index from a that gets set to zero). For matrix $\hat{\Theta}$, we can immediately compute the eigenpairs corresponding to the matrix block defined with Σ_2. For the non-trivial block with diagonal matrix Σ_1, we can use the described secular solvers (see Section 2.6.2.1) to find the corresponding eigendecomposition. Having computed the eigenvectors $\hat{V}$ of matrix $\hat{\Theta}$, we need to transform them to obtain the eigenvectors $V = G\hat{V}$ for the corresponding eigendecomposition of Θ.

The computation of an eigenvector of Θ can be numerically unstable when the secular root is too close to the corresponding interval endpoints. Typically, this happens when the gap between the successive diagonal values from Σ that determine a secular root interval is too small. To avoid this numerical instability, it is possible to first set such successive entries from Σ to identical values and then transform the matrix Θ using the Givens rotations. This transformation sets the corresponding components of the vector a to zero and adds numerical stability to the computation of an eigendecomposition of Θ.

2.7 Hyperparameter Optimization

In this section, we show how to improve the inductive bias (Baxter, 2000) of our approach by automatically tuning the hyperparameters while performing inner cross-validation. In this process, we split the training data into training and validation folds and select a validation function that will be optimized with respect to the hyperparameter vector. The optimization can be performed with an off-the-shelf optimization algorithm (e.g., L-BFGS-B solver) as long as we are able to derive the hyperparameter gradient of the validation function in a closed form. In the remainder of the section, we show how to achieve this for knowledge-based kernel principal components defined with soft control point placements and classification constraints, combined with soft orthogonality. We note here that the hyperparameter optimization problem is, in general, non-convex and that the optimization procedure outlined in this section is guaranteed to find a locally optimal set of hyperparameters. However, in our empirical evaluation (Section 2.9) we demonstrate that it is possible to find a good set of hyperparameters with a suitable initial solution to this non-convex optimization problem.

Let us begin by considering the case where knowledge-base constraints are given by the placements of control points. For this type of knowledge-based principal components, we choose the mean squared error loss as our validation function. If we denote with F and $F^{\perp}$ the training and validation examples, then the validation function is given by

$$\Xi(F, f) = \frac{1}{|F^{\perp}|} \sum_{(x,y)\in F^{\perp}} (f(x) - y)^2 = \frac{1}{|F^{\perp}|} \sum_{(x,y)\in F^{\perp}} \left(K_x^{\top}\alpha - y\right)^2 ,$$

where $f = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$ is a knowledge-based kernel principal component obtained using training examples F. Now, denote the hyperparameter vector with θ such that it includes the hyperparameters of the model R, λ_1, λ_0, together with a hyperparameter vector defining the kernel function η. Then, the gradient of the validation function is given by

$$\nabla\Xi(F, f) = \frac{2}{|F^{\perp}|} \sum_{(x,y)\in F^{\perp}} \left(K_x^{\top}\alpha - y\right)\left(\left(\frac{\partial K_x}{\partial\theta}\right)^{\top} \alpha + K_x^{\top}\frac{\partial\alpha}{\partial\theta}\right) . \tag{2.32}$$

Now, using the first stationary constraint from Eq. (2.16), we obtain that

$$\alpha = K^{-1}\left(S - \mu_{\min}\mathbb{I}\right)^{-1} b .$$

From here, we can compute the gradient of the solution α with respect to hyperparameters,

$$\begin{aligned} K_x^\top \frac{\partial \alpha}{\partial \theta} = \quad & -K_x^\top \left(SK - \mu_{\min} K\right)^{-1} \left(\frac{\partial S}{\partial \theta} K + S \frac{\partial K}{\partial \theta} - \frac{\partial \mu_{\min}}{\partial \theta} K - \mu_{\min} \frac{\partial K}{\partial \theta} \right) \left(SK - \mu_{\min} K\right)^{-1} b + \\ & K_x^\top \left(SK - \mu_{\min} K\right)^{-1} \frac{\partial b}{\partial \theta} = \\ & K_x^\top K^{-1} \left(S - \mu_{\min} \mathbb{I}\right)^{-1} \left(\frac{\partial b}{\partial \theta} - \frac{\partial S}{\partial \theta} K\alpha - S \frac{\partial K}{\partial \theta} \alpha + \frac{\partial \mu_{\min}}{\partial \theta} K\alpha + \mu_{\min} \frac{\partial K}{\partial \theta} \alpha \right) . \end{aligned}$$

If we now denote with

$$\iota = \frac{2}{|F^\perp|} \sum_{(x,y) \in F^\perp} \left(K_x^\top \alpha - y \right) K_x$$

and solve the linear system $\left(S - \mu_{\min} \mathbb{I}\right) u = \iota$ with $K \hat{u} = u$, then we get

$$\iota^\top \frac{\partial \alpha}{\partial \theta} = u^\top \left(\frac{\partial b}{\partial \theta} - \frac{\partial S}{\partial \theta} K\alpha - S \frac{\partial K}{\partial \theta} \alpha + \frac{\partial \mu_{\min}}{\partial \theta} K\alpha + \mu_{\min} \frac{\partial K}{\partial \theta} \alpha \right) . \tag{2.33}$$

Here, it is important to note that the system $\left(S - \mu_{\min} \mathbb{I}\right) u = \iota$ can be solved in quadratic time using an eigendecomposition of the matrix S, which can be obtained efficiently from the eigendecomposition of K for the first knowledge-based kernel principal component.

Before we proceed with the derivatives of the matrix-based terms, we need to find the derivative of the optimal Lagrange multiplier $\mu_{\min}$. In order to do this, we plug the expression for α into the second stationary constraint from Eq. (2.16) to deduce that

$$b^\top \left(S - \mu_{\min} \mathbb{I}\right)^{-2} b = R^2.$$

Thus, to find the derivative of $\mu_{\min}$ with respect to θ we need to implicitly derive the latter equation. In particular, taking the derivative of both sides with respect to θ we obtain

$$\frac{\partial R^2}{\partial \theta} = 2 b^\top \left(S - \mu_{\min} \mathbb{I}\right)^{-2} \left[\frac{\partial b}{\partial \theta} - \left(\frac{\partial S}{\partial \theta} - \frac{\partial \mu_{\min}}{\partial \theta} \mathbb{I} \right) \left(S - \mu_{\min} \mathbb{I}\right)^{-1} b \right] .$$

Now, plugging $K\alpha = \left(S - \mu_{\min} \mathbb{I}\right)^{-1} b$ we can transform this equation into

$$\frac{\partial R^2}{\partial \theta} = 2 \alpha^\top K \left(S - \mu_{\min} \mathbb{I}\right)^{-1} \left(\frac{\partial b}{\partial \theta} - \frac{\partial S}{\partial \theta} K\alpha + \frac{\partial \mu_{\min}}{\partial \theta} K\alpha \right) .$$

If we now solve the linear system $\left(S - \mu_{\min} \mathbb{I}\right) \nu = K\alpha$, then

$$\frac{\partial R^2}{\partial \theta} = 2 \nu^\top \left(\frac{\partial b}{\partial \theta} - \frac{\partial S}{\partial \theta} K\alpha + \frac{\partial \mu_{\min}}{\partial \theta} K\alpha \right) .$$

Before we give the derivatives of the optimal Lagrange multiplier with respect to the individual hyperparamters, let us remind ourselves that in the considered case $S = K^{-1} + \frac{\lambda_1^2}{|\mathcal{A}|} A + \lambda_0^2 H$ and $b = \frac{\lambda_1^2}{|\mathcal{A}|} y^\top A$ (hyperparameters are squared to ensure their non-negativity). Thus, we have that it holds

$$\begin{aligned}
\frac{\partial \mu_{\min}}{\partial R} &= \frac{R}{\nu^\top K \alpha} \\
\frac{\partial \mu_{\min}}{\partial \lambda_1} &= \frac{2}{\nu^\top K \alpha}\left(\frac{\lambda_1}{|\mathcal{A}|}\nu^\top A K \alpha - \frac{\nu^\top b}{\lambda_1}\right) \\
\frac{\partial \mu_{\min}}{\partial \lambda_0} &= \frac{2\lambda_0}{\nu^\top K \alpha}\nu^\top H K \alpha \\
\frac{\partial \mu_{\min}}{\partial \eta} &= -\frac{1}{\nu^\top K \alpha}\nu^\top K^{-1}\frac{\partial K}{\partial \eta}\alpha \ .
\end{aligned}$$

Having computed the gradient of the optimal Lagrange multiplier, we now turn to the gradient of the validation function. Plugging the computed gradients for the optimal multiplier into Eq. (2.33) we obtain

$$\begin{aligned}
\iota^\top \frac{\partial \alpha}{\partial R} &= R\frac{u^\top K \alpha}{\nu^\top K \alpha} \\
\iota^\top \frac{\partial \alpha}{\partial \lambda_1} &= 2\Delta^\top\left(\frac{b}{\lambda_1} - \frac{\lambda_1}{|\mathcal{A}|}AK\alpha\right) \\
\iota^\top \frac{\partial \alpha}{\partial \lambda_0} &= -2\lambda_0 \Delta^\top H K \alpha \\
\iota^\top \frac{\partial \alpha}{\partial \eta} &= \Delta^\top K^{-1}\frac{\partial K}{\partial \eta}\alpha - u^\top S \frac{\partial K}{\partial \eta}\alpha + \mu_{\min} u^\top \frac{\partial K}{\partial \eta}\alpha \ ,
\end{aligned}$$

where $\Delta = u - \frac{u^\top K\alpha}{\nu^\top K \alpha}\nu$. Now, plugging these individual gradients into Eq. (2.32) we obtain the gradient of the validation function $\Xi(F, f)$.

For domain knowledge specified through classification constraints, the mean squared error loss function might not be the best choice on the validation folds. Instead, we propose to use the squared hinge loss as the validation function, i.e.,

$$\Xi(F, f) = \frac{1}{|F^\perp|}\sum_{(x,y)\in F^\perp} \max\{0, 1 - yf(x)\}^2 = \frac{1}{|F^\perp|}\sum_{(x,y)\in F^\perp} \max\left\{0, 1 - yK_x^\top \alpha\right\}^2 \ .$$

The gradient of this validation function is given by

$$\nabla \Xi(F, z) = -\frac{2}{|F^\perp|}\sum_{(x,y)\in F_*^\perp} y\left(1 - yK_x^\top \alpha\right)\left(\left(\frac{\partial K_x}{\partial \theta}\right)^\top \alpha + K_x^\top \frac{\partial \alpha}{\partial \theta}\right) ,$$

where $F_*^\perp = \{(x, y) \in F^\perp \mid 1 - yf(x) > 0\}$. Now, following the derivation for the mean squared error validation function it is possible to derive the gradients of the latter validation function with respect to the individual hyperparameters.

Having shown how to compute the hyperparameter gradient on a validation sample, we now discuss the choice of fold splits in inner cross-validation. In order to select good and reliable hyperparameters, we propose to use (stratified) K-fold splits for inner cross-validation. For one such split, the training is performed on the batch of $(K - 1)$ training folds and the hyperparameters are optimized on the remaining validation fold. Each inner cross-validation fold is used exactly once as a validation fold and we refer to the hyperparameter gradients computed on these folds as *fold gradients*. Having determined the fold gradients for all inner cross-validation splits, the ultimate hyperparameter gradient is defined as their average.

2.8 Discussion

Traditional dimensionality reduction methods are in many problems used primarily for visualization purposes. There are several well known methods for dimensionality reduction, but the majority of them is unsupervised and unable to incorporate domain knowledge into a low-dimensional data representation. Some of the well known traditional methods for dimensionality reduction are principal component analysis (Jolliffe, 1986), metric multidimensional scaling (Cox and Cox, 2000), isomap (Tenenbaum et al., 2000), maximum variance unfolding (Weinberger and Saul, 2004), and locally linear embedding (Roweis and Saul, 2000). These methods can be also viewed as instances of kernel principal component analysis with a suitably defined kernel matrix (Ham et al., 2004).

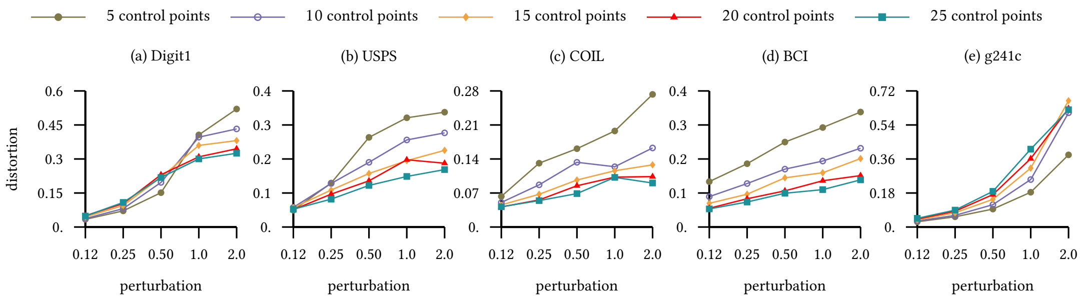
Interaction with these traditional dimensionality reduction methods is hardly intuitive and forces domain experts to express their reasoning about parameters of the algorithm instead of data points. In contrast, a user study conducted by Andrews et al. (2010) shows domain experts prefer to interact directly with a visualization by placing a few control points in accordance with the current understanding of data. In general, user studies report benefits of the interactive over the static approach in data visualization (e.g., see Callahan and Koenemann, 2000; Jeong et al., 2009). To overcome this problem several tools for data visualization were designed with the ultimate goal to facilitate the understanding of the underlying algorithm and the interaction with model parameters. One such tool (Jeong et al., 2009) facilitates the understanding of principal component analysis (Jolliffe, 1986) through four coordinated views of that approach: *i*) projection view, *ii*) eigenvector view, *iii*) data view, and *iv*) correlation view. In the first view, instances are projected onto two principal components (typically, the top two principal components). The second view displays instances as lines passing through points for which the x-axis represents the index of a principal component and the y-axis corresponds to projections of instances onto principal components. Similar to this, the third view displays instances as lines with respect to the input features and their values. The correlation view shows the Pearson correlation between pairs of features and plots the instances in two dimensional spaces given by the pairs of features. Beside these views, the tool also provides means for feature weighting such that the covariance matrix for the input data is computed using the weighted inner products between instances. Interaction with a data visualization is provided through the projection view and feature weighting. In particular, for movement of a point in the projection space along one of the two principal directions the tool updates all four views simultaneously. In this way, interactive principal component analysis (Jeong et al., 2009) provides an interpretation of the influence of the movement of a point in the projection space on the coordinates of the corresponding instance in the input space. In addition to this, sliders for feature weights provide means for domain experts to reason about the importance of particular features for an insight gained from a data visualization. Similar to interactive principal component analysis, Buja et al. (2008) and Broekens et al. (2006) have developed a variant of interactive multi-dimensional scaling. The interaction with a visualization is restricted to a static placement of a small number of control points in the projection space, whereby the algorithm positions the remaining points.

From the interactive visualization perspective, the most related to our work are techniques developed by Endert et al. (2011) and Leman et al. (2013). The proposed techniques allow movement of control points in a projection space and the update to the projections is interpreted as a feedback similar to must- and cannot-link constraints. Both approaches (Endert et al., 2011; Leman et al., 2013) perform interaction by incorporating expert feedback into probabilistic principal component analysis (Tipping and Bishop, 1999), multi-dimensional scaling, and generative topographic mapping. In particular, the two interactive variants of multidimensional scaling compute the pairwise distances between instances using weighted features. Then, after re-arranging a small number of control points in the projection space the algorithm optimizes the feature weights such that the weighted pairwise distances in the input space reflect the pairwise distances between the control points in the projection space. This type of interaction allows experts to reason about the importance of particular features for knowledge injected into multi dimensional scaling through a data visualization. For interactive probabilistic principal component analysis, Endert et al. (2011) assume that the covariance matrix of input features conditioned on the hyperparameters of the model follows the inverse Wishart distribution. The interaction with probabilistic principal component analysis is then modeled by first constructing a hypothetical covariance matrix that interprets movement of control points in the projection space and then replacing the actual covariance of input features with the maximum a posteriori estimator given by an inverse Wishart distribution that combines the two covariance matrices. The hypothetical covariance matrix is constructed by interpreting the movement of control points in terms of input features and then enforcing low/high variance along features for which the pre-images of control points are close/far from each other. The approaches by Endert et al. (2011) and Leman et al. (2013) do not offer means for structural exploration of the dataset by observing how the embedding reacts to a placement of a selected control point. Moreover, probabilistic principal component analysis and multi-dimensional scaling are highly sensitive to outliers and produce visualizations with huge number of overlapping points for such datasets.

The Invis tool (Paurat and Gärtner, 2013) is one of the first truly interactive tools that enables interaction with a visualization through explicit placement of control points, whereby all related data points automatically and smoothly adjust their location accordingly. The tool realizes this type of interaction using the least square projections (LSP). As argued in our workshop paper (Paurat et al., 2013b), the least square projections are in general not a good choice for data visualization and the same can be said about any purely supervised learning algorithm (e.g., linear discriminant analysis Izenman, 2008). To see this, consider training a linear regression on a sparse high dimensional dataset. If it is trained using a very few instances, the weight vector will only be non-zero over the union of their non-zero attributes and all instances having different non-zero attributes will be mapped to the origin.

From the methodological perspective, our method is a spectral method for semi-supervised learning and closely related to spectral graph transducer (Joachims, 2003) and semi-supervised kernel principal component analysis (Walder et al., 2010). These two approaches can be seen as a relaxation of transductive support vector machines and/or a generalization of kernel principal component analysis. In spectral graph transducer (Joachims, 2003), the goal is to find a hypothesis such that the predictions at neighboring instances are similar in value and of the same sign over two different classes. Joachims (2003) proposed to find such hypothesis by solving a constrained version of the normalized graph cut problem. The relaxed version of that problem can directly be related to problems in Eq. (2.11) and Eq. (2.12) In particular, for a positive definite kernel matrix given by the pseudo-inverse of a graph Laplacian and domain knowledge specified via placement of control points the optimization problem in Eq. (2.11) with $s = 1$ is equivalent to the relaxed graph cut problem proposed by Joachims (2003). In contrast to our approach, spectral graph transducer does not consider a variety of domain knowledge constraints nor cases in which the matrix defining the quadratic term in the optimization problem from Eq. (2.12) is indefinite (e.g., as a result of using pairwise constraints). Moreover, that approach is restricted to transductive settings and does not provide means for hyperparameter optimization. Motivated by spectral

Figure 2.1: The distortion of embeddings generated using knowledge-based kernel principal component analysis as the number of perturbed control points increases.



graph transducer, Walder et al. (2010) considered the least square variant of kernel principal component analysis that corresponds to the discussed variant of our knowledge-based kernel principal component analysis, in which domain knowledge can be injected only through placement of control points. As the approach does not offer means for the optimization of a large number of hyperparameters (radius, regularization parameter, and kernel specific hyperparameters), the process of inner cross-validation can be computationally intensive. Also, neither of the two approaches has been considered in the context of data visualization.

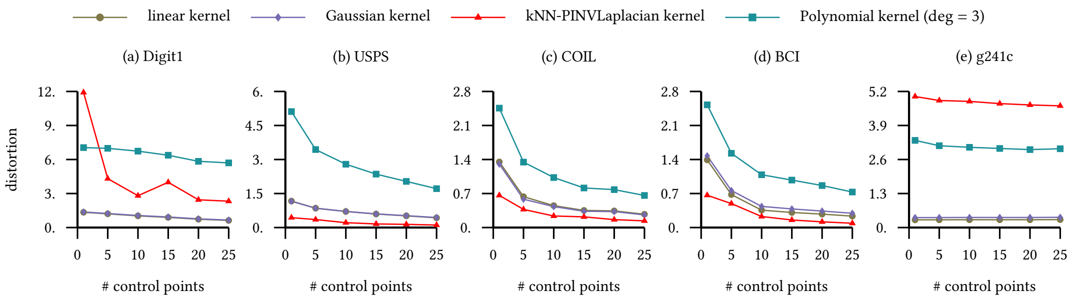
2.9 Experiments

The best, and ultimately only true, way to evaluate an algorithm for interactive data visualization and exploration is via a study with real domain experts that are using a tool implementing that algorithm. In the absence of the study we performed a number of in silico experiments which aim at illustrating the utility and sensibility of our approach. In particular, we show that: (i) a rendered embedding is robust under small changes in the placement of control points; (ii) the approach is flexible in choosing a low-dimensional embedding from the many possible ones; (iii) 'sufficient' amount of information is retained in a visualization; (iv) it is possible to detect structures that do not necessarily exhibit the highest correlation with variance and which are, therefore, obscured in a regular kernel principal component analysis embedding. We study the properties $(i) - (iii)$ on benchmark data sets for semi-supervised learning (Chapelle et al., 2006) and use an artificial and a real-world dataset to show the property (iv). In the experiments we use different kernels: the Gaussian kernel with the bandwidth parameter equal to the median of pairwise distances between instances, the pseudo-inverse Laplacian kernel with the k-NN graph defined by the cosine metric, linear and polynomial kernel of degree three. All the reported results are averaged over ten runs.

How stable is our approach?

In exploratory data visualization it should be possible to smoothly change the embedding by moving a control point throughout the projection space. In other words, small perturbations of a control point should result in small perturbations of the overall embedding. We empirically verify the stability of the proposed method by moving control points randomly throughout the projection space. More specifically, for each coordinate of a control point we sample an additive perturbation value from the zero mean Gaussian distribution with the variance given by a fraction of the median of pairwise distances between projections generated by kernel principal component analysis. The distortion or the difference between the two embeddings is measured by the average displacement of a point between them and this value is scaled

Figure 2.2: The distortion between the target and current embeddings with respect to the number of re-arranged control points. The results show that we can recover a target embedding with a small number of re-arrangements.



by the median pairwise distance between the projections generated by kernel principal component analysis. In Figure 2.1 we show the distortion as the perturbation increases across five benchmark data sets (Chapelle et al., 2006). The empirical results clearly indicate that the proposed method provides means for a stable interactive visualization of datasets.

How flexible is our approach?

It is possible to generate different embeddings of the same dataset with kernel principal component analysis using different kernels. To show the flexibility of the proposed method we set up an experiment with a sum of different kernels and show that the proposed method can choose the kernel principal component analysis embedding corresponding to a kernel by re-arranging control points accordingly. In particular, we combine the Gaussian and pseudo-inverse Laplacian kernel that produce geometrically very different kernel principal component analysis embeddings of the considered datasets. We again report the distortion between the two embeddings as a measure of their difference. The empirical results indicate that, for each used kernel, it is possible to recover the corresponding embedding generated by kernel principal component analysis. Figure 2.2 shows the distortion between the current and the target embeddings as the number of selected control points increases.

Table 2.1: The test error is given by the percentage of misclassified instances on the semi-supervised learning benchmark datasets prepared by Chapelle et al. (2006). The results for all baselines also originate from that work.

Algorithm	g241c		g241d		Digit1		usps		coil		bci		Text	
	10	100	10	100	10	100	10	100	10	100	10	100	10	100
1-nn	47.88	43.93	46.72	42.45	13.65	3.89	16.66	5.81	63.36	17.35	49.00	48.67	38.12	30.11
svm	47.32	23.11	46.66	24.64	30.60	5.53	20.03	9.75	68.36	22.93	49.85	34.31	45.37	26.45
mvu + 1-nn	48.68	44.05	47.28	43.21	11.92	3.99	14.88	6.09	65.72	32.27	49.76	47.42	39.40	30.74
isomap + 1-nn	47.88	43.93	46.72	42.45	13.65	3.89	16.66	5.81	63.36	17.35	49.00	48.67	38.12	30.11
lle + 1-nn	47.15	43.03	45.56	38.20	14.42	2.83	23.34	6.50	62.62	28.71	47.95	47.89	45.32	32.83
pca + 1-nn	39.38	33.51	37.03	25.92	21.70	8.27	23.40	9.50	67.88	28.41	49.17	48.58	41.65	28.83
LapEig + 1-nn	47.47	42.14	45.34	39.43	12.04	2.52	19.14	6.09	67.96	36.49	49.94	48.64	40.84	30.92
lem + 1-nn	44.05	40.28	42.22	37.49	23.47	6.12	19.82	7.64	65.91	23.27	48.74	44.83	39.44	30.77
qc + cmn	39.96	22.05	46.55	28.20	9.80	3.15	13.61	6.36	59.63	10.03	50.36	46.22	40.79	25.71
DiscreteReg	49.59	43.65	49.05	41.65	12.64	2.77	16.07	4.68	63.38	9.61	49.51	47.67	40.37	24.00
t-svm	24.71	18.46	50.08	22.42	17.77	6.15	25.20	9.77	67.50	25.80	49.15	33.25	31.21	24.52
sgt	22.76	17.41	18.64	9.11	8.92	2.61	25.36	6.80	–	–	49.59	45.03	29.02	23.09
ClusterKernel	48.28	13.49	42.05	4.95	18.73	3.79	19.41	9.68	67.32	21.99	48.31	35.17	42.72	24.38
Data-DepReg	41.25	20.31	45.89	32.82	12.49	2.44	17.96	5.10	63.65	11.46	50.21	47.47	–	–
lds	28.85	18.04	50.63	23.74	15.63	3.46	17.57	4.96	61.90	13.72	49.27	43.97	27.15	23.15
LapRLS	43.95	24.36	45.68	26.46	5.44	2.92	18.99	4.68	54.54	11.92	48.97	31.36	33.68	23.57
chm	39.03	24.82	43.01	25.67	14.86	3.79	20.53	7.65	–	–	46.90	36.03	–	–
kb-kpca	14.58	12.41	43.79	22.54	8.18	1.96	19.87	6.40	*17.51	*7.31	48.69	29.25	30.76	24.98

* This is a multi-class classification problem and the reported error (kb-kpca) is the classification error of *class* 1 *vs all* classifier.

* In Chapelle et al. (2006), the authors do not specify whether the errors reported for other baselines are multi-class or *1 vs all* classification errors.

How informative is our approach?

A satisfactory embedding should be able to retain a fair amount of information from the input space. To measure the amount of information the proposed method retains we simulate a classification task. We use the semi-supervised learning benchmark datasets with a small number of labeled instances, prepared by Chapelle et al. (2006). The benchmark was designed to provide an objective evaluation of semi-supervised learning algorithms, which typically rely either on the clustering or manifold assumption (e.g., see Section 2.2). In constructing knowledge-based kernel principal components we use the classification constraints and measure the effectiveness of the approach using the 1-NN classifier on test samples. We compare our method using only 1 dimensional projections against the state-of-the-art unsupervised dimensionality reduction techniques with many more dimensions and other approaches designed especially for semi-supervised classification. The considered dimensionality reduction algorithms use the estimated intrinsic dimension of a dataset as the dimension of the projection space (Chapelle et al., 2006). In particular, the dimensionality reduction baselines use (Chapelle et al., 2006): 38 dimensions for g241c, 4 dimensions for Digit1, 9 dimensions for USPS, 8 dimensions for BCI, and 3 dimensions for COIL dataset. Table 2.1 presents the results of our simulations and gives the numbers reported by Chapelle et al. (2006) for different baselines. Our empirical results demonstrate that the proposed approach is competitive with the state-of-the-art baselines over the whole collection of benchmark datasets and, thus, capable of retaining a satisfactory amount of information from the input space. The simulations of knowledge-based kernel principal component analysis were performed using the Gaussian and pseudo-inverse Laplacian kernels.

Can we discover structures hidden by the plain principal component analysis?

We have created a three dimensional artificial dataset to demonstrate that the proposed approach is able to discover structures in datasets that do not exhibit the highest correlation with variance. Such structures remain hidden in kernel principal component analysis and the proposed method is capable of detecting them by the appropriate placement of control points. In particular, we sample 3 sets/clusters of points from a two dimensional Gaussian distribution and embed these sets into three dimensional space such that the z-axis coordinates for each cluster are obtained by sampling from the normal distributions with means at $1, 0$ and -1. We choose the variance for the z-axis sampling such that the clusters/sets of points barely touch each other. For a sample generated in this way the within-clusters variance is higher than the between-cluster variance and the cluster structure remains hidden in kernel principal component analysis (see the leftmost picture in Figure 2.3). Moving the two most distant points of each cluster apart (a total of 6 displacements) we discover the cluster structure obscured by kernel principal component analysis (the rightmost picture in Figure 2.3).

Figure 2.3: Discovering clusters that are hidden in the embedding generated by kernel principal components.

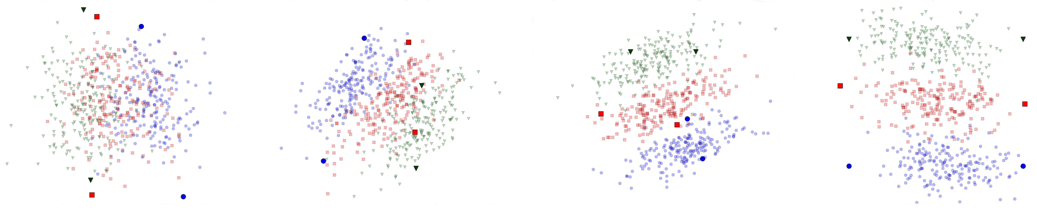
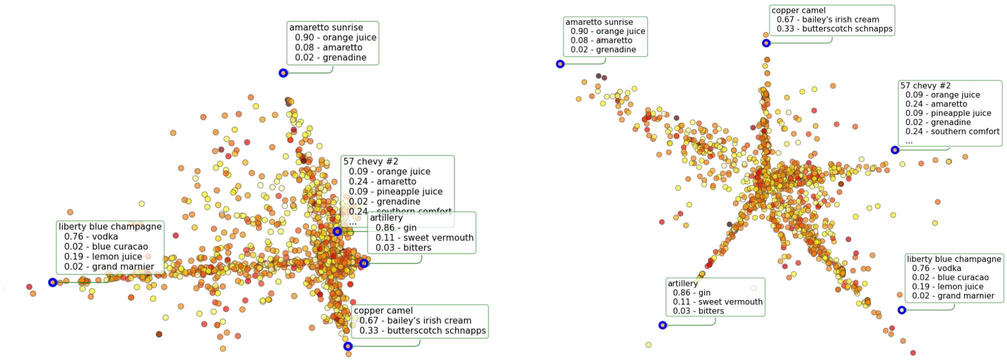


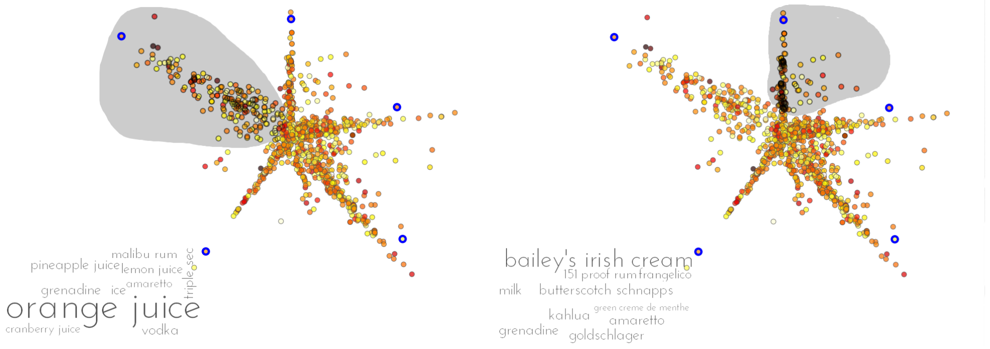
Figure 2.4: Discovering cluster structures which are hidden in the embedding of a cocktail dataset (Paurat et al., 2014) generated by kernel principal component analysis. The plots are produced using a variant of the InVis tool (Paurat and Gärtner, 2013) for interactive data visualization.



Having demonstrated that knowledge-based kernel principal component analysis can discover clusters hidden in embeddings generated by plain kernel principal component analysis, we now perform a study on a real-world dataset. The dataset contains cocktail recipes with fractions of each ingredient required to make a cocktail (Paurat et al., 2013b). The left panel in Figure 2.4 shows the embedding generated using kernel principal component analysis. The panel also shows the location of a few selected control points. After we re-arrange these points and compute the corresponding knowledge-based kernel principal components we obtain the embedding at the right panel in Figure 2.4. This panel shows additional cluster structures which are obscured by plain kernel principal component analysis.

Having discovered additional clusters, we proceed further to understand their relation to actual cocktails. For that, we generate ingredient clouds which depict the dominating ingredients appearing in cocktails corresponding to projections within selected clusters. The ingredient clouds depicted in Figure 2.5 indicate that discovered clusters group cocktails according to their flavors (e.g., see also Paurat et al., 2014). More specifically, the ingredient cloud at the left panel in Figure 2.5 contains cocktails in which juices dominate and indeed in the labels provided for the dataset (Paurat et al., 2014) these points corresponds to *juicy cocktails*. Similarly, the ingredient cloud at the right panel of the figure contains points corresponding to *creamy cocktails*, which are created by using dominantly creamy ingredients.

Figure 2.5: An interpretation of cluster structures discovered using the embedding of a cocktail dataset (Paurat et al., 2014) generated by knowledge-based kernel principal component analysis. The plots are produced using a variant of the InVis tool (Paurat and Gärtner, 2013) for interactive data visualization.



CHAPTER 3

Greedy Feature Construction

Every supervised learning algorithm with the ability to generalize from training examples to unseen data points has some type of inductive bias (Baxter, 2000). The bias can be defined as a set of assumptions that together with the training data explain the predictions at unseen points (Mitchell, 1997). In order to simplify theoretical analysis of learning algorithms, the inductive bias is typically represented by a choice of a hypothesis space (e.g., the inductive bias of linear regression models is the assumption that the relationship between inputs and outputs is linear). The fundamental limitation of learning procedures with an a priori specified hypothesis space (e.g., linear models or kernel methods with a preselected kernel) is that they can learn good concept descriptions only if the hypothesis space selected beforehand is large enough to contain a good solution to the considered problem and small enough to allow good generalization from a small number of training examples. As finding a good hypothesis space is equivalent to finding a good set of features (Baxter, 2000), we propose an effective supervised feature construction method to tackle this problem. The main goal of the approach is to embed the data into a feature space for which a set of linear hypotheses is of sufficient capacity. The motivation for this choice of hypotheses is in the desire to exploit the scalability of existing algorithms for training linear models. It is for their scalability that these models are frequently a method of choice for learning on large scale datasets. For example, the implementation of linear SVM (Fan et al., 2008) has won the large scale learning challenge at ICML 2008 and KDD CUP 2010. However, as the set of linear hypotheses defined on a small or moderate number of input features is usually of low capacity these methods often learn inaccurate descriptions of target concepts. The proposed approach surmounts this and exploits the scalability of existing algorithms for training linear models while overcoming their low capacity on input features. The latter is achieved by harnessing the information contained in the labeled training data and constructing features by empirically fitting squared error residuals (Section 3.1).

We draw motivation for our approach by considering the minimization of the expected squared error using functional gradient descent (Section 3.1.1). In each step of the descent, the current estimator is updated by moving in the direction of the residual function. We want to mimic this behavior by constructing a feature representation incrementally so that for each step of the descent we add a feature which approximates well the residual function. In this constructive process, we select our features from a predetermined set of basis functions which can be chosen so that a high capacity set of linear hypotheses corresponds to the constructed feature space (Section 3.1.2). In our theoretical analysis of the approach, we provide a convergence rate for this constructive procedure (Section 3.1.3) and give a generalization bound for the empirical fitting of squared error residuals (Section 3.1.4). The latter is needed because the feature construction is performed based on an independent and identically distributed sample of labeled instances. The approach, presented in Section 3.1.5, is highly flexible and allows for an extension of a feature representation without complete re-training of the model. As the constructive procedure imitates gradient descent, a stopping criteria based on an accuracy threshold can be devised and the algorithm can then be simulated without specifying the number of features *a priori*. In this way, the algorithm can terminate sooner than alternative approaches for simple hypotheses. The method is easy to implement and it can be scaled to millions of instances with a parallel implementation.

Having described a distributed version of greedy feature construction, we turn our attention to an instance of the approach that can construct a feature representation corresponding to a high capacity set of linear hypotheses (Section 3.2). In particular, we focus on greedy feature construction with Fourier features as basis functions and review a connection to an important class of kernel functions known as stationary kernels (Section 3.2.1). This connection between Fourier features and stationary kernels allows us to show that our approach can approximate arbitrarily well any bounded function from any stationary reproducing kernel Hilbert space (Section 3.2.2). Moreover, previous work (Micchelli et al., 2006) has shown that some kernels from this class (e.g., the Gaussian kernel) correspond to high capacity hypothesis spaces capable of approximating any continuous function defined on a compact set containing instances in its interior. Thus, our approach can for a particular choice of basis function overcome problems with low capacity of linear hypotheses on input features.

To evaluate the effectiveness of our approach empirically, we compare it to other related approaches by training linear ridge regression models in the feature spaces constructed by these methods. The focus of the evaluation is on the described instance of our approach with Fourier features as basis functions. For this particular choice of features, our approach is directly comparable to two popular algorithms for learning with Fourier features: random kitchen sinks (Section 3.2.1) and à la carte (Section 3.2.3). Our empirical results indicate a superior performance of the proposed approach over these two baselines. The results are presented in Section 3.3 and the approaches are discussed in Section 3.4.

3.1 Greedy Feature Construction

In this section, we present our feature construction approach. We start with an overview where we introduce the problem setting and motivate our approach by considering the minimization of the expected squared error using functional gradient descent. We then define a set of features and show that our greedy constructive procedure converges. Following this, we give a generalization bound for the empirical fitting of squared error residuals and provide a pseudo-code description of our approach.

3.1.1 Overview

We consider a learning problem with the squared error loss function where the goal is to find a mapping from a Euclidean space to the set of reals. In these problems, it is typically assumed that a sample $\mathbf{z} = ((x_1, y_1), \ldots, (x_n, y_n))$ of n examples is drawn independently from a Borel probability measure ρ defined on $Z = X \times Y$, where X is a compact subset of a finite dimensional Euclidean space with the inner product $\langle \cdot, \cdot \rangle$ and $Y \subset \mathbb{R}$. For every $x \in X$ let

$\rho(y \mid x)$ be the conditional probability measure on Y and ρ_X be the marginal probability measure on X. For the sake of brevity, when it is clear from the context, we will write ρ instead of ρ_X. Let $f_\rho(x) = \int y \, d\rho(y \mid x)$ be the bounded target/regression function of the measure ρ. Our goal is to construct a feature representation such that there exists a linear hypothesis on this feature space that approximates well the target function. For an estimator f of the function f_ρ, we measure the goodness of fit with the expected squared error in ρ,

$$\mathcal{E}_\rho(f) = \int (f(x) - y)^2 \, d\rho \, .$$

The empirical counterpart of the error, defined over a sample $\mathbf{z} \in Z^n$, is denoted with

$$\mathcal{E}_\mathbf{z}(f) = \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 \, .$$

Having defined the problem setting, we proceed to motivate our approach by considering the minimization of the expected squared error using functional gradient descent. For that, we first review the definition of functional gradient (e.g., see Section 3.2 in Gelfand and Fomin, 1963). For a functional F defined on a normed linear space and an element p from this space, the functional gradient $\nabla F(p)$ is the principal linear part of a change in F after it is perturbed in the direction of q,

$$F(p+q) = F(p) + \psi(q) + \varepsilon \|q\| \, ,$$

where $\psi(q)$ is the linear functional with $\nabla F(p)$ as its principal linear part, and $\varepsilon \to 0$ as $\|q\| \to 0$. In our case, the normed space is the Hilbert space of square integrable functions, $\mathcal{L}^2_\rho(X)$, and for the expected squared error functional on this space we have that it holds

$$\mathcal{E}_\rho(f + \varepsilon q) - \mathcal{E}_\rho(f) = \left\langle 2\left(f - f_\rho\right), \varepsilon q \right\rangle_{\mathcal{L}^2_\rho(X)} + \mathcal{O}\left(\varepsilon^2\right) \, .$$

Hence, an algorithm for the minimization of the expected squared error using functional gradient descent on this space could be specified by

$$f_{t+1} = \nu f_t + 2(1-\nu)\left(f_\rho - f_t\right) \, ,$$

where $0 \leq \nu \leq 1$ denotes the learning rate and f_t is the estimate at step t. The functional gradient direction $2\left(f_\rho - f_t\right)$ is the residual function at the step t and the main idea behind our approach is to iteratively refine our feature representation by extending it with a new feature that approximates well the current residual function. In this way, for a suitable choice of learning rate ν, the functional descent would be performed through a convex hull of features and in each step we would have an estimate of the target function f_ρ expressed as a convex combination of the constructed features.

3.1.2 Greedy Features

We introduce now a set of features parameterized with a ridge basis function and hyperparameters controlling the smoothness of these features. As each subset of features corresponds to a set of hypothesis (Baxter, 2000), in this way we specify a family of possible hypothesis spaces. For a particular choice of ridge basis function, we later argue (in Section 3.2.2) that the approach outlined in the previous section can construct a highly expressive feature representation (i.e., a high capacity hypothesis space).

Let $\mathcal{C}(X)$ be the Banach space of continuous functions on X with the uniform norm. For a Lipschitz continuous function $\phi : \mathbb{R} \to \mathbb{R}$, $\left\|\phi\right\|_\infty \leq 1$, and constants $r, s, t > 0$ let $\mathcal{F}_\Theta \subset \mathcal{C}(X)$, $\Theta = (\phi, r, s, t)$, be a set of ridge-wave functions defined on the set X,

$$\mathcal{F}_\Theta = \left\{ a\,\phi\left(\langle w, x\rangle + b\right) \mid w \in \mathbb{R}^d, a, b \in \mathbb{R}, |a| \leq r, \|w\|_2 \leq s, |b| \leq t \right\} .$$

From this definition, it follows that for all $g \in \mathcal{F}_\Theta$ it holds $\|g\|_\infty \leq r$. As all the ridge-wave functions from $\mathcal{F}_\Theta$ are bounded and Lipschitz continuous, they are also square integrable in the measure ρ. Therefore, $\mathcal{F}_\Theta$ is a subset of the Hilbert space of square integrable functions defined on X with respect to the probability measure ρ, i.e., $\mathcal{F}_\Theta \subset \mathcal{L}^2_\rho(X)$.

3.1.3 Convergence

For the purpose of this chapter, it suffices to show the convergence of ε-greedy sequences of functions (see Definition 3.1) in Hilbert spaces. We, however, choose to provide a stronger result which holds for ε-greedy sequences in uniformly smooth Banach spaces. In the remainder of the chapter, $\mathrm{co}(S)$ and $\overline{S}$ will be used to denote the convex hull of elements from a set S and the closure of S, respectively.

Definition 3.1. *Let $\mathcal{B}$ be a Banach space with norm $\|\cdot\|$ and let $S \subseteq \mathcal{B}$. An incremental sequence is any sequence $\{f_m\}_{m\geq 1}$ of elements of $\mathcal{B}$ such that $f_1 \in S$ and for each $m \geq 1$ there is some $g \in S$ such that $f_{m+1} \in \mathrm{co}(\{f_m, g\})$. An incremental sequence is greedy with respect to an element $f \in \overline{\mathrm{co}(S)}$ if for all $m \in \mathbb{N}$ it holds*

$$\|f_{m+1} - f\| = \inf\left\{ \|h - f\| \mid h \in \mathrm{co}(\{f_m, g\}),\ g \in S \right\} .$$

Given a positive sequence of allowed slack terms $\{\varepsilon_m\}_{m\geq 1}$, an incremental sequence $\{f_m\}_{m\geq 1}$ is called ε-greedy with respect to f if for all $m \in \mathbb{N}$ it holds

$$\|f_{m+1} - f\| < \inf\left\{ \|h - f\| \mid h \in \mathrm{co}(\{f_m, g\}),\ g \in S \right\} + \varepsilon_m .$$

Having introduced the notion of an ε-greedy incremental sequence of functions, let us now relate it to our feature construction approach. In the outlined constructive procedure (Section 3.1.1), we proposed to select new features corresponding to the functional gradient at the current estimate of the target function. Now, if at each step of the functional gradient descent there exists a ridge-wave function from our set of features which approximates well the residual function (w.r.t. f_ρ) then this sequence of functions defines a descent through $\mathrm{co}(\mathcal{F}_\Theta)$ which is an ε-greedy incremental sequence of functions with respect to $f_\rho \in \overline{\mathrm{co}(\mathcal{F}_\Theta)}$. In Section 3.1.1, we have also demonstrated that $\mathcal{F}_\Theta$ is a subset of the Hilbert space $\mathcal{L}^2_\rho(X)$ and this is by definition a Banach space. Thus, in accordance with Definition 3.1, we now consider under what conditions an ε-greedy sequence of functions from this space converges to any target function $f_\rho \in \overline{\mathrm{co}(\mathcal{F}_\Theta)}$. Note that this relates to our result from Section 3.2.2 where we will show that the capacity of $\overline{\mathrm{co}(\mathcal{F}_\Theta)}$ can be large. Before we show the convergence of our constructive procedure, we need to prove that an ε-greedy incremental sequence of functions/features can be constructed in our setting. For that, we characterize the Banach spaces in which it is always possible to construct such sequences of functions/features.

Definition 3.2. *Let $\mathcal{B}$ be a Banach space, $\mathcal{B}^*$ the dual space of $\mathcal{B}$, and $f \in \mathcal{B}$, $f \neq 0$. A peak functional for f is a bounded linear operator $F \in \mathcal{B}^*$ such that $\|F\|_{\mathcal{B}^*} = 1$ and $F(f) = \|f\|_{\mathcal{B}}$. The Banach space $\mathcal{B}$ is said to be smooth if for each $f \in \mathcal{B}$, $f \neq 0$, there is a unique peak functional.*

The existence of at least one peak functional for all $f \in \mathcal{B}$, $f \neq 0$, is guaranteed by the Hahn-Banach theorem (Rudin, 1991). For a Hilbert space $\mathcal{H}$, for each element $f \in \mathcal{H}$, $f \neq 0$, there exists a unique peak functional $F = \langle f, \cdot \rangle_{\mathcal{H}} / \|f\|_{\mathcal{H}}$. Thus, every Hilbert space is a smooth Banach space. Donahue et al. (1997, Theorem 3.1) have shown that in smooth Banach spaces, and in particular in the Hilbert space $\mathcal{L}_{\rho}^{2}(X)$, an ε-greedy incremental sequence of functions can always be constructed. However, not every such sequence of functions converges to the function with respect to which it was constructed (Appendix D, Donahue et al., 1997). For the convergence to hold, a stronger notion of smoothness is needed.

Definition 3.3. *The modulus of smoothness of a Banach space $\mathcal{B}$ is a function $\tau \colon \mathbb{R}_0^+ \to \mathbb{R}_0^+$ such that*

$$\tau(r) = \frac{1}{2} \sup_{\|f\|=\|g\|=1} \Big(\|f + rg\| + \|f - rg\| \Big) - 1 \;,$$

where $f, g \in \mathcal{B}$. The Banach space $\mathcal{B}$ is said to be uniformly smooth if $\tau(r) \in o(r)$ as $r \to 0$.

We need to observe now that every Hilbert space is a uniformly smooth Banach space. For the sake of completeness, we provide a proof of this proposition.

Proposition 3.1. *(Donahue et al., 1997) For any Hilbert space the modulus of smoothness is equal to $\tau(r) = \sqrt{1 + r^2} - 1$.*

Proof. Expanding norms using the dot product we get

$$2(\tau(r) + 1) = \sup_{\|f\|=\|g\|=1} \left(\sqrt{1 + r^2 + 2r\langle f, g\rangle} + \sqrt{1 + r^2 - 2r\langle f, g\rangle} \right) .$$

Denoting with $u = 1 + r^2$ and $v = 2r\langle f, g\rangle$ and using the inequality between arithmetic and quadratic mean we get

$$\sqrt{u + v} + \sqrt{u - v} \leq 2\sqrt{\frac{u + v + u - v}{2}} = 2\sqrt{u} \;.$$

As the equality is attained for $v = 0$, it follows that the modulus of smoothness of a Hilbert space is given by

$$\tau(r) = \sqrt{1 + r^2} - 1 \;.$$

□

Having shown that Hilbert spaces are uniformly smooth Banach spaces, we proceed with two results specifying a convergence rate of an ε-greedy incremental sequence of functions. What is interesting about these results is the fact that a feature does not need to match exactly the residual function in a greedy descent step (Section 3.1.1); it is only required that condition (*ii*) from the next theorem is satisfied.

Theorem 3.2. *(Donahue et al., 1997) Let $\mathcal{B}$ be a uniformly smooth Banach space having modulus of smoothness $\tau(u) \leq \gamma u^t$, with γ being a constant and $t > 1$. Let S be a bounded subset of $\mathcal{B}$ and let $f \in \overline{\operatorname{co}(S)}$. Let $K > 0$ be chosen such that $\|f - g\| \leq K$ for all $g \in S$, and let $\varepsilon > 0$ be a fixed slack value. If the sequences $\{f_m\}_{m \geq 1} \subset \operatorname{co}(S)$ and $\{g_m\}_{m \geq 1} \subset S$ are chosen such that*

i) $f_1 \in S$,

ii*)* $F_m(g_m - f) \leq \frac{2\gamma\left((K+\varepsilon)^t - K^t\right)}{m^{t-1}\|f_m - f\|^{t-1}}$*, and*

iii*)* $f_{m+1} = \frac{m}{m+1} f_m + \frac{1}{m+1} g_m$*,*

where F_m *is the peak functional of* $f_m - f$*, then it holds*

$$\|f_m - f\| \leq \frac{(2\gamma t)^{\frac{1}{t}}(K+\varepsilon)}{m^{1-\frac{1}{t}}}\left[1 + \frac{(t-1)\log_2 m}{2tm}\right]^{\frac{1}{t}}.$$

The following corollary gives a convergence rate for an ε-greedy incremental sequence of functions constructed according to Theorem 3.2 with respect to $f_\rho \in \overline{\mathrm{co}(\mathcal{F}_\Theta)}$. As this result holds for all such sequences of functions, it also holds for our constructive procedure.

Corollary 3.3. *Let* $\{f_m\}_{m\geq 1} \subset \mathrm{co}(\mathcal{F}_\Theta)$ *be an* ε*-greedy incremental sequence of functions constructed according to the procedure described in Theorem 3.2 with respect to a function* $f \in \overline{\mathrm{co}(\mathcal{F}_\Theta)}$*. Then, it holds*

$$\|f_m - f\|_\rho \leq \frac{K+\varepsilon}{\sqrt{m}}\sqrt{2 + \frac{\log_2 m}{2m}}.$$

Proof. As $\mathcal{L}^2_\rho(X)$ is a Hilbert space, it follows from Proposition 3.1 that the modulus of smoothness of this space is $\tau(r) = \sqrt{1+r^2} - 1$. While it is straightforward to show that $\sqrt{1+r^2} \leq 1 + r$ for $r \in \mathbb{R}^+_0$, this bound is not tight enough as $r \to 0$. A tighter upper bound for this modulus of smoothness can be derived from the inequality $\sqrt{1+r^2} \leq 1 + \frac{r^2}{2}$. To see that this is a better bound for the case when $r \to 0$, it is sufficient to check that $1 + \frac{r^2}{2} \leq 1 + r$ for all $0 \leq r \leq 2$. Hence, all conditions of Theorem 3.2 are satisfied and the claim follows by setting $t = 2$ and $\gamma = \frac{1}{2}$. □

3.1.4 Generalization Bound

In step $t+1$ of the empirical residual fitting, based on a sample $\{(x_i, y_i)\}_{i=1}^n$ and the current estimate of the target function f_t, the approach selects a ridge-wave function from $\mathcal{F}_\Theta$ that approximates well the residual function $\left(f_\rho - f_t\right)$. In Section 3.1.3, we have specified in which cases such ridge-wave functions can be constructed and provided a convergence rate for this constructive procedure. As the convergence result is not limited to target functions from $\mathcal{F}_\Theta$ and $\mathrm{co}(\mathcal{F}_\Theta)$, we give a bound on the generalization error for hypotheses from $\mathcal{F} = \overline{\mathrm{co}(\mathcal{F}_\Theta)}$, where the closure is now taken with respect to $\mathcal{C}(X)$. In the remainder of this section, we present a proof of this result, organized into several parts/steps. In the first part, we introduce a notion of ε-capacity (Kolmogorov and Tikhomirov, 1959) to characterize the massiveness of a set in a metric space by means of the order at which the minimal number of disks/hyperspheres of radius ε covering that set increases as $\varepsilon \to 0$. In the second part, we quantify the ε-capacity of our hypothesis space by first showing that $\mathcal{F}$ is a convex and compact subset of the metric space $\mathcal{C}(X)$ and then bounding its ε-capacity using a result by Kolmogorov and Tikhomirov (1959). Following this, we provide two results by Cucker and Smale (2002) that guarantee the existence of the unique minimizer of the expected squared error on $\mathcal{F}$ as an element of $\mathcal{L}^2_\rho(X)$. Having established all the relevant auxiliary results, we give our generalization bound for the empirical squared error residual fitting.

For a subset A of a metric space and any $\varepsilon > 0$, we measure the complexity/capacity of A with the ε-covering number, given by the minimal number of disks of radius ε that cover A. Following Kolmogorov and Tikhomirov (1959), we denote the ε-covering number of A with $\mathcal{N}_{\varepsilon}(A; \|\cdot\|)$, where $\|\cdot\|$ is the metric function defined on A. The instance space X is a compact subset of a Euclidean space and, for all $\varepsilon > 0$, it has a finite ε-covering number. On the other hand, our ridge-wave basis function ϕ defined on X is a Lipschitz continuous function and Kolmogorov and Tikhomirov (1959) have given an upper bound on the ε-covering number of the set of 1-Lipschitz continuous functions defined on a compact set. For the sake of completeness, we present here their proof and later on use this result to show that $\mathcal{F}$ is a compact subset of $\mathcal{C}(X)$. However, before presenting this result we need to introduce the notion of a centralizable space, required in the proof.

Definition 3.4. *The space is called centralizable if in it, for any open set U of diameter $2R$, there exists a point x_0 from which any point x is at a distance no greater than R.*

Having introduced the notion of a centralizable space, we now give a result that bounds the ε-covering number of the set of 1-Lipschitz continuous functions defined on a compact set X, with respect to the metric space $\mathcal{C}(X)$.

Theorem 3.4. *(Kolmogorov and Tikhomirov, 1959) Let S be a connected totally bounded set which is contained in a centralizable space and let $\mathrm{Lip}_1(S)$ be a set of bounded 1-Lipschitz continuous functions on S. If all functions from $\mathrm{Lip}_1(S)$ are bounded by a constant $C > 0$, then*

$$\mathcal{N}_{\varepsilon}(\mathrm{Lip}_1(S); \|\cdot\|_{\infty}) \leq 2^{\mathcal{N}_{\frac{\varepsilon}{2}}(S; \|\cdot\|)}\left(2\left\lceil\frac{2C}{\varepsilon}\right\rceil + 1\right).$$

Proof. As the set S is totally bounded, then for any $\varepsilon > 0$ there exists a finite ε-covering of S. Let $\{U_i\}_{i=1}^{n}$ denote an $\frac{\varepsilon}{2}$-covering of the set S and let x_i be the center of the set U_i. Let $\hat{f}$ be an approximation of $f \in \mathrm{Lip}_1(S)$. Define $\hat{f}$ over the set U_1 with the value $\left\lceil\frac{2f(x_1)}{\varepsilon}\right\rceil\frac{\varepsilon}{2}$. Then, for all $x \in U_1$

$$\left|f(x) - \hat{f}(x)\right| = \left|f(x) - \hat{f}(x_1)\right| \leq \left|f(x) - f(x_1) + \frac{\varepsilon}{2}\right| \leq \varepsilon.$$

Setting $x = x_1$, we see that $\left|f(x_1) - \hat{f}(x_1)\right| \leq \frac{\varepsilon}{2}$.

On the other hand, for the center of the set U_i that is adjacent to U_1, $U_i \cap U_1 \neq \emptyset$, it holds

$$\left|f(x_i) - \hat{f}(x_1)\right| \leq |f(x_i) - f(x_1)| + \left|f(x_1) - \hat{f}(x_1)\right| \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

The first inequality follows from the properties of an $\frac{\varepsilon}{2}$-covering of S, i.e., if the neighboring disk centers are at distance more than $\frac{\varepsilon}{2}$ then there exists a point violating the definition of such covering. Thus, knowing the value at the center of U_1 with precision $\frac{\varepsilon}{2}$ suffices to approximate with precision ε the value at the centers of neighboring sets in the cover. From here it follows that by taking $\hat{f}(x) = \hat{f}(x_1) \pm \frac{\varepsilon}{2}$ for all $x \in U_i$, such that U_1 and U_i are adjacent, we can approximate $f(x_i)$ by $\hat{f}(x_i)$ with precision $\frac{\varepsilon}{2}$. As the space S is connected it is possible to connect any two non-adjacent sets U_i and U_j by a sequence of intersecting sets U_k. Hence, we can construct the entire functional $\hat{f}$ in this way and approximate the function f such that $\left\|f - \hat{f}\right\|_{\infty} \leq \varepsilon$.

Now, covering the range of these functions, $[-C, C]$, with ε-intervals we see that it is sufficient to take $2\left\lceil\frac{2C}{\varepsilon}\right\rceil + 1$ numbers as the center-values at x_1. For each of the sets in the $\frac{\varepsilon}{2}$-cover we have two choices and thus the ε-covering number of $\mathrm{Lip}_1(S)$ satisfies

$$\mathcal{N}_{\varepsilon}(\mathrm{Lip}_1(S); \|\cdot\|_{\infty}) \leq 2^{\mathcal{N}_{\frac{\varepsilon}{2}}(S; \|\cdot\|)}\left(2\left\lceil\frac{2C}{\varepsilon}\right\rceil + 1\right).$$

□

Theorem 3.4 provides an upper bound on the ε-covering number of the set of 1-Lipschitz continuous functions defined on a compact set as a function of the $\frac{\varepsilon}{2}$-covering number of that domain set (i.e., compact subset of a finite dimensional Euclidean space). The following result complements the latter bound by providing an upper bound on the ε-covering number of a disk containing the domain set of interest in its interior.

Proposition 3.5. *(Carl and Stephani, 1990) Let $\mathbb{E}$ be a finite dimensional Banach space and let B_R be the ball of radius R centered at the origin. Then, for $d = \dim(\mathbb{E})$*

$$\mathcal{N}_\varepsilon \left(B_R;\ \|\cdot\|\right) \leq \left(\frac{4R}{\varepsilon}\right)^d .$$

Having introduced all the preliminary results, we are now ready to show that our hypothesis space $\mathcal{F}$ is a convex and compact subset of the metric space $\mathcal{C}(X)$.

Proposition 3.6. *The hypothesis space $\mathcal{F}$ is a convex and compact subset of the metric space $\mathcal{C}(X)$. Moreover, the elements of this hypothesis space are Lipschitz continuous functions.*

Proof. Let $f, g \in \mathcal{F}$. As the hypothesis space $\mathcal{F}$ is the closure of the convex hull, $\operatorname{co}(\mathcal{F}_\Theta)$, it follows that there are sequences of functions $\{f_n\}_{n\geq 1}, \{g_n\}_{n\geq 1} \in \operatorname{co}(\mathcal{F}_\Theta)$ such that for every $\varepsilon > 0$ and sufficiently large n it holds $\|f - f_n\|_\infty < \varepsilon$ and $\|g - g_n\|_\infty < \varepsilon$. Then, for a convex combination of functions f and g and sufficiently large n we have

$$\|\alpha f + (1-\alpha) g - \alpha f_n - (1-\alpha) g_n\|_\infty \leq \alpha \|f - f_n\|_\infty + (1-\alpha)\|g - g_n\|_\infty < \varepsilon.$$

From here it follows that for every $0 \leq \alpha \leq 1$ and $f, g \in \mathcal{F}$ it holds $\alpha f + (1-\alpha) g \in \mathcal{F}$. Thus, we have showed that the hypothesis space $\mathcal{F}$ is a convex set.

As a convex combination of Lipschitz continuous functions is again a Lipschitz continuous function, we have that all functions $f \in \operatorname{co}(\mathcal{F}_\Theta)$ are Lipschitz continuous. It remains to prove that all functions from the closure are Lipschitz continuous, as well. Let f and $\{f_n\}_{n\geq 1}$ be defined as above and let L_ϕ be the Lipschitz constant of the function ϕ. We have that it holds

$$\left|f(x) - f(y)\right| \leq |f(x) - f_n(x)| + \left|f_n(x) - f_n(y)\right| + \left|f_n(y) - f(y)\right| < 2\|f - f_n\|_\infty + rL_\phi \left\|x - y\right\|.$$

Taking the limit of both sides as $n \longrightarrow \infty$, we deduce that function f is Lipschitz continuous with a Lipschitz constant bounded by rL_ϕ.

A metric space is compact if and only if it is complete and totally bounded (Rudin, 1991), i.e., for all $\varepsilon > 0$ there exists a finite ε-net of $\mathcal{F}$. Thus, as the hypothesis space $\mathcal{F}$ is complete by definition, it is sufficient to show that for all $\varepsilon > 0$ there exists a finite ε-net of $\mathcal{F}$ in $\mathcal{C}(X)$. The set X is a compact subset of a finite dimensional Euclidean space and as such it is totally bounded and contained in a centralizable space (see Definition 3.4 for details). Then, from Theorem 3.4 it follows that

$$\mathcal{N}_\varepsilon\left(\operatorname{Lip}_1(X);\ \|\cdot\|_\infty\right) \leq 2^{\mathcal{N}_{\frac{\varepsilon}{2}}(X;\ \|\cdot\|)}\left(2\left\lceil\frac{2C}{\varepsilon}\right\rceil + 1\right),$$

where $\operatorname{Lip}_1(X)$ denotes the set of 1-Lipschitz continuous functions defined on a set X and $C > 0$ is an upper bounds on all functions from $\operatorname{Lip}_1(X)$. This result allows us to bound the ε-covering number of the space of Lipschitz continuous functions on X. Namely, from the assumptions about $\mathcal{F}$ we conclude that all functions in $\mathcal{F}$ have Lipschitz constant bounded by $L_\mathcal{F} = rL_\phi$, where L_ϕ denotes the Lipschitz constant of the function ϕ. Then, an upper bound on the ε-covering number of the space $\operatorname{Lip}_{L_\mathcal{F}}(X)$ is given by

$$2^{\mathcal{N}_{\frac{\varepsilon}{2L_\mathcal{F}}}(X;\ \|\cdot\|_2)}\left(2\left\lceil\frac{2r}{\varepsilon}\right\rceil + 1\right).$$

As $\mathcal{F} \subset \operatorname{Lip}_{L_\mathcal{F}}(X)$ and $\mathcal{N}_\varepsilon\left(\operatorname{Lip}_{L_\mathcal{F}}(X);\ \|\cdot\|_\infty\right)$ is finite for all $\varepsilon > 0$, the result follows. □

The choice of a compact hypothesis space is important because it guarantees that a minimizer of the expected squared error $\mathcal{E}_\rho$ and its empirical counterpart $\mathcal{E}_{\mathbf{z}}$ exists. In particular, a continuous function attains its minimum and maximum value on a compact set and this guarantees the existence of minimizers of $\mathcal{E}_\rho$ and $\mathcal{E}_{\mathbf{z}}$. The following result by Cucker and Smale (2002) shows that the functionals $\mathcal{E}_\rho$ and $\mathcal{E}_{\mathbf{z}}$ are continuous on $\mathcal{F}$.

Proposition 3.7. *(Cucker and Smale, 2002) Let $f_1, f_2 \in \mathcal{C}(X)$, $M \in \mathbb{R}_+$, and $\left|f_i(x) - y\right| \leq M$ on a set $U \subset Z$ of full measure for $i = 1, 2$. Then for all $\mathbf{z} \in U^n$ functions $\mathcal{E}_\rho$ and $\mathcal{E}_{\mathbf{z}}$ are Lipschitz continuous on the metric space $\mathcal{C}(X)$.*

Proof. For all $f_1, f_2 \in \mathcal{C}(X)$ we have that

$$\left|(f_1(x) - y)^2 - (f_2(x) - y)^2\right| = |f_1(x) - f_2(x)| \left|f_1(x) - y + f_2(x) - y\right| \leq 2M \|f_1 - f_2\|_\infty ,$$

and the claim follows from this inequality. □

In addition to this, for a hypothesis space that is both convex and compact, the minimizer of the expected squared error is *unique as an element of* $\mathcal{L}^2_\rho(X)$. A simple proof of the uniqueness of such a minimizer in $\mathcal{L}^2_\rho(X)$ can also be found in Cucker and Smale (2002). For the sake of completeness, we provide here a proof of this result.

Proposition 3.8. *(Cucker and Smale, 2002) Let $\mathcal{K}$ be a convex and compact subset of $\mathcal{C}(X)$. Then there exists a function in $\mathcal{C}(X)$ with a minimal distance to f_ρ in $\mathcal{L}^2_\rho(X)$. Moreover, this function is unique as an element of $\mathcal{L}^2_\rho(X)$.*

Proof. From the compactness of the subspace it follows that a minimizer exists. However, it does not have to be unique. Let f_1 and f_2 be two minimizers and let

$$\mathcal{S} = \{\alpha f_1 + (1 - \alpha) f_2 \mid 0 \leq \alpha \leq 1\}$$

be the line segment connecting these two points. As the subspace $\mathcal{K}$ is convex, then the segment $\mathcal{S}$ is contained within $\mathcal{K}$. Furthermore, for all $f \in \mathcal{S}$, it holds

$$\left\|f_1 - f_\rho\right\|_\rho = \left\|f_2 - f_\rho\right\|_\rho \leq \left\|f - f_\rho\right\|_\rho .$$

From the inequality for the first term, we have

$$\left\langle f_\rho - f_1, f - f_1\right\rangle_\rho + \left\langle f_\rho - f_1, f_\rho - f\right\rangle_\rho \leq \left\|f_\rho - f\right\|^2_\rho \Rightarrow \left\langle f_\rho - f_1, f - f_1\right\rangle_\rho \leq \left\langle f_1 - f, f_\rho - f\right\rangle_\rho .$$

Similarly, from the inequality for the second term, we obtain

$$\left\langle f_\rho - f_2, f - f_2\right\rangle_\rho \leq \left\langle f_2 - f, f_\rho - f\right\rangle_\rho .$$

As the cosine is a decreasing function over $[0, \pi]$, it follows that $\angle f_\rho f_1 f \geq \angle f_\rho f f_1$ and $\angle f_\rho f_2 f \geq \angle f_\rho f f_2$ for all $f \in \mathcal{S}$. Hence, if $f_1 \neq f_2$ then the angles $\angle f_\rho f_1 f$ and $\angle f_\rho f_2 f$ are obtuse. As there does not exist a triangle with two obtuse angles, then $f_1 = f_2$. □

Having established that the hypothesis space is a compact and convex set with a unique minimizer of the expected squared error, we can now give a generalization bound for learning on this hypothesis space. In particular, the fact that $\mathcal{F}$ is compact allows us to derive a sample complexity bound by using the ε-covering number of the space as a measure of its capacity (Kolmogorov and Tikhomirov, 1959). The following theorem and corollary give a generalization bound for learning on the hypothesis space $\mathcal{F}$.

Theorem 3.9. *Let $M > 0$ be a finite constant such that, for all $f \in \mathcal{F}$, $\left|f(x) - y\right| \leq M$ almost everywhere. Then, for all $\varepsilon > 0$,*

$$\mathbb{P}_{\mathbf{z}\in Z^n}\left[\mathcal{E}_\rho(f_{\mathbf{z}}) - \mathcal{E}_\rho(f^*) \leq \varepsilon\right] \geq 1 - \mathcal{N}_{\frac{\varepsilon}{24M}}(\mathcal{F};\ \|\cdot\|_\infty)\exp\left(-\frac{n\varepsilon}{288M^2}\right),$$

where $f_{\mathbf{z}}$ and f^* are the minimizers of $\mathcal{E}_{\mathbf{z}}$ and $\mathcal{E}_\rho$ on the set $\mathcal{F}$.

Our proof of the theorem relies on the following result by Cucker and Smale (2002).

Theorem 3.10. *(Cucker and Smale, 2002) Let $\mathcal{K}$ be a compact and convex subset of $\mathcal{C}(X)$ and let $M > 0$ be a finite constant such that for all $f \in \mathcal{K}$, $\left|f(x) - y\right| \leq M$ almost everywhere. Then, for all $\varepsilon > 0$,*

$$\mathbb{P}_{\mathbf{z}\in Z^n}\left[\mathcal{E}_\rho(f_{\mathbf{z}}) - \mathcal{E}_\rho(f_{\mathcal{K}}) \leq \varepsilon\right] \geq 1 - \mathcal{N}_{\frac{\varepsilon}{24M}}(\mathcal{K};\ \|\cdot\|_\infty)\exp\left(-\frac{n\varepsilon}{288M^2}\right),$$

where $f_{\mathbf{z}}$ and $f_{\mathcal{K}}$ are the minimizers of $\mathcal{E}_{\mathbf{z}}$ and $\mathcal{E}_\rho$ over $\mathcal{K}$.

Proof of Theorem 3.9. The claim follows from Proposition 3.6 and Theorem 3.10. □

Corollary 3.11. *For all $\varepsilon, \delta > 0$, with probability $1 - \delta$, a minimizer of the empirical squared error on the hypothesis space $\mathcal{F}$ is (ε, δ)-consistent when the number of samples*

$$n \in \Omega\left(r(Rs + t)L_\phi\frac{1}{\varepsilon^2} + \frac{1}{\varepsilon}\ln\frac{1}{\delta}\right).$$

Here, R is the radius of a ball containing the set of instances X in its interior, L_ϕ is the Lipschitz constant of a function ϕ, and r, s, and t are hyperparameters of $\mathcal{F}_\Theta$.

Proof. To derive the sample complexity bound from this corollary, we need a tighter bound on the ε-covering number of our hypothesis space than the one provided in Proposition 3.6. We first give one such bound and then prove the corollary.

The set of instances X is a compact subset of a Euclidean space and we can, without loss of generality, assume that there exists a ball of radius R centered at the origin and containing the set X in its interior. From the definition of the hypothesis space $\mathcal{F}$ we see that the argument of the ridge function ϕ is bounded, i.e.,

$$|\langle w, x\rangle + b| \leq \|w\|\|x\| + t \leq Rs + t.$$

From here we conclude that the hypothesis space $\mathcal{F}$ is a subset of the space of 1-dimensional Lipschitz continuous functions on the compact interval $[-(Rs + t), Rs + t]$. Then, the covering number of $\mathcal{F}$ is upper bounded by the covering number of the space of $L_{\mathcal{F}}$-Lipschitz continuous one dimensional functions defined on the segment $[-(Rs + t), Rs + t]$. From Proposition 3.5, it follows that the ε-covering number of the segment $[-(Rs + t), Rs + t]$ is upper bounded by ${}^{4(Rs+t)}\!/\!{}_{\varepsilon}$. This, together with Theorem 3.4 implies that the upper bound on the ε-covering number of the hypothesis space $\mathcal{F}$ is given by

$$\mathcal{N}_\varepsilon(\mathcal{F};\ \|\cdot\|_\infty) \leq 2^{\frac{8r(Rs+t)L_\phi}{\varepsilon}}\left(2\left\lceil\frac{2r}{\varepsilon}\right\rceil + 1\right). \tag{3.1}$$

On the other hand, from Theorem 3.9 we obtain that for all $\delta > 0$ with probability $1 - \delta$ the empirical estimator is (ε, δ)-consistent when

$$2^{\frac{192r(Rs+t)ML_\phi}{\varepsilon}}\left(2\left\lceil\frac{48Mr}{\varepsilon}\right\rceil+1\right)\exp\left(-\frac{n\varepsilon}{288M^2}\right)\leq\delta \quad\Longleftrightarrow$$
$$\frac{192r(Rs+t)ML_\phi}{\varepsilon}\ln 2+\ln\left(2\left\lceil\frac{48Mr}{\varepsilon}\right\rceil+1\right)\leq\frac{n\varepsilon}{288M^2}-\ln\frac{1}{\delta}\ .$$

Hence, for all $\varepsilon, \delta > 0$ and

$$n \geq \frac{288M^2}{\varepsilon}\left[\frac{192r(Rs+t)ML_\phi}{\varepsilon}\ln 2+\ln\left(2\left\lceil\frac{48Mr}{\varepsilon}\right\rceil+1\right)+\ln\frac{1}{\delta}\right] \tag{3.2}$$

with probability $1 - \delta$ the empirical estimator is (ε, δ)-consistent. □

The concentration inequality from Theorem 3.10 is tighter by a factor of $1/\varepsilon$ for convex and compact compared to compact only hypothesis spaces. For instance, this can be seen by comparing the bounds from the corresponding theorems in Cucker and Smale (2002). In our case with convex and compact hypothesis space $\mathcal{F}$, the sample complexity bound from Theorem 3.9 is still $\Omega\left(1/\varepsilon^2\right)$ due to the $1/\varepsilon$ factor coming from the ε-covering number of $\mathcal{F}$.

We conclude this section by noting that a detailed study of the properties of ridge basis functions in high dimensional Euclidean spaces can be found in Mayer et al. (2015).

3.1.5 Algorithm

Algorithm 3.1 is a pseudo-code description of the outlined approach. To construct a feature space with a good set of linear hypotheses the algorithm takes as input a set of labeled instances and an initial empirical estimate of the target function. A dictionary of features is specified with a ridge basis function and the smoothness of individual features is controlled with a regularization parameter. Other parameters of the algorithm are the maximum allowed number of descent steps and a precision term that defines the convergence of the descent. As outlined in Sections 3.1.1 and 3.1.3, the algorithm works by selecting a feature that matches the residual function at the current estimate of the target function. For each selected feature the algorithm also chooses a suitable learning rate and performs a functional descent step (note that we are inferring the learning rate instead of setting it to $1/m+1$ as in Theorem 3.2). To avoid solving these two problems separately, we have coupled both tasks into a single optimization problem (line 3). In particular, we fit a linear model to a feature representation consisting of the current empirical estimate of the target function and a ridge function parameterized with a d-dimensional vector w. The regularization term Ω is chosen to control the smoothness of the new feature and avoid over-fitting. The optimization problem over the coefficients of the linear model and the spectrum of the ridge basis function is solved by casting it as a hyperparameter optimization problem (see Section 3.2.2 or Keerthi et al., 2007).

While the hyperparameter optimization problem is in general non-convex, Theorem 3.2 indicates that a globally optimal solution is not (necessarily) required and instead specifies a weaker condition. To account for the non-convex nature of the problem and compensate for the sequential generation of features, we propose to parallelize the feature construction process by running several instances of the greedy descent simultaneously. A pseudo-code description of this parallelized approach is given in Algorithm 3.2. The algorithm takes as input parameters required for running the greedy descent and some parameters specific to the

Algorithm 3.1 GreedyDescent

```
Input: sample $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^{s}$, initial estimates at sample points $\{f_{0,i}\}_{i=1}^{s}$, ridge basis function $\phi$,
    maximum number of descent steps $p$, regularization parameter $\lambda$, and precision $\varepsilon$
W ← $\emptyset$
for $k = 1, 2, \ldots, p$ do
    $w_k, c_k \leftarrow \operatorname{argmin}_{w, c=(c', c'')} \sum_{i=1}^{s} \left(c' f_{k-1,i} + c'' \phi\left(w^\top x_i\right) - y_i\right)^2 + \lambda \Omega\left(c, w\right)$
    $W \leftarrow W \cup \{w_k\}$ and $f_{k,i} \leftarrow c'_k f_{k-1,i} + c''_k \phi\left(w_k^\top x_i\right)$, $i = 1, \ldots, s$
    if $|\mathcal{E}_{\mathbf{z}}(f_k) - \mathcal{E}_{\mathbf{z}}(f_{k-1})| / \max\{\mathcal{E}_{\mathbf{z}}(f_k), \mathcal{E}_{\mathbf{z}}(f_{k-1})\} < \varepsilon$ then Exit For Loop end if
end for
return $W$
```

Algorithm 3.2 Greedy Feature Construction (gfc)

```
Input: sample $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^{n}$, ridge basis function $\phi$, number of data passes $T$, maximum number
    of greedy descent steps $p$, number of machines/cores $M$, regularization parameters $\lambda$ and $\nu$,
    precision $\varepsilon$, and feature cut-off threshold $\eta$
1: $W \leftarrow \{\mathbf{0}\}$ and $f_{0,k} \leftarrow \frac{1}{n} \sum_{i=1}^{n} y_i, k = 1, \ldots, n$
2: for $i = 1, \ldots, T$ do
3:     for $j = 1, 2, \ldots, M$ parallel do
4:         $S_j \sim \mathcal{U}_{\{1,2,\ldots,n\}}$ and $W \leftarrow W \cup \text{GreedyDescent}\left(\{(x_k, y_k)\}_{k \in S_j}, \{f_{i-1,k}\}_{k \in S_j}, \phi, p, \lambda, \varepsilon\right)$
5:     end for
6:     $a^* \leftarrow \operatorname{argmin}_a \sum_{k=1}^{n} \left(\sum_{l=1}^{|W|} a_l \phi\left(w_l^\top x_k\right) - y_k\right)^2 + \nu \|a\|_2^2$
7:     $W \leftarrow W \setminus \left\{w_l \in W \mid |a_l^*| < \eta, 1 \leq l \leq |W|\right\}$ and $f_{i,k} \leftarrow \sum_{l=1}^{|W|} a_l^* \phi\left(w_l^\top x_k\right)$, $k = 1, \ldots, n$
8: end for
9: return $(W, a^*)$
```

parallelization scheme: number of data passes and available machines/cores, regularization parameter for the fitting of linear models in the constructed feature space, and cut-off parameter for the elimination of redundant features. The whole process is started by adding a bias feature and setting the initial empirical estimates at sample points to the mean value of the outputs (line 1). Following this, the algorithm mimics stochastic gradient descent and makes a specified number of passes through the data (line 2). In the first step of each pass, the algorithm performs greedy functional descent in parallel using a pre-specified number of machines/cores M (lines 3-5). This step is similar to the splitting step in parallelized stochastic gradient descent (Zinkevich et al., 2010). Greedy descent is performed on each of the machines for a maximum number of iterations p and the estimated parameter vectors are added to the set of constructed features W (line 4). After the features have been learned the algorithm fits a linear model to obtain the amplitudes (line 6). To fit a linear model, we use least square regression penalized with the l_2-norm because it can be solved in a closed form and cross-validation of the capacity parameter involves optimizing a 1-dimensional objective function (e.g., see Section 3.2.2 or Keerthi et al., 2007). Fitting of the linear model can be understood as averaging of the greedy approximations constructed on different chunks of the data. At the end of each pass, the empirical estimates at sample points are updated and redundant features are removed (line 7).

One important detail in the implementation of Algorithm 3.1 is the data splitting between the training and validation samples for the hyperparameter optimization. In particular, during the descent we are more interested in obtaining a good spectrum than the amplitude because a linear model will be fit in Algorithm 3.2 over the constructed features and the amplitude values will be updated. For this reason, during the hyperparameter optimization over a k-fold splitting in Algorithm 3.1, we propose to choose a single fold as the training sample and a batch of folds as the validation sample.

3.2 Learning with Fourier Features

In this section, we present an instance of our approach with a set of Fourier features (Rahimi and Recht, 2008a) as ridge-wave bases. We start by introducing a notion of Fourier feature and then review the relation between these features and stationary kernels (Section 3.2.1), the class of kernel functions that corresponds to a rich set of hypotheses (Micchelli et al., 2006). In Section 3.2.2, we exploit the relationship between Fourier features and stationary kernels to show that this particular instance of our approach can approximate any bounded hypothesis from any stationary reproducing kernel Hilbert space. Following this, we give a detailed description of the proposed procedure for solving the optimization problem for greedy feature construction with Fourier features as ridge bases (i.e., optimization problem in line 3 of Algorithm 3.1). The section concludes with a review of a related, alternative approach for learning with Fourier features (Section 3.2.3), that contrary to our approach approximates a target regression function by optimizing jointly over an efficiently parameterized set of Fourier features and corresponding regression coefficients.

3.2.1 Fourier Features

Fourier features were first introduced to the machine learning community by Rahimi and Recht (2008a) as a mean to approximate stationary kernel functions and scale kernel methods to large scale datasets with millions of instances. This section reviews that work and a relation between Fourier features and a class of high-capacity kernel functions known as stationary kernels (Section 3.2.1.1). Following this, we provide a brief overview of a frequently used approach for learning with random Fourier features, random kitchen sinks (Sections 3.2.1.2). The approach serves as one of the baselines in the empirical study of the effectiveness of our greedy feature construction approach (e.g., see Section 3.3).

3.2.1.1 Stationary Kernels

This section provides a brief review of a class of positive definite kernel functions known as stationary kernels. The definitions and terminology used throughout the section follow along the lines of Genton (2002).

Let $D \subset \mathbb{R}^d$ be an open set. A positive definite kernel $k\colon D \times D \to \mathbb{R}$ is called stationary or *anisotropic* if there exists a function $s\colon D \to \mathbb{R}$ such that $k(x, x') = s(x - x')$, for all $x, x' \in D$. Alternatively, a function $s\colon D \to \mathbb{R}$ is said to be of positive type if there exists a positive definite kernel $k\colon D \times D \to \mathbb{R}$ such that $s(x - x') = k(x, x')$ for all $x, x' \in D$. Thus, a stationary kernel function depends only on the *lag vector* separating two instances x and x' and not on the instances themselves. A large number of stationary positive definite kernels can be derived from their spectral representation given by Bochner (1932).

Theorem 3.12. *(Bochner, 1932) The Fourier transform of a bounded positive measure on $\mathbb{R}^d$ is a continuous function of positive type. Conversely, any function of positive type is the Fourier transform of a bounded positive measure.*

This theorem implies that any stationary positive definite kernel k satisfies

$$k(x,x') = s(x-x') = \int_{\mathbb{R}^d} \exp(-i\langle w, x-x'\rangle)\, d\mu(w) \ ,$$

where μ is a positive and bounded measure. The quantity ${}^{\mu}\!/_{s(0)}$ is called the spectral distribution function. Now, as $k(x,x')$ is a real function in both arguments, the complex part in the integral on the right-hand side is equal to zero, and we have that (Rahimi and Recht, 2008a)

$$k(x,x') = 2\int_{\mathbb{R}^d\times[-\pi,\pi]} \cos\left(w^\top x + b\right)\cos\left(w^\top x' + b\right) d\hat{\mu}(w,b) \ , \tag{3.3}$$

where $\hat{\mu}(w,b) = {}^{\mu(w)}\!/_{2\pi} > 0$ for all $w \in \mathbb{R}^d$ and $b \in [-\pi,\pi]$.

A stationary kernel function that depends only on the distance between two instances is called *isotropic* or homogeneous. The spectral representation of isotropic stationary kernels has been derived by Yaglom (1957). In particular, for any isotropic kernel it holds

$$k(x,x') = s\left(\left\|x-x'\right\|\right) = \int_0^\infty \Omega_d\left(w\left\|x-x'\right\|\right) d\mu(w) \ ,$$

where

$$\Omega_d(x) = \left(\frac{2}{x}\right)^{\frac{(d-2)}{2}} \cdot \Gamma\left(\frac{d}{2}\right) \cdot J_{\frac{(d-2)}{2}}(x)$$

form a basis for kernel functions in $\mathbb{R}^d$ (Genton, 2002). Here, μ is some non-decreasing bounded function, $\Gamma({}^{d}\!/_{2})$ is the gamma function, and $J_{{}^{(d-2)}\!/_{2}}(x)$ is the Bessel function of the first kind of order ${}^{(d-2)}\!/_{2}$. As pointed by Genton (2002), all isotropic stationary kernels obtained with Ω_d are positive definite in $\mathbb{R}^d$ and in lower dimensions, but not necessarily in higher dimensions. Stein (1999) has provided a lower bound on isotropic stationary kernels,

$$k(x,x') = s\left(\left\|x-x'\right\|\right) \geq s(0) \cdot \inf_{x\geq 0} \Omega_d(x) \ .$$

From this lower bound and properties of Ω_d one can observe that isotropic stationary kernels fall off quickly as the dimension of the instance space increases (Stein, 1999; Genton, 2002). Schönberg (1938, Section 2) showed that if $\mathcal{B}_d$ is the class of isotropic positive definite kernels in $\mathbb{R}^d$, then the classes for all d have the property

$$\mathcal{B}_\infty \subset \cdots \subset \mathcal{B}_d \subset \cdots \subset \mathcal{B}_2 \subset \mathcal{B}_1 \ .$$

As $\Omega_d(x) \to \exp(-x^2)$ when $d \to \infty$, then only isotropic positive definite kernels with basis function $\exp(-x^2)$ are contained in all the classes. This observation implies that as the dimension of the instance space increases, the space of available isotropic positive definite kernels reduces (Schönberg, 1938; Genton, 2002).

An isotropic positive definite kernel that is frequently used with kernel methods in machine learning is the Gaussian or squared exponential kernel. For this particular kernel, from Theorem 3.12 we obtain that the spectral distribution is also Gaussian. More formally,

$$k(x,x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right) = \left(\frac{\sigma^2}{2\pi}\right)^{{}^{d}\!/_{2}} \int_{\mathbb{R}^d} \exp(-i\langle w, x-x'\rangle)\exp\left(-\frac{\sigma^2\|w\|^2}{2}\right) dw \ .$$

Thus, for the Gaussian kernel we have that $s(0) = 1$ and the spectral distribution is given by $\mu(w) = \sigma \exp\left(-\sigma^2 \|w\|^2 / 2\right) / \sqrt{2\pi}$. Beside the Gaussian kernel, Genton (2002) has provided several other isotropic positive definite kernels used with kernel methods such as the Laplace or exponential kernel. Similar to the Gaussian kernel, we can compute the spectral density corresponding to the Laplace kernel using Theorem 3.12. More specifically, we have that it holds (Rahimi and Recht, 2008a)

$$k(x, x') = \exp\left(-\frac{\|x - x'\|}{\theta}\right) = \int_{\mathbb{R}^d} \exp\left(-i \langle w, x - x' \rangle\right) \prod_{i=1}^{d} \frac{\theta / \pi}{1 + \theta^2 w_i^2} \, dw \, .$$

Thus, for the Laplace kernel the spectral distribution is given by the product of one dimensional Cauchy distributions, $\mu(w) = \prod_{i=1}^{d} \frac{\theta}{\pi} / (1 + \theta^2 w_i^2)$.

This review of stationary kernels concludes with the Matérn kernel (Matérn, 1986) that has been advocated recently for its ability to control the smoothness of hypotheses via kernel parameters (e.g., see Le et al., 2013). The kernel is formally defined as (Genton, 2002)

$$k(x, x') = \frac{1}{2^{\nu - 1} \Gamma(\nu)} \left(\frac{2\sqrt{\nu} \|x - x'\|}{\theta}\right)^{\nu} H_{\nu}\left(\frac{2\sqrt{\nu} \|x - x'\|}{\theta}\right),$$

where Γ is the Gamma function and H_ν is the modified Bessel function of the second kind of order ν. For $\nu = 1/2$ the Matérn kernel reduces to the Laplace kernel and for $\nu \longrightarrow \infty$ to the Gaussian kernel. As pointed by Rasmussen and Williams (2005), perhaps the most interesting cases for machine learning community are $\nu = 3/2$ and $\nu = 5/2$, for which

$$\begin{aligned} k_{\nu = 3/2}(x, x') &= \left(1 + \frac{\sqrt{3} \|x - x'\|}{\theta}\right) \exp\left(-\frac{\sqrt{3} \|x - x'\|}{\theta}\right) \\ k_{\nu = 5/2}(x, x') &= \left(1 + \frac{\sqrt{5} \|x - x'\|}{\theta} + \frac{5 \|x - x'\|^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5} \|x - x'\|}{\theta}\right) . \end{aligned}$$

For $\nu = 1/2$, hypotheses are rough and not necessarily mean squared differentiable (e.g., see Section 4.2.1 in Rasmussen and Williams, 2005), and for $\nu \geq 7/2$ combined with a noisy set of examples it is not easy to distinguish between hypotheses corresponding to the Matérn kernel with finite ν and that with $\nu \longrightarrow \infty$ (i.e., the Gaussian or squared exponential kernel).

In the remainder of the chapter, we will use the term stationary/isotropic kernel for kernels that are both stationary/isotropic and positive definite.

3.2.1.2 Random Kitchen Sinks

Having introduced stationary kernels, we now review an efficient learning approach capable of approximating and scaling kernel machines with stationary kernels to datasets with millions of instances. The materials and terminology in this section follow along the lines of works by Rahimi and Recht (2008a,b, 2009).

In their seminal work, Rahimi and Recht (2008a) have introduced *Fourier features* as $\zeta_{w,b}(x) = \sqrt{2} \cos(w^\top x + b)$ with $w \in \mathbb{R}^d$ and $-\pi \leq b \leq \pi$. Fourier features, together with Theorem 3.12, allow one to express the kernel value between instances x and x' as

$$k(x,x') / s(0) = \mathbb{E}_{(w,b) \sim \frac{\mu(w)}{s(0)} \times \mathcal{U}_{[-\pi,\pi]}} \left[\zeta_{w,b}(x) \zeta_{w,b}(x')\right] \, .$$

For simplicity of the presentation, in the remainder of the section we assume that $s(0) = 1$ (i.e., μ is a probability measure). Thus, we can approximate the kernel value at instances x and

x' using a Monte Carlo estimate of the expectation with respect to random variables $w \sim \mu$ and $b \sim \mathcal{U}_{[-\pi,\pi]}$. More formally, if we let $\mathcal{S}_m = \{(w_1, s_1), \dots, (w_m, b_m)\}$ be an independent sample from $\mu \times \mathcal{U}_{[-\pi,\pi]}$ and denote with $\zeta(x \mid \mathcal{S}_m) = {}^{\mathrm{vec}(\zeta_{w_1,b_1}(x),\dots,\zeta_{w_m,b_m}(x))}/_{\sqrt{m}}$, then

$$\zeta(x \mid \mathcal{S}_m)^\top \zeta(x' \mid \mathcal{S}_m) = \frac{1}{m} \sum_{i=1}^{m} \zeta_{w_i,b_i}(x)\, \zeta_{w_i,b_i}(x')$$

is an approximation of the expectation in Eq. (3.3). For any two fixed instances and a stationary kernel, the concentration bound follows from the Hoeffding's inequality.

Proposition 3.13. *(Rahimi and Recht, 2008a) For $x, x' \in \mathbb{R}^d$ and a sample $\mathcal{S}_m$ from the spectral distribution $\mu \times \mathcal{U}_{[-\pi,\pi]}$ of a stationary positive definite kernel k, we have that it holds*

$$P\left(\left| \zeta(x \mid \mathcal{S}_m)^\top \zeta(x' \mid \mathcal{S}_m) - k(x,x') \right| \geq \varepsilon \right) \leq 2 \exp\left(-\frac{m\varepsilon^2}{4} \right) .$$

This result holds only pointwise, that is for a fixed pair of instances x and x' and a given sample of spectral parameters $\mathcal{S}_m$. Rahimi and Recht (2008a) have extended the bound from Proposition 3.13 to a uniform bound holding for any two instances from a compact set X using a standard ε-net argument (e.g., see Kolmogorov and Tikhomirov, 1959). The following theorem is a formal statement of that much stronger result.

Theorem 3.14. *(Rahimi and Recht, 2008a) Let X be a compact subset of $\mathbb{R}^d$ with diameter R. Then, for the mapping $\zeta(x \mid \mathcal{S}_m)$ defined above, we have*

$$P\left(\sup_{x,x' \in X} \left| \zeta(x \mid \mathcal{S}_m)^\top \zeta(x' \mid \mathcal{S}_m) - k(x,x') \right| \geq \varepsilon \right) \leq 256 \left(\frac{R\gamma}{\varepsilon} \right)^2 \exp\left(-\frac{m\varepsilon^2}{4(d+2)} \right) , \qquad (3.4)$$

where $\gamma = \mathbb{E}_w \left[\|w\|^2 \right]$ is the second moment of the Fourier transform of k. Moreover,

$$\sup_{x,x' \in X} \left| \zeta(x \mid \mathcal{S}_m)^\top \zeta(x' \mid \mathcal{S}_m) - k(x,x') \right| \leq \varepsilon$$

with any constant probability when $m = \Omega\left(\frac{d}{\varepsilon^2} \log \frac{R\gamma}{\varepsilon} \right)$.

The constant γ^2 quantifies the curvature of the kernel at the origin and it can be computed using standard Fourier identities. For example, for the Gaussian kernel Rahimi and Recht (2008a) have computed that $\gamma^2 = {}^{d}/_{\sigma^2}$. The computational and space complexities of transforming a dataset with n instances from $\mathbb{R}^d$ to a random Fourier feature representation with m features are $\mathcal{O}(nmd)$. Le et al. (2013) have proposed an approach, *Fastfood*, that can speed up such transformations for the Gaussian and Matérn kernels. The computational complexity of that approach is $\mathcal{O}(nm \log d)$ and is especially effective for problems with high dimensional instance spaces. However, this improvement in computational complexity comes at the cost of a slightly worst concentration bound for the approximation. In particular, the result by Le et al. (2013) for the pointwise concentration (i.e., directly comparable to Proposition 3.13) can be formally stated as follows.

Theorem 3.15. *(Le et al., 2013) For $x, x' \in \mathbb{R}^d$ let $\hat{k}(x,x')$ denote the approximation of the Gaussian kernel, denoted with $k(x,x')$, that arises from a Fastfood block of size $m \times m$ with $m = d$. Then, for all $\delta > 0$*

$$P\left(\left| \hat{k}(x,x') - k(x,x') \right| \geq 2 \frac{\|x - x'\|}{\sigma} \sqrt{\frac{\log {}^{2m}/_{\delta} \log {}^{2}/_{\delta}}{m}} \right) \leq 2\delta .$$

This pointwise result can be extended to a concentration bound over a compact set X by mimicking the ε-net argument from Rahimi and Recht (2008a). A direct comparison to Theorem 3.14 then implies that the error of the Fastfood approximation is at most $\mathcal{O}\left(\sqrt{\log {}^{m}\!/\!_{\delta}}\right)$ times larger than that of random Fourier features (Le et al., 2013). Moreover, for the approximation bound from Theorem 3.14 that holds with any constant probability, the Fastfood approach requires at least $\log {}^{1}\!/\!_{\delta}$ times more Fourier features.

The results from Theorems 3.14 and 3.15 quantify the approximation quality of random Fourier features in the task of approximating stationary kernels. To learn with such approximate stationary kernels, one can first transform data to a representation given by random Fourier features and then apply a linear learning algorithm such as linear regression or support vector machine to the resulting representation. The computational complexity of learning with random Fourier features in that case is linear in the number of instances and cubic in the number of Fourier features. A moderate number of Fourier features then allows kernel methods to scale to millions of instances. To quantify the effectiveness of such approximations, Rahimi and Recht (2008b, 2009) have provided generalization bound and approximation properties of learning with random Fourier features. Their results relate learning with random features to kernel methods with stationary kernels. In the remainder of this section, we review these results and in Section 3.2.2 provide similar bounds to quantify the approximation properties of our greedy feature construction approach.

In our review of random kitchen sinks (Rahimi and Recht, 2008b), we follow the approach by Rahimi and Recht (2008b, 2009) and instead of Fourier features use a more general basis function $\phi\colon X \times \Theta \to \mathbb{R}$ with $\Theta \subseteq \mathbb{R}^d$. In summary, random kitchen sinks first sample spectral parameters $\{\theta_i\}_{i=1}^m$ independently from a probability density function p such that

$$k(x,x') = \int_{\Theta} \phi(x,\theta)\,\phi(x',\theta)\,p(\theta)\,d\theta\ .$$

Then, the algorithm solves the following convex optimization problem

$$\begin{aligned} \alpha^* = \quad & \min_{\alpha\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n} l\left(\alpha^\top \phi(x_i), y_i\right) \\ & \|\alpha\|_\infty \leq \frac{C}{m}\ , \end{aligned} \tag{3.5}$$

where $\phi(x) = \operatorname{vec}(\phi(x,\theta_1),\ldots,\phi(x,\theta_m))$ and $l(y,y') = l(yy')$ is an L-Lipschitz continuous loss function. In practice, the uniform norm regularization from the latter optimization problem is replaced with the squared norm regularization. Such relaxation of the random kitchen sinks optimization problem is equivalent to learning with linear ridge regression/support vector machine. For a fixed spectral distribution p, the hypothesis space of random kitchen sinks is given by (Rahimi and Recht, 2008b, 2009)

$$\mathcal{F}_p = \left\{ f(x) = \int_{\Theta} \alpha(\theta)\,\phi(x,\theta)\,d\theta \;\middle|\; |\alpha(\theta)| \leq C p(\theta) \right\}\ , \tag{3.6}$$

with a constant $C \in \mathbb{R}^+$. For learning with random kitchen sinks on this hypothesis space Rahimi and Recht (2009) have given the following generalization bound.

Theorem 3.16. *(Rahimi and Recht, 2009) Let p be a probability density function defined on $\Theta \in \mathbb{R}^d$ and let $\sup_{x\in X,\theta\in\Theta} \left|\phi(x,\theta)\right| \leq 1$. If the training data $\{(x_i,y_i)\}_{i=1}^n$ are drawn iid from a* *probability measure ρ, then, for all $\delta > 0$, the random kitchen sinks algorithm returns a function $f_{n,m}$ such that*

$$\mathbb{E}_{(x,y)\sim\rho}\left[l\left(f_{n,m}(x), y\right)\right] - \min_{f\in\mathcal{F}_p} \mathbb{E}_{(x,y)\sim\rho}\left[l\left(f(x), y\right)\right] \in \mathcal{O}\left(LC\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right)\sqrt{\log\frac{1}{\delta}}\right)$$

with probability at least $1-2\delta$ over the training dataset and the choice of the parameters $\{\theta_i\}_{i=1}^m$.

Moreover, for learning with squared error loss function, the bound from Theorem 3.16 can be improved under the assumption (see also Section 3.1.1) that the target regression function $f_\rho \in \mathcal{F}_p$. More specifically, we derive the following result using mainly the auxiliary claims from Rahimi and Recht (2009).

Proposition 3.17. *Suppose that the target regression function $f_\rho \in \mathcal{F}_p$. Then, for all $\delta > 0$, the random kitchen sinks algorithm returns a function $f_{n,m}$ such that*

$$\mathbb{E}_{(x,y)\sim\rho}\left[\left(f_{n,m}(x) - y\right)^2\right] - \mathbb{E}_{(x,y)\sim\rho}\left[\left(f_\rho(x) - y\right)^2\right] \in \mathcal{O}\left(C^2\left(\frac{1}{\sqrt{n}} + \frac{1}{m}\right)\log\frac{1}{\delta}\right)$$

with probability at least $1-2\delta$ over the training dataset and the choice of the parameters $\{\theta_i\}_{i=1}^m$.

Proof. Suppose that the random kitchen sinks algorithm has drawn an i.i.d. sample of random features $\{\theta_i\}_{i=1}^m$ from the spectral distribution p. Then, this sample of Fourier features fixes the following hypothesis space

$$\hat{\mathcal{F}}_p = \left\{ f(x) = \sum_{i=1}^m \alpha_i \phi(x, \theta_i) \;\middle|\; |\alpha_i| \leq \frac{C}{m} \right\} .$$

Let $\hat{f} \in \hat{\mathcal{F}}_p$ and observe that

$$\begin{aligned}&\mathbb{E}_{(x,y)\sim\rho}\left[\left(\hat{f}(x) - y\right)^2\right] - \mathbb{E}_{(x,y)\sim\rho}\left[\left(f_\rho(x) - y\right)^2\right] = \\ &\int \hat{f}(x)^2 d\rho - 2\left\langle \hat{f}, f_\rho\right\rangle_{\mathcal{L}^2_\rho(X)} - \int f_\rho(x)^2 d\rho + 2\left\langle f_\rho, f_\rho\right\rangle_{\mathcal{L}^2_\rho(X)} = \left\|\hat{f} - f_\rho\right\|_\rho^2 .\end{aligned}$$

On the other hand, Rahimi and Recht (Lemma 1, 2009) show that there exists a hypothesis $\hat{f} \in \hat{\mathcal{F}}_p$ defined by m random features such that

$$\left\|\hat{f} - f_\rho\right\|_\rho \in \mathcal{O}\left(\frac{C}{\sqrt{m}}\sqrt{\log\frac{1}{\delta}}\right) .$$

If we now denote the minimizer of the expected squared error on $\hat{\mathcal{F}}_p$ with $\hat{f}_m$, we obtain (Lemma 3, Rahimi and Recht, 2009)

$$\begin{aligned}&\mathbb{E}_{(x,y)\sim\rho}\left[\left(f_{n,m}(x) - y\right)^2\right] - \mathbb{E}_{(x,y)\sim\rho}\left[\left(f_\rho(x) - y\right)^2\right] \leq \\ &\left|\mathbb{E}_{(x,y)\sim\rho}\left[\left(f_{n,m}(x) - y\right)^2\right] - \mathbb{E}_{(x,y)\sim\rho}\left[\left(\hat{f}_m(x) - y\right)^2\right]\right| + \left\|\hat{f}_m - f_\rho\right\|^2 \in \mathcal{O}\left(C^2\left(\frac{1}{\sqrt{n}} + \frac{1}{m}\right)\log\frac{1}{\delta}\right).\end{aligned}$$

□

Thus, if $f_\rho \in \mathcal{F}_p$ then the latter proposition implies that it is sufficient to take $m \in \mathcal{O}\left(\sqrt{n}\right)$ random Fourier features for the consistency of the corresponding empirical hypothesis $f_{n,m}$. This is a significant improvement over the bound from Theorem 3.16 that requires n random Fourier features for the consistency of hypotheses generated by random kitchen sinks. In addition to providing a generalization bound for learning with random kitchen sinks, Rahimi and Recht (2008b) have investigated the approximation properties of hypotheses from $\mathcal{F}_p$. The following theorem gives a concentration bound for functions returned by the random kitchen sinks algorithm (concentration around hypotheses from $\mathcal{F}_p$), as the number of sampled random (Fourier) features increases.

Theorem 3.18. *(Rahimi and Recht, 2008b) Let ρ be any probability measure with support on a compact set X and let $f \in \mathcal{F}_p$. Suppose $\phi \colon X \times \Theta \to \mathbb{R}$ satisfies $\sup_{x,\theta} \left|\phi(x,\theta)\right| \leq 1$. Then, for all $\delta > 0$, with probability at least $1-\delta$ over $\{\theta_i\}_{i=1}^m$ drawn iid from p, there exists $\{c_i\}_{i=1}^m$ such that the function $f_m(x) = \sum_{i=1}^m c_i \phi(x, w_i)$ satisfies*

$$\|f_m - f\|_\rho < \frac{\|f\|_\rho}{\sqrt{m}} \left(1 + \sqrt{2 \log \frac{1}{\delta}}\right) .$$

Thus, as we increase the number of random Fourier features our approximation of a hypothesis from $\mathcal{F}_p$ concentrates around it in the Hilbert space $\mathcal{L}_\rho^2(X)$. The following is an even stronger result with a concentration bound in the uniform norm.

Theorem 3.19. *(Rahimi and Recht, 2008b) Let $f \in \mathcal{F}_p$ and let $\phi \colon \mathbb{R} \to \mathbb{R}$ be an L-Lipschitz function such $\phi(x,\theta) = \phi(\theta^\top x)$, $\phi(0) = 0$, and $\left\|\phi\right\|_\infty < 1$. Suppose furthermore that p has a finite second moment. Then, for all $\delta > 0$, with probability at least $1-\delta$ over $\{\theta_i\}_{i=1}^m$ drawn iid from p there exist $\{c_i\}_{i=1}^m$ such that the function $f_m(x) = \sum_{i=1}^m c_i \phi\left(\theta_i^\top x\right)$ satisfies*

$$\|f_m - f\|_\infty < \frac{\|f\|_\rho}{\sqrt{m}} \left(\sqrt{\log \frac{1}{\delta}} + 4LB\sqrt{\mathbb{E}\left[\|\theta\|^2\right]}\right) .$$

While these two results show that the hypotheses returned by the random kitchen sinks algorithm concentrate, they do not provide an insight into the approximation properties of the hypothesis space $\mathcal{F}_p$. For that, Rahimi and Recht (2008b) have shown that $\mathcal{F}_p$ is dense in the reproducing Hilbert space of the stationary kernel corresponding to the spectral distribution p that defines the hypothesis space $\mathcal{F}_p$. More formally, we have the following theorem.

Theorem 3.20. *(Rahimi and Recht, 2008b) Let $\mathcal{H}_k$ be the reproducing kernel Hilbert space corresponding to a positive definite kernel*

$$k(x,x') = \int_\Theta \phi(x,\theta)\, \phi(x',\theta)\, p(\theta)\, d\theta .$$

Then, the hypothesis space $\mathcal{F}_p$ is dense in $\mathcal{H}_k$.

Hence, if the reproducing kernel Hilbert space is dense in the space of continuous functions defined on a compact set X, then the same holds for the hypothesis space $\mathcal{F}_p$. From previous work (Micchelli et al., 2006), we know that the reproducing kernel Hilbert space of the Gaussian kernel is dense in the space of continuous functions. Thus, the random kitchen sinks algorithm can, with sufficiently large number of random Fourier features (sampled from the Gaussian density function) and sufficiently large number of training examples, approximate arbitrarily well any continuous function defined on a compact set $X \subset \mathbb{R}^d$.

3.2.2 Learning with Greedily Constructed Fourier Features

The random kitchen sinks algorithm, reviewed in the previous section, is an efficient method for the approximation of hypotheses from a stationary reproducing kernel Hilbert space. However, for good generalization properties that approach requires an *a priori* specification of a suitable spectral measure which is often not feasible. To address this shortcoming of random kitchen sinks, we investigate here an instance of our approach in which Fourier features act as greedy features. We start by showing that, for this particular choice of greedy features, our approach is capable of approximating arbitrarily well any bounded function from any stationary reproducing kernel Hilbert space. Following this, we discuss the regularization term in the feature construction step at line 3 of Algorithm 3.1 and provide the hyperparameter gradient for this optimization problem. The hyperparameter gradient can then be used in combination with any off-the-shelf minimization algorithm such as the L-BFGS-B solver available in most numerical packages (e.g., SciPy, Matlab etc.).

Taking $\phi(\cdot) = \cos(\cdot)$ in the definition of $\mathcal{F}_\Theta$ we obtain a set of cosine-wave features

$$\mathcal{F}_{\cos} = \left\{ a\cos\left(\langle w, x\rangle + b\right) \mid w \in \mathbb{R}^d, a, b \in \mathbb{R}, |a| \leq r, \|w\|_2 \leq s, |b| \leq t \right\} .$$

For this set of features, the approach outlined in Section 3.1.1 can construct a rich set of hypotheses. To demonstrate this, we make a connection to stationary reproducing kernel Hilbert spaces and show that the approach can approximate any bounded function from any stationary reproducing kernel Hilbert space. This means that a set of linear hypotheses defined by cosine features can be of high capacity and our approach can overcome the problems with the low capacity of linear hypotheses defined on few input features.

Theorem 3.21. *Let $\mathcal{H}_k$ be a reproducing kernel Hilbert space corresponding to a continuous stationary/shift-invariant and positive definite kernel k defined on a compact set X. Let μ be the positive and bounded spectral measure whose Fourier transform is the kernel k. For any probability measure ρ defined on X, it is possible to approximate any bounded function $f \in \mathcal{H}_k$ using a convex combination of m ridge-wave functions from $\mathcal{F}_{\cos}$ such that the approximation error in $\|\cdot\|_\rho$ decays with rate $\mathcal{O}(1/\sqrt{m})$.*

Proof. Let $f \in \mathcal{H}_k$ be any bounded function. From the definition of $\mathcal{H}_k$ it follows that the set $\mathcal{H}_0 = \operatorname{span}\{k(x, \cdot) \mid x \in X\}$ is a dense subset of $\mathcal{H}_k$. In other words, for every $\varepsilon > 0$ there is a bounded function $g \in \mathcal{H}_0$ such that $\|f - g\|_{\mathcal{H}_k} < \varepsilon$. As feature functions $k(x, \cdot)$ are continuous and defined on the compact set X, they are also bounded. Thus, we can assume that there exists a constant $B > 0$ such that $\sup_{x,y \in X} \left|k(x, y)\right| < B$. From here it follows

$$\|f - g\|_\infty = \sup_{x \in X} \left|\langle f - g, k(x, \cdot)\rangle_{\mathcal{H}_k}\right| \leq \sqrt{B}\|f - g\|_{\mathcal{H}_k} .$$

This means that convergence in $\|\cdot\|_{\mathcal{H}_k}$ implies the uniform convergence. The uniform convergence, on the other hand, implies the convergence in $\mathcal{L}^2_\rho(X)$ norm, i.e., for any probability measure ρ on the set X, for any $\varepsilon > 0$, and for any $f \in \mathcal{H}_k$ there exists $g \in \mathcal{H}_0$ such that

$$\|f - g\|_\rho < \varepsilon. \tag{3.7}$$

The function g is by definition a finite linear combination of feature functions $k(x_i, \cdot)$ (see, e.g., Chapter 1 in Bertinet and Agnan, 2004) and by Theorem 2.5 it can be written as

$$\begin{aligned} g(x) &= \sum_{i=1}^{l} \alpha_i k(x_i, x) = 2\int \left(\sum_{i=1}^{l} \alpha_i \cos\left(w^\top x_i + b\right) \right) \cos\left(w^\top x + b\right) d\hat{\mu}(w,b) \\ &= 2\mu(0) \int u(w,b) \cos\left(w^\top x + b\right) d\tilde{\mu}(w,b) \ , \end{aligned}$$

where $\tilde{\mu}$ is a probability measure on $\mathbb{R}^d \times [-\pi, \pi]$, $u(w,b) = \sum_{i=1}^{l} \alpha_i \cos(w^\top x_i + b)$, and $\int d\hat{\mu}(w,b) = \mu(0) < \infty$. From the boundedness of g, it follows that the function u is bounded for all w and b from the support of $\tilde{\mu}$, i.e., $|u(w,b)| \leq \sum_{i=1}^{l} |\alpha_i| < \infty$. Denoting with $\gamma(w,b) = 2\mu(0)\, u(w,b)$, we see that it is sufficient to prove that

$$\mathbb{E}_{\tilde{\mu}(w,b)}\left[\gamma(w,b)\cos\left(w^\top x + b\right)\right] \in \overline{\mathrm{co}\left(\mathcal{F}_{\cos}\right)} \ ,$$

where the closure is taken with respect to the norm in $\mathcal{L}^2_\rho(X)$. In particular, for a sample $(\mathbf{w}, \mathbf{b}) = \{(w_i, b_i)\}_{i=1}^{s}$ drawn independently from $\tilde{\mu}$ we have

$$\mathbb{E}_{(\mathbf{w},\mathbf{b})}\left[\int \left(g(x) - \frac{1}{s}\sum_{i=1}^{s} \gamma(w_i, b_i)\cos\left(w_i^\top x + b_i\right) \right)^2 d\rho \right] =$$

$$\frac{1}{s^2}\int \mathbb{E}_{(\mathbf{w},\mathbf{b})}\left[\left(\sum_{i=1}^{s} \underbrace{g(x) - \gamma(w_i, b_i)\cos\left(w_i^\top x + b_i\right)}_{\xi(x;\ w_i, b_i)} \right)^2\right] d\rho =$$

$$\frac{1}{s^2}\int \mathbb{E}_{(\mathbf{w},\mathbf{b})}\left[\left(\sum_{i=1}^{s} \xi(x;\ w_i, b_i) \right)^2\right] d\rho = \frac{1}{s}\int \mathbb{E}_{\tilde{\mu}}\left[\xi(x;\ w,b)^2\right] d\rho =$$

$$\frac{1}{s}\int \mathrm{Var}_{\tilde{\mu}}\left[g(x) - \gamma(w,b)\cos\left(w^\top x + b\right)\right] d\rho = \frac{1}{s}\int \mathrm{Var}_{\tilde{\mu}}\left[\gamma(w,b)\cos\left(w^\top x + b\right)\right] d\rho \ .$$

Note that the third equation follows from the fact that $\xi(x;\ w_i, b_i)$ are independent and identically distributed random variables and $\mathbb{E}\left[\xi(x;\ w_i, b_i)\,\xi\left(x;\ w_j, b_j\right)\right] = 0$. As established earlier, coefficients $\gamma(w,b)$ are bounded and, therefore, random variable $\eta_x(w,b) = \gamma(w,b)\cos(w^\top x + b)$ is bounded, as well. Hence, from $\sup_{w,b}\left|\eta_x(w,b)\right| = D < \infty$ it follows that $\mathrm{Var}_{\tilde{\mu}}(\eta_x(w,b)) \leq D^2$ and consequently

$$g_s(x; (\mathbf{w}, \mathbf{b})) = \frac{1}{s}\sum_{i=1}^{s} \gamma(w_i, b_i)\cos\left(w_i^\top x + b_i\right) \quad \Longrightarrow \quad \mathbb{E}_{g_s}\left[\|g - g_s\|_\rho^2\right] \leq \frac{D^2}{s} \ .$$

As the expected value of the norm $\|g - g_s\|_\rho$ is bounded by a constant, it follows that there exists a function g_s which can be represented as a convex combination of s ridge-wave functions from $\mathcal{F}_{\cos}$ and for which it holds $\|g - g_s\|_\rho \in \mathcal{O}\left(\frac{1}{\sqrt{s}}\right)$. Moreover, there exists a sequence of functions $\{g_m\}_{m \geq 1}$ converging to g in $\|\cdot\|_\rho$ such that each g_m is a convex combination of m elements from $\mathcal{F}_{\cos}$ and $\|g - g_m\|_\rho \in \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$.

Hence, we have proved that $g \in \overline{\mathrm{co}\left(\mathcal{F}_{\cos}\right)}$, where the closure is taken with respect to $\|\cdot\|_\rho$. It is then possible to approximate any bounded function $f \in \mathcal{H}_k$ using a convex combination of m ridge-wave functions from $\mathcal{F}_{\cos}$ with the rate $\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$, i.e., for all $m \in \mathbb{N}$

$$\|f - g_m\|_\rho \leq \|f - g\|_\rho + \|g - g_m\|_\rho \in \mathcal{O}\left(\frac{1}{\sqrt{m}}\right) .$$

□

Having established that the proposed approach can construct a rich set of hypotheses, we discuss the regularization term in the feature construction step of Algorithm 3.1. It is frequently the case that generalization properties and capacity of a hypothesis space are controlled by penalizing the objective function with the squared l_2 norm of parameter vectors defining the features. For instance, this is the case for the majority of standard activation functions in neural networks literature (e.g., see Anthony and Bartlett, 2009). A reason behind this choice of regularization lies in the fact that those activation functions are monotone and the variation of any such basis function corresponds with the variation in its ridge argument. Assuming that the data is centered, the variation of the ridge argument is

$$\int w^\top x x^\top w \, d\rho = w^\top \underbrace{\int x x^\top \, d\rho}_{\Sigma} \; w = \|w\|_\Sigma^2 .$$

If now the instances are also normalized such that the features have the unit variance over the dataset, the diagonal of Σ is then the vector of all ones. Thus, for learning with monotone activation/ridge functions regularization via the squared l_2 norm of the parameter vectors defining features can be interpreted as the penalization of an upper bound on a regularization term that would impose a low variation constraint on the features.

In contrast to monotone basis functions, for cosine ridge-wave bases it is not straightforward to relate the variation of the basis function to its argument (considered over a given finite sample of the data). Namely, cosine is a periodic function and while spectral parameters with large norms can cause significant variation in the ridge argument, this does not necessarily imply a large variation of the basis function over a finite sample. It is also possible for a parameter vector with the smaller norm to cause more variation in the basis function over a finite sample than the one with the larger norm. We, therefore, opt to regularize the spectrum of the cosine ridge function by penalizing the objective with its squared $\mathcal{L}_\rho^2(X)$ norm. Before we give the regularization term, we first note that the bias term from the cosine-wave features can be eliminated using the trigonometric additive formulas and then the cosine-wave basis function takes the from

$$\phi_{w,a}(x) = a_1 \sin\left(w^\top x\right) + a_2 \cos\left(w^\top x\right) . \tag{3.8}$$

Now, taking the squared $\mathcal{L}_\rho^2(X)$ norm of this function we get

$$\begin{aligned} \left\|\phi_{w,a}\right\|_\rho^2 &= a_1^2 \int \sin^2\left(w^\top x\right) d\rho + a_2^2 \int \cos^2\left(w^\top x\right) d\rho + a_1 a_2 \int \sin\left(2 w^\top x\right) d\rho \\ &= \frac{a_1^2 + a_2^2}{2} + \frac{a_2^2 - a_1^2}{2} \int \cos(2r)\, d\mu_w + a_1 a_2 \int \sin(2r)\, d\mu_w , \end{aligned}$$

where $\mu_w(r) = \rho(\{x \mid w^\top x = r\})$. If we assume that the probability measure ρ is symmetric, then we have that $\mu_w(r) = \mu_w(-r)$ and using the fact that $\sin(2r)$ is an odd function, we obtain $\int \sin(2w^\top x)\, d\rho = 0$. In the absence of the marginal distribution ρ, the integral $\int \cos(2r)\, d\mu_w$ can be estimated from the training sample with $\frac{1}{n}\sum_{i=1}^{n} \cos(2w^\top x_i)$, where $x_i \overset{\text{i.i.d.}}{\sim} \rho(x)$. Moreover, if under these assumptions $a_1 = a_2$ (i.e., the bias term in the cosine-wave is given by $b = \frac{\pi}{4} + l\pi$, $l \in \mathbb{N}^+$) then the dependence on the spectrum is lost and the variation of $\phi_{w,a}$ can be controlled with $\|a\|_2^2$. Alternatively, if the probability measure ρ is not symmetric and $a_1 = a_2$, then we can attempt to control the variation in $\phi_{w,a}$ by penalizing the objective with $\|a\|_2^2 (1 + \|w\|_\Sigma^2)$. The latter is motivated by the fact that for $r > 0$ the term $\sin(2r)$ can be upper bounded with $2r$.

Having discussed means to control the variation in the ridge-wave basis function, we now formulate the optimization problem (line 3, Algorithm 3.1) for the setting with cosine-wave features and provide the gradients for the hyperparameters. The optimization problem, for this particular choice of features, can be specified as

$$\min_{w,\lambda,c} \; \frac{1}{n}\sum_{i=1}^{n}\Big(c_0 f_{0,i} + c_1 \sin\big(w^\top x_i\big) + c_2 \cos\big(w^\top x_i\big) - y_i\Big)^2 +$$
$$\lambda\left(\frac{c_0^2}{n}\sum_{i=1}^{n} f_{0,i}^2 + \frac{c_1^2 + c_2^2}{2} + \frac{c_2^2 - c_1^2}{2n}\sum_{i=1}^{n}\cos\big(2w^\top x_i\big) + \frac{2c_0c_1}{n}\sum_{i=1}^{n}\sin\big(w^\top x_i\big) f_{0,i} + \right.$$
$$\left. \frac{2c_0c_2}{n}\sum_{i=1}^{n}\cos\big(w^\top x_i\big) f_{0,i} + \frac{c_1c_2}{n}\sum_{i=1}^{n}\sin\big(2w^\top x_i\big)\right),$$

where w and λ are optimized as hyperparameters and the amplitude vector c as a regressor. This optimization problem is convex in c because the regularization term can be expressed using a quadratic term defined with a positive definite matrix (see the matrix D given below). For a fixed choice of the hyperparameters w and λ, an optimal amplitude vector c can be computed in a closed form. As such an amplitude vector is completely determined by the choice of the hyperparameters, it is sufficient to optimize this problem only by w and λ. The hyperparameter optimization is, in general, non-convex and typically results in a local optimum. In Section 3.1.3 we have, however, demonstrated that for the convergence of the greedy procedure an optimal solution is not required in each step of the constructive process.

In the feature construction step of Algorithm 3.1, we want to choose the hyperparameters via k-fold cross-validation and in order to achieve this we follow the procedure proposed by Keerthi et al. (2007). Let us denote the above described 3-dimensional feature representation of the data with $Z_w \in \mathbb{R}^{n\times 3}$ and set $\sigma_0(w) = \frac{1}{n}\sum_{i=1}^{n} f_{0,i}^2$, $\sigma_1(w) = \frac{1}{n}\sum_{i=1}^{n}\sin(w^\top x_i) f_{0,i}$, $\sigma_2(w) = \frac{1}{n}\sum_{i=1}^{n}\cos(w^\top x_i) f_{0,i}$, $\sigma_3(w) = \frac{1}{n}\sum_{i=1}^{n}\sin(2w^\top x_i)$, and $\sigma_4(w) = \frac{1}{n}\sum_{i=1}^{n}\cos(2w^\top x_i)$. Now, in the place of the identity matrix in the standard derivation of the ridge regression objective function we have the following symmetric and positive definite matrix

$$D = \begin{bmatrix} \sigma_0 & \sigma_1 & \sigma_2 \\ \sigma_1 & 0.5(1-\sigma_4) & 0.5\sigma_3 \\ \sigma_2 & 0.5\sigma_3 & 0.5(1+\sigma_4) \end{bmatrix} = \frac{1}{n}\sum_{i=1}^{n} z_w(x_i)\, z_w(x_i)^\top \;,$$

where $z_w(x_i) = \operatorname{vec}(f_{0,i}, \sin(w^\top x_i), \cos(w^\top x_i)) \in \mathbb{R}^3$.

At this point our derivation follows closely the derivation by Keerthi et al. (2007). Taking the derivatives with respect to c and setting the gradient of the loss to zero we get

$$Z_w^\top Z_w c - Z_w^\top y + n\lambda D c = 0 \quad \Longrightarrow \quad \Big(Z_w^\top Z_w + n\lambda D\Big)c = Z_w^\top y \;.$$

Let us denote with $P = Z_w^\top Z_w + n\lambda D$, $q = Z_w^\top y$, and $\theta = (w, \lambda)$. We note here that P and q are defined over the *training instances* x and their labels y. We now take the implicit derivative of this equation to obtain the derivative of the regressor c with respect to the hyperparameters,

$$\frac{\partial c}{\partial \theta} = P^{-1}\left(\frac{\partial q}{\partial \theta} - \frac{\partial P}{\partial \theta} c\right) .$$

As already stated, the choice of λ directly determines the coefficients c and to obtain these we need to perform the hyperparameter selection which is done over the validation samples. In other words,

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\ \frac{1}{k}\sum_{i=1}^{k} \frac{1}{|F_i|} \sum_{(x,y)\in F_i} \left(c^\top z_w(x) - y\right)^2 ,$$

where F_i denotes one of the k validation folds in k-fold cross-validation. Let us now consider only the sample from one validation fold and denote it with F. At the same time let F^c denotes its complement or the training sample when F is used as the validation fold. Here, we note that $(x, y) \in F$ are *different* from samples participating in the definitions of P and q when taking F as the validation fold. Taking derivatives with respect to θ we obtain the hyperparameter gradient

$$\frac{2}{|F|} \sum_{(x,y)\in F} \left(c^\top z_w(x) - y\right)\left(\frac{\partial z_w(x)}{\partial \theta} c + z_w(x) P^{-1}\left(\frac{\partial q}{\partial \theta} - \frac{\partial P}{\partial \theta} c\right)\right) .$$

Let us introduce the vector $t = \operatorname{vec}(t_0, t_1, t_2) \in \mathbb{R}^3$ as a solution to the following linear system

$$Pt = \frac{1}{|F|} \sum_{(x,y)\in F} \left(c^\top z_w(x) - y\right) z_w(x) .$$

We then write the derivative of each term in the hyperparameter gradient separately as

$$\begin{aligned}
&\frac{\partial}{\partial w}\left(c^\top z_w(x)\right) = \left(c_1 \cos\left(w^\top x\right) - c_2 \sin\left(w^\top x\right)\right) x ,\\
&\frac{\partial}{\partial w}\left(t^\top q\right) = \sum_{(x,y)\in F^c} \left(t_1 \cos\left(w^\top x\right) - t_2 \sin\left(w^\top x\right)\right) xy ,\\
&\frac{\partial}{\partial w}\left(t^\top P c\right) = \\
&(1+\lambda)(t_0 c_1 + t_1 c_0) \sum_{i=1}^{|F^c|} f_{0,i} \cos\left(w^\top x_i\right) x_i - (1+\lambda)(t_0 c_2 + t_2 c_0) \sum_{i=1}^{|F^c|} f_{0,i} \sin\left(w^\top x_i\right) x_i + \\
&(1+\lambda)(t_1 c_2 + t_2 c_1) \sum_{x\in F^c} \cos\left(2w^\top x\right) x + (1+\lambda)(t_1 c_1 - t_2 c_2) \sum_{x\in F^c} \sin\left(2w^\top x\right) x ,\\
&\frac{\partial}{\partial \lambda} = n\, t^\top D c .
\end{aligned}$$

Performing the gradient descent using these hyperparameter gradients we obtain both the spectrum w and the amplitudes c. The spectrum regularization term which is defined using the empirical estimates of the sine and cosine integrals affects the gradient with respect to w via the λ factor in the third expression. In our experiments (Section 3.3), we have observed that the capacity parameter λ usually takes the value below 10^{-4}. Thus, the influence of the spectrum regularization term is less significant than the amplitude regularization term. For this reason, in our implementation we only penalize the empirical squared error objective with the squared norm of the amplitude vector, i.e., $\Omega(c, w) = \|c\|_2^2$. We leave it for future work to empirically evaluate the behavior of the regularization operator $\Omega(c, w) = \|c\|_2^2 (1 + \|w\|_\Sigma^2)$.

3.2.3 À la Carte

À la carte is a related, recent approach (Yang et al., 2015) for learning with Fourier features that estimates a suitable spectral distribution of features with a mixture of Gaussians and learns a linear regression model in that feature space. As the mixture of Gaussians is dense in the space of probability distributions (Silverman, 1986), this approach can also approximate any bounded hypotheses from any reproducing kernel Hilbert space. The quality of such approximations crucially depends on the number of components in the Gaussian mixture and the number of samples allocated to individual components. These parameters are typically unknown prior to model fitting and are often estimated via cross-validation. Thus, an efficient Fourier features parametrization is specified beforehand and all the parameters are optimized jointly together with the regression coefficients, rather than sequentially in a greedy manner. More formally, Yang et al. (2015) represent their regression estimator as

$$f(x) = \sum_{i=1}^{m} \alpha_i \sin\left(w_i^\top x\right) + \alpha_i^{'} \cos\left(w_i^\top x\right) ,$$

where m denotes the number of spectral features, and

$$w \sim \sum_{j=1}^{Q} \frac{\gamma_j}{\sqrt{(2\pi)^d \left|\Sigma_j\right|}} \exp\left(-\frac{\left(x - \mu_j\right)^\top \Sigma_j^{-1} \left(x - \mu_j\right)}{2}\right) ,$$

with Σ_j diagonal, $\gamma_j \geq 0$, and $\sum_{j=1}^{Q} \gamma_j = 1$. The proposed algorithm then finds a feature representation together with a linear model by optimizing the marginal likelihood of the corresponding Gaussian process. As we have chosen to compare all feature construction approaches using the standard linear regression, we present an equivalent implementation of this approach based on the hyperparameter optimization method by Keerthi et al. (2007). More specifically, we solve the following optimization problem

$$\begin{aligned} \min \quad & \frac{1}{n} \sum_{i=1}^{n} \left[\sum_{q=1}^{Q} \nu_q^2 \sum_{j=1}^{s} \alpha_{qj} \sin\left(u_{qj}^\top \Sigma_q^{1/2} x_i + \mu_q^\top x_i\right) + \beta_{qj} \cos\left(u_{qj}^\top \Sigma_q^{1/2} x_i + \mu_q^\top x_i\right) - y_i \right]^2 + \\ & \lambda \left(\|\alpha\|^2 + \left\|\beta\right\|^2 \right) , \end{aligned}$$

where α and β are optimized as regressors and μ_q, Σ_q (diagonal covariance matrix), ν_q, and λ as hyperparameters. The u-vectors are random vectors sampled from the multivariate standard normal distribution. These vectors act as a regularization term on the spectrum of the cosine features forcing the frequencies to stay in the pre-specified number of clusters/components. The optimization problem with respect to regression coefficients α and β is convex and solvable in a closed form. In particular, let us denote $\Sigma_q^{1/2}$ with D_q, parameterized features with $Z_\theta \in \mathbb{R}^{n \times 2Qs}$, hyperparameters with $\theta = (\mu, D, \nu) \in \mathbb{R}^{Q(2d+1)}$, and regressors with $c = (\alpha, \beta) \in \mathbb{R}^{2Qs}$. Then, if we set $P = Z_\theta^\top Z_\theta + n\lambda\mathbb{I}$ and $q = Z_\theta^\top y$, the latter optimization problem becomes

$$\min \; c^\top P c - 2c^\top q \; .$$

While this problem is convex in $c = (\alpha, \beta)$, the optimization over the regressors only does not define an à la carte hypothesis. In particular, that approach finds a hypothesis by optimizing over the regression coefficients c and the hyperparameters θ. As noted earlier, the hyperparameter optimization is, in general, a non-convex problem and typically results in a local optimum. Yang et al. (2015) have, however, for this particular approach provided a heuristic for initializing the hyperparameters which often results in a good approximation of an optimal solution (e.g., see the empirical results in Section 3.3).

From this point onwards, we follow the derivation from the previous section and denote with $t \in \mathbb{R}^{2Qs}$ the solution of the following linear system of equations

$$Pt = \frac{1}{|F|} \sum_{(x,y)\in F} \left(c^\top z_\theta(x) - y\right) z_\theta(x) \; .$$

Let us also denote with $\oplus$ and $\otimes$ the element-wise addition and multiplication operators for vectors/matrices, with $U \in \mathbb{R}^{Qd\times d}$ the block matrix comprised of vertically stacked standard normal matrices $U_q \in \mathbb{R}^{d\times d}$, with $D \in \mathbb{R}^{Qd\times d}$ the block matrix comprised of vertically stacked diagonal matrices $D_q \in \mathbb{R}^{d\times d}$, $\mu \in \mathbb{R}^{d\times Q}$ the matrix with stacked mean parameter vectors $\mu_q \in \mathbb{R}^d$, $\tau_q = U_q D_q x \oplus \mu_q^\top x$, and $\tau = UDx \oplus \mu^\top x$. Then, following the outlined principles for implicit derivation (Section 3.2.2), we obtain the hyperparameter gradients with respect to the validation objective:

$$\frac{\partial}{\partial \mu_q}\left(c^\top z_\theta(x)\right) = \nu_q^2 \Big(\alpha_q^\top \cos\left(\tau_q\right) - \beta_q^\top \sin\left(\tau_q\right)\Big) x \; ,$$

$$\frac{\partial}{\partial D_q}\left(c^\top z_\theta(x)\right) = \nu_q^2 \Big(\left(\alpha_q \odot U_q\right)^\top \cos\left(\tau_q\right) - \left(\beta_q \odot U_q\right)^\top \sin\left(\tau_q\right)\Big) \odot x \; ,$$

$$\frac{\partial}{\partial \nu_q}\left(c^\top z_\theta(x)\right) = 2\nu_q \Big(\alpha_q^\top \sin\left(\tau_q\right) + \beta_q^\top \cos\left(\tau_q\right)\Big) \; ,$$

$$\frac{\partial}{\partial \mu_q}\left(t^\top q\right) = \sum_{(x,y)\in F^c} y\nu_q^2 \Big(t_{q\alpha}^\top \cos\left(\tau_q\right) - t_{q\beta}^\top \sin\left(\tau_q\right)\Big) x \; ,$$

$$\frac{\partial}{\partial D_q}\left(t^\top q\right) = \sum_{(x,y)\in F^c} y\nu_q^2 \Big(\left(t_{q\alpha} \odot U_q\right)^\top \cos\left(\tau_q\right) - \left(t_{q\beta} \odot U_q\right)^\top \sin\left(\tau_q\right)\Big) \odot x \; ,$$

$$\frac{\partial}{\partial \nu_q}\left(t^\top q\right) = 2\nu_q \sum_{(x,y)\in F^c} t_{q\alpha}^\top \sin\left(\tau_q\right) + t_{q\beta}^\top \cos\left(\tau_q\right) \; ,$$

$$\frac{\partial}{\partial \mu_q}\left(t^\top P c\right) = \sum_{(x,y)\in F^c} \nu_q^4 \left\{ \left[t_\alpha^\top \sin(\tau) + t_\beta^\top \cos(\tau) \right] \cdot \left[\alpha_q^\top \cos\left(\tau_q\right) - \beta_q^\top \sin\left(\tau_q\right) \right] + \left[\alpha^\top \sin(\tau) + \beta^\top \cos(\tau) \right] \cdot \left[t_{q\alpha}^\top \cos\left(\tau_q\right) - t_{q\beta}^\top \sin\left(\tau_q\right) \right] \right\} x \, ,$$

$$\frac{\partial}{\partial \nu_q}\left(t^\top P c\right) = 2 \sum_{(x,y)\in F^c} \nu_q^3 \left\{ \left[t_\alpha^\top \sin(\tau) + t_\beta^\top \cos(\tau) \right] \cdot \left[\alpha^\top \sin(\tau) + \beta^\top \cos(\tau) \right] \right\} ,$$

$$\frac{\partial}{\partial D_q}\left(t^\top P c\right) = \sum_{(x,y)\in F^c} \nu_q^4 \left\{ \left[t_\alpha^\top \sin(\tau) + t_\beta^\top \cos(\tau) \right] \cdot \left[\left(\alpha_q \odot U_q\right)^\top \cos\left(\tau_q\right) - \left(\beta_q \odot U_q\right)^\top \sin\left(\tau_q\right) \right] + \left[\alpha^\top \sin(\tau) + \beta^\top \cos(\tau) \right] \cdot \left[\left(t_{q\alpha} \odot U_q\right)^\top \cos\left(\tau_q\right) - \left(t_{q\beta} \odot U_q\right)^\top \sin\left(\tau_q\right) \right] \right\} \odot x \, ,$$

$$\frac{\partial}{\partial \lambda} = n \, t^\top c \, .$$

The cost of computing the gradient at a hyperparameter vector involves solving a $(2Qs)$-dimensional linear system. Moreover, this system needs to be solved for each validation fold in a k-fold splitting, required for the hyperparameter optimization over validation samples. In contrast to this, our greedy feature contruction approach has the linear worst case runtime complexity in the number of instances and the number of Fourier features (see Section 3.4).

Related and quite similar to à la carte is an approach for learning sparse Gaussian processes (Lázaro-Gredilla et al., 2010). The approach specifies only the number of Fourier features in the representation and works by optimizing jointly over the feature parameters and regression coefficients. A derivation similar to the one provided here for à la carte shows that the hyperparameter gradient requires solving a $(2m)$-dimensional linear system in problems with m Fourier features, which can be computationally inefficient (e.g., see Section 3.3). Moreover, the approach does not rely on an efficient parametrization of the spectral distribution and the hyperparameter gradients can also be significantly more expensive to compute compared to à la carte. This can be seen by comparing the total number of hyperparameters for the two approaches, i.e., $2md \gg Q(2d+1)$.

3.3 Experiments

In this section, we assess the performance of our approach (see Algorithm 3.2) by comparing it to other feature construction approaches on synthetic and real-world datasets. We evaluate the effectiveness of the approach with a variant of Fourier features as ridge bases. For this particular set of features, our approach is directly comparable to random Fourier features (Section 3.2.1) and à la carte (Section 3.2.3). The implementation details of the three approaches are provided in Section 3.3.1 and the results of the experiments are discussed in Section 3.3.2.

Before we proceed with a detailed description of the baselines, we briefly describe the datasets and the experimental setting. The experiments were conducted on three groups of datasets. The first group contains four UCI datasets on which we performed parameter tuning of all three algorithms (Table 3.1, datasets 1-4). The second group contains the datasets with more than 5000 instances available from Torgo (2016). The idea is to use this group of datasets to test the generalization properties of the considered algorithms (Table 3.1, datasets 5-10). The third group contains two artificial and very noisy datasets that are frequently used in regression tree benchmark tests. For each considered dataset, we split the data into 10 folds; we refer to these splits as the outer cross-validation folds. In each step of the outer cross-validation, we use nine folds as the training sample and one fold as the test sample. For the purpose of the hyperparameter tuning we split the training sample into five folds; we refer to these splits as the inner cross-validation folds. We run all algorithms on identical outer cross-validation folds and construct feature representations with 100 and 500 features. The performance of the algorithms is assessed by comparing the root mean squared error of linear ridge regression models trained in the constructed feature spaces and the average time needed for the outer cross-validation of one fold.

3.3.1 Baselines

Having described the experimental setting, we now provide implementation details for all the considered algorithms: greedy feature construction, à la carte method (Yang et al., 2015), and random kitchen sinks (Rahimi and Recht, 2009).

We have implemented a distributed version of Algorithm 3.2 using a python package *mpi4py*. For the experiments with 100 spectral features the algorithm is simulated using 5 cores on a single physical machine – each core corresponds to one instance of greedy functional descent. The remaining parameters are: the number of data passes $T = 1$, the maximum number of greedy descent steps $p = 20$, precision parameter $\varepsilon = 0.01$ that stops the greedy descent when the successive improvement in the accuracy is less than 1%, and feature cut-off η that is set to 0.0001% of the range of the output variable. For the experiments with 500 spectral features the algorithm is simulated using 5 physical machines. To communicate features more efficiently 5 cores on each of the physical machines are used giving the total number of 25 cores corresponding to 25 instances of greedy functional descent. The remaining parameters for this setting are identical to the ones used in the experiments with 100 features. As the greedy functional descent is stopped when the successive improvement in the accuracy is below 1%, the approach terminates sooner than the alternative approaches (w.r.t. the number of constructed features) for simple hypotheses (see Section 3.3.2). Having described the parameter configuration for Algorithm 3.2, we now address the choice of the regularization term. In particular, to control the smoothness of newly constructed features, we penalize the objective in line 3 so that the solutions with the small $\mathcal{L}^2_\rho(X)$ norm are preferred. For this choice of regularization term and cosine-wave features, we empirically observe that the optimization objective is almost exclusively penalized by the l_2 norm of the coefficient vector c. Following this observation, we have simulated the greedy descent with $\Omega(c, w) = \|c\|_2^2$. In contrast to á la carte (see below), we *did not engineer a heuristic for the initial solution* of the hyperparameter optimization problem. Instead, we have initialized the spectral features by sampling from the standard normal distribution and dividing the entries of the sampled vector with the square root of its dimension.

To be as objective as possible to the best performing competing method, we have parallelized the implementation of this algorithm and simulated it by following the ARD-heuristic for choosing the initial solution (e.g., see the supplementary materials in Yang et al., 2015).

This, in particular, refers to the initial choice of diagonal covariance matrices that define components in a mixture of Gaussians. We also follow the instructions from the supplementary material of Yang et al. (2015) and initialize the means to vectors that are close to zero. The ν parameters are initialized by setting their values to the standard deviation of outputs divided by the number of components Q. We have optimized the hyperparameters with the L-BFGS-B solver from *SciPy*. As reported in Yang et al. (2015), we simulate the algorithm with 10 random restarts such that for each initial solution the algorithm makes 20 iterations of L-BFGS-B minimization and then continues with the best hyperparameter vector for another 200 iterations. In all the experimental settings (with 100 and 500 features), we have run this algorithm using $Q = 1$, $Q = 2$, and $Q = 5$ mixture components. As this can be computationally intensive on a single core, we have parallelized our implementation of à la carte by computing the parts of hyperparameter gradient that correspond to different validation folds on different cores. For the inner cross-validation performed with 5-fold splitting this has resulted in a speed up of approximately 4-5 times compared to a single core implementation. In Section 3.3.2, we report the walltimes of the parallelized implementation of à la carte for all the experimental settings (with 100 and 500 features).

As already outlined in Section 3.2.1, any stationary positive definite kernel can be represented as a Fourier transform of a positive measure. Thus, in order to generate a kernel feature map it is sufficient to sample spectral frequencies from this measure. Genton (2002) and Rahimi and Recht (2008a) have provided the parameterized spectral density functions corresponding to Gaussian, Laplace, and Cauchy kernels. We use these parameterizations to generate spectral features and then train a linear ridge regression model in the constructed feature space. To choose the most suitable parameterization, we cross-validate 10 parameters from the log-space of $[-3, 2]$.

Table 3.1: To facilitate the comparison between datasets we have normalized the outputs so that their range is one. The accuracy of the algorithms is measured using the root mean squared error, multiplied by 100 to mimic percentage error (w.r.t. the range of the outputs). The mean and standard deviation of the error are computed after performing 10-fold cross-validation. The reported walltime is the average time it takes a method to cross-validate one fold. To assess whether a method performs statistically significantly better than the other on a particular dataset we perform the paired Welch t-test (Welch, 1947) with $p = 0.05$. The significantly better results for the considered settings are marked in bold.

Dataset	n	d	$m = 100$				$m = 500$			
			GFC		ALC		GFC		ALC	
			Error	Walltime	Error	Walltime	Error	Walltime	Error	Walltime
parkinsons tm (total)	5875	21	2.73 (±0.19)	00 : 03 : 49	**0.78** (±0.13)	00 : 05 : 19	2.20 (±0.27)	00 : 04 : 15	**0.31** (±0.17)	00 : 27 : 15
ujindoorloc (latitude)	21048	527	**3.17** (±0.15)	00 : 21 : 39	6.19 (±0.76)	01 : 21 : 58	**3.04** (±0.19)	00 : 36 : 49	6.99 (±0.97)	02 : 23 : 15
ct-slice	53500	380	**2.93** (±0.10)	00 : 52 : 05	3.82 (±0.64)	03 : 31 : 25	**2.59** (±0.10)	01 : 24 : 41	2.73 (±0.29)	06 : 11 : 12
Year Prediction MSD	515345	90	10.06 (±0.09)	01 : 20 : 12	**9.94** (±0.08)	05 : 29 : 14	10.01 (±0.08)	01 : 30 : 28	**9.92** (±0.07)	11 : 58 : 41
delta-ailerons	7129	5	3.82 (±0.24)	00 : 01 : 23	3.73 (±0.20)	00 : 05 : 13	3.79 (±0.25)	00 : 01 : 57	3.73 (±0.24)	00 : 25 : 14
kinematics	8192	8	5.18 (±0.09)	00 : 04 : 02	5.03 (±0.23)	00 : 11 : 28	4.65 (±0.11)	00 : 04 : 44	5.01 (±0.76)	00 : 38 : 53
cpu-activity	8192	21	2.65 (±0.12)	00 : 04 : 23	2.68 (±0.27)	00 : 09 : 24	2.60 (±0.16)	00 : 04 : 24	2.62 (±0.15)	00 : 25 : 13
bank	8192	32	9.83 (±0.27)	00 : 01 : 39	9.84 (±0.30)	00 : 12 : 48	9.83 (±0.30)	00 : 02 : 01	9.87 (±0.42)	00 : 49 : 48
pumadyn	8192	32	3.44 (±0.10)	00 : 02 : 24	**3.24** (±0.07)	00 : 13 : 17	**3.30** (±0.06)	00 : 02 : 27	3.42 (±0.15)	00 : 57 : 33
delta-elevators	9517	6	5.26 (±0.17)	00 : 00 : 57	5.28 (±0.18)	00 : 07 : 07	5.24 (±0.17)	00 : 01 : 04	5.23 (±0.18)	00 : 32 : 30
ailerons	13750	40	4.67 (±0.18)	00 : 02 : 56	4.89 (±0.43)	00 : 16 : 34	4.51 (±0.12)	00 : 02 : 11	4.77 (±0.40)	01 : 05 : 07
pole-telecom	15000	26	7.34 (±0.29)	00 : 10 : 45	7.16 (±0.55)	00 : 20 : 34	5.55 (±0.15)	00 : 11 : 37	5.20 (±0.51)	01 : 39 : 22
elevators	16599	18	3.34 (±0.08)	00 : 03 : 16	3.37 (±0.55)	00 : 21 : 20	3.12 (±0.20)	00 : 04 : 06	3.13 (±0.24)	01 : 20 : 58
cal-housing	20640	8	**11.55** (±0.24)	00 : 05 : 49	12.69 (±0.47)	00 : 11 : 14	**11.17** (±0.25)	00 : 06 : 16	12.70 (±1.01)	01 : 01 : 37
breiman	40768	10	**4.01** (±0.03)	00 : 02 : 46	4.06 (±0.04)	00 : 13 : 52	4.01 (±0.03)	00 : 03 : 04	4.03 (±0.03)	01 : 04 : 16
friedman	40768	10	3.29 (±0.09)	00 : 06 : 07	3.37 (±0.46)	00 : 18 : 43	**3.16** (±0.03)	00 : 07 : 04	3.25 (±0.09)	01 : 39 : 37

3.3.2 Results

As the best performing configuration of à la carte on the development datasets is the one with $Q = 5$ components, we report in Table 3.1 the error and walltime for this configuration.

Table 3.2: This table presents the results of experiments with the à la carte method using 100 Fourier features. The mean and standard deviation of the root mean squared error are computed after performing 10-fold cross-validation. The fold splitting is performed such that all algorithms train and predict over identical samples. The reported walltime is the average time it takes a method to cross-validate one fold.

Dataset	n	d	$m = 100$					
			$Q = 1, s = 100$		$Q = 2, s = 50$		$Q = 5, s = 20$	
			Error	Walltime	Error	Walltime	Error	Walltime
parkinsons tm (total)	5875	21	0.81 (±0.67)	00 : 07 : 58	0.73 (±0.33)	00 : 08 : 29	0.78 (±0.13)	00 : 05 : 19
ujindoorloc (latitude)	21048	527	6.21 (±0.41)	00 : 27 : 41	6.94 (±0.66)	00 : 45 : 55	6.19 (±0.76)	01 : 21 : 58
ct-slice	53500	380	4.11 (±0.25)	00 : 46 : 56	3.86 (±0.28)	01 : 18 : 00	3.82 (±0.64)	03 : 31 : 25
Year Prediction MSD	515345	90	10.10 (±0.07)	02 : 49 : 21	10.03 (±0.08)	02 : 32 : 09	9.94 (±0.08)	05 : 29 : 14
delta-ailerons	7129	5	3.83 (±0.18)	00 : 04 : 19	3.84 (±0.27)	00 : 05 : 27	3.73 (±0.20)	00 : 05 : 13
kinematics	8192	8	6.21 (±0.54)	00 : 10 : 21	5.31 (±0.34)	00 : 09 : 30	5.03 (±0.23)	00 : 11 : 28
cpu-activity	8192	21	2.59 (±0.17)	00 : 08 : 22	2.77 (±0.33)	00 : 06 : 19	2.68 (±0.27)	00 : 09 : 24
bank	8192	32	9.72 (±0.32)	00 : 12 : 03	9.79 (±0.29)	00 : 10 : 27	9.84 (±0.30)	00 : 12 : 48
pumadyn	8192	32	3.17 (±0.07)	00 : 10 : 34	3.18 (±0.06)	00 : 11 : 01	3.24 (±0.07)	00 : 13 : 17
delta-elevators	9517	6	5.28 (±0.17)	00 : 03 : 31	5.27 (±0.17)	00 : 06 : 52	5.28 (±0.18)	00 : 07 : 07
ailerons	13750	40	4.62 (±0.34)	00 : 08 : 42	4.57 (±0.12)	00 : 09 : 54	4.89 (±0.43)	00 : 16 : 34
pole-telecom	15000	26	8.73 (±0.52)	00 : 12 : 39	7.34 (±0.32)	00 : 15 : 00	7.16 (±0.55)	00 : 20 : 34
elevators	16599	18	3.46 (±0.23)	00 : 07 : 51	3.70 (±0.55)	00 : 07 : 41	3.37 (±0.55)	00 : 21 : 20
cal-housing	20640	8	13.61 (±0.35)	00 : 09 : 49	13.07 (±1.53)	00 : 12 : 17	12.69 (±0.47)	00 : 11 : 14
breiman	40768	10	4.01 (±0.03)	00 : 12 : 34	4.02 (±0.04)	00 : 09 : 13	4.06 (±0.04)	00 : 13 : 52
friedman	40768	10	3.16 (±0.03)	00 : 18 : 58	3.16 (±0.03)	00 : 19 : 46	3.37 (±0.46)	00 : 18 : 43

From the walltime numbers we see that our approach is in both considered settings – with 100 and 500 features – always faster than à la carte. Moreover, the proposed approach is able to generate a feature representation with 500 features in less time than required by à la carte for a representation of 100 features. In order to compare the performance of the two methods with respect to accuracy, we use the Wilcoxon signed rank test (Wilcoxon, 1945; Demšar, 2006). As our approach with 500 features is on all datasets faster than à la carte with 100 features, we first compare the errors obtained in these experiments. For 95% confidence, the threshold value of the Wilcoxon signed rank test with 16 datasets is $T = 30$ and from our results we get the T-value of 28. As the T-value is below the threshold, our algorithm can with 95% confidence generate in less time a statistically significantly better feature representation than à la carte. For the errors obtained in the settings where both methods have the same number of features, we obtain the T-values of 60 and 42. While in the first case for the setting with 100 features the test is inconclusive, in the second case our approach is with 80% confidence statistically significantly more accurate than à la carte. To evaluate the performance of the approaches on individual datasets, we perform the paired Welch (1947) t-test with $p = 0.05$. Again, the results indicate a good/competitive performance of our algorithm compared to this baseline. An extensive summary containing the results of experiments with different configurations of à la carte, is provided in Tables 3.2 and 3.3.

In addition to à la carte, we have also evaluated the approach against random kitchen sinks (Rahimi and Recht, 2008b) with random Fourier features corresponding to Gaussian, Laplace, and Cauchy kernels. Table 3.4 provides an extensive summary of the results of these experiments. From this table, we can observe that both, greedy feature construction and à la carte, can always construct a better feature representation than random kitchen sinks. However, while these two approaches are significantly more accurate than random kitchen sinks, the latter approach is computationally more effective.

3.4 Discussion

In this section, we discuss the advantages of the proposed method over the state-of-the-art baselines in learning fast stationary kernels and other related approaches.

Table 3.3: This table presents the results of experiments with the à la carte method using 500 Fourier features. The mean and standard deviation of the root mean squared error are computed after performing 10-fold cross-validation. The fold splitting is performed such that all algorithms train and predict over identical samples. The reported walltime is the average time it takes a method to cross-validate one fold.

Dataset	n	d	$m = 500$					
			$Q = 1, s = 500$		$Q = 2, s = 250$		$Q = 5, s = 100$	
			Error	Walltime	Error	Walltime	Error	Walltime
parkinsons tm (total)	5875	21	0.29 (±0.33)	00 : 30 : 00	0.34 (±0.17)	00 : 37 : 05	0.31 (±0.17)	00 : 27 : 15
ujindoorloc (latitude)	21048	527	8.08 (±1.67)	01 : 34 : 01	7.83 (±1.05)	02 : 02 : 19	6.99 (±0.97)	02 : 23 : 15
ct-slice	53500	380	2.98 (±0.07)	02 : 43 : 28	2.97 (±0.19)	04 : 09 : 43	2.73 (±0.29)	06 : 11 : 12
Year Prediction MSD	515345	90	10.00 (±0.07)	07 : 51 : 20	9.94 (±0.07)	08 : 55 : 38	9.92 (±0.07)	11 : 58 : 41
delta-ailerons	7129	5	3.82 (±0.18)	00 : 14 : 37	3.85 (±0.37)	00 : 18 : 23	3.73 (±0.24)	00 : 25 : 14
kinematics	8192	8	5.34 (±0.48)	00 : 29 : 45	4.82 (±0.32)	00 : 41 : 13	5.01 (±0.76)	00 : 38 : 53
cpu-activity	8192	21	2.47 (±0.36)	00 : 52 : 16	2.52 (±0.20)	00 : 29 : 34	2.62 (±0.15)	00 : 25 : 13
bank	8192	32	9.62 (±0.29)	00 : 51 : 08	9.97 (±0.37)	00 : 48 : 22	9.87 (±0.42)	00 : 49 : 48
pumadyn	8192	32	3.12 (±0.07)	00 : 44 : 17	3.17 (±0.05)	00 : 44 : 28	3.42 (±0.15)	00 : 57 : 33
delta-elevators	9517	6	5.27 (±0.18)	00 : 15 : 59	5.28 (±0.18)	00 : 22 : 44	5.23 (±0.18)	00 : 32 : 30
ailerons	13750	40	4.50 (±0.10)	00 : 41 : 45	4.49 (±0.17)	00 : 36 : 54	4.77 (±0.40)	01 : 05 : 07
pole-telecom	15000	26	6.30 (±0.45)	01 : 08 : 32	5.35 (±0.27)	01 : 17 : 48	5.20 (±0.51)	01 : 39 : 22
elevators	16599	18	3.28 (±0.27)	01 : 01 : 44	3.37 (±0.12)	00 : 36 : 30	3.13 (±0.24)	01 : 20 : 58
cal-housing	20640	8	12.27 (±1.51)	01 : 03 : 49	12.15 (±0.43)	00 : 55 : 06	12.70 (±1.01)	01 : 01 : 37
breiman	40768	10	4.01 (±0.04)	00 : 39 : 36	4.02 (±0.04)	00 : 35 : 45	4.03 (±0.03)	01 : 04 : 16
friedman	40768	10	3.16 (±0.04)	00 : 55 : 19	3.24 (±0.06)	00 : 56 : 33	3.25 (±0.09)	01 : 39 : 37

Table 3.4: This table presents the results of experiments with the random kitchen sinks approach using 100 and 500 Fourier features. The mean and standard deviation of the root mean squared error are computed after performing 10-fold cross-validation. The fold splitting is performed such that all algorithms train and predict over identical samples. The reported walltime is the average time it takes a method to cross-validate one fold.

Dataset	n	d	$m = 100$				$m = 500$			
			Gauss	Cauchy	Laplace	Walltime	Gauss	Cauchy	Laplace	Walltime
parkinsons tm	5875	21	5.81 (±0.32)	5.79 (±0.41)	6.22 (±1.31)	00 : 04 : 38	4.75 (±0.70)	4.63 (±0.37)	4.34 (±0.22)	00 : 11 : 10
ujindoorloc	21048	527	12.55 (±0.60)	12.36 (±0.67)	10.23 (±0.88)	00 : 05 : 02	7.40 (±0.25)	7.19 (±0.31)	5.53 (±0.48)	00 : 24 : 41
ct-slice	53500	380	11.45 (±0.30)	11.44 (±0.31)	11.69 (±0.50)	00 : 04 : 32	7.85 (±0.17)	7.77 (±0.09)	7.90 (±0.13)	00 : 49 : 10
Year Prediction	515345	90	10.75 (±0.04)	10.79 (±0.34)	11.07 (±0.13)	00 : 10 : 26	10.53 (±0.04)	10.51 (±0.03)	10.46 (±0.06)	03 : 10 : 41
delta-ailerons	7129	5	3.84 (±0.14)	3.84 (±0.14)	3.86 (±0.14)	00 : 04 : 02	3.82 (±0.13)	3.84 (±0.15)	3.81 (±0.15)	00 : 15 : 35
kinematics	8192	8	11.09 (±0.26)	11.01 (±0.25)	11.47 (±0.39)	00 : 03 : 27	7.33 (±0.53)	7.37 (±0.43)	8.17 (±0.31)	00 : 11 : 53
cpu-activity	8192	21	6.72 (±0.62)	5.94 (±0.59)	3.90 (±0.66)	00 : 04 : 38	3.10 (±0.17)	3.05 (±0.17)	2.75 (±0.25)	00 : 14 : 31
bank	8192	32	10.15 (±0.46)	10.13 (±0.42)	10.10 (±0.46)	00 : 04 : 38	9.91 (±0.44)	9.97 (±0.49)	9.92 (±0.45)	00 : 15 : 53
pumadyn	8192	32	15.19 (±0.29)	15.18 (±0.29)	15.20 (±0.29)	00 : 04 : 50	15.20 (±0.28)	15.18 (±0.31)	15.25 (±0.27)	00 : 16 : 52
delta-elevators	9517	6	5.30 (±0.14)	5.30 (±0.13)	5.28 (±0.13)	00 : 05 : 16	5.29 (±0.15)	5.27 (±0.14)	5.27 (±0.14)	00 : 14 : 49
ailerons	13750	40	4.77 (±0.16)	4.77 (±0.21)	4.89 (±0.07)	00 : 03 : 58	4.53 (±0.11)	4.52 (±0.10)	4.58 (±0.12)	00 : 17 : 40
pole-telecom	15000	26	24.26 (±0.75)	22.62 (±0.68)	25.07 (±1.37)	00 : 04 : 44	18.08 (±0.56)	17.53 (±0.46)	15.63 (±0.63)	00 : 19 : 08
elevators	16599	18	4.11 (±0.23)	3.88 (±0.21)	4.09 (±0.54)	00 : 04 : 43	3.44 (±0.19)	3.56 (±0.37)	3.39 (±0.15)	00 : 19 : 24
cal-housing	20640	8	12.99 (±0.36)	12.66 (±0.35)	12.83 (±0.53)	00 : 05 : 27	11.78 (±0.38)	11.80 (±0.43)	11.51 (±0.37)	00 : 19 : 02
breiman	40768	10	4.01 (±0.03)	4.01 (±0.03)	4.02 (±0.03)	00 : 04 : 26	4.01 (±0.03)	4.01 (±0.03)	4.01 (±0.03)	00 : 24 : 45
friedman	40768	10	5.15 (±0.10)	5.25 (±0.16)	5.06 (±0.32)	00 : 03 : 58	3.30 (±0.03)	3.29 (±0.03)	3.26 (±0.04)	00 : 21 : 17

Flexibility

The presented approach is a highly flexible supervised feature construction method. In contrast to random kitchen sinks (Rahimi and Recht, 2008a,b), the proposed method does not require a spectral measure to be specified a priori. In the experiments, we have demonstrated that the choice of spectral measure is important as, for the considered measures (corresponding to Gaussian, Laplace, and Cauchy kernels), the random kitchen sinks approach is outperformed on all datasets. The second baseline, à la carte, is more flexible when it comes to the choice of spectral measure and works by approximating it with a mixture of Gaussians. However, the number of components and features per component needs to be specified beforehand or cross-validated. In contrast, our approach mimics functional gradient descent and can be simulated without specifying the size of the feature representation beforehand. Instead, a stopping criteria (see, e.g., Algorithm 3.1) based on the successive decay of the error can be devised. As a result, the proposed approach terminates sooner than the alternative approaches for simple concepts/hypotheses (i.e., outputs a sparse solution). The proposed method is also easy to implement (the hyperparameter gradients are provided in Section 3.2.2) and allows us to extend the existing feature representation without complete re-training of the model. We note that the approaches based on random Fourier features are also simple to implement and can be re-trained efficiently with the increase in the number of features (Dai et al., 2014). À la carte, on the other hand, is less flexible in this regard – due to the number of hyperparameters and the complexity of gradients it is not straightforward to implement it.

Scalability

The fact that our greedy descent can construct a feature in time linear in the number of instances n and dimension of the problem d makes the proposed approach highly scalable. In particular, the complexity of the proposed parallelization scheme is dominated by the cost of fitting a linear model and the whole algorithm runs in time $\mathcal{O}\left(T(m^3 + m^2 n + nmd)\right)$, where T denotes the number of data passes (i.e., linear model fits) and m number of constructed features. To scale this scheme to problems with millions of instances, it is possible to fit linear models using the parallelized stochastic gradient descent (Zinkevich et al., 2010). For linear models fitted with a variant of this optimization algorithm, our approach has better than linear worst case runtime complexity $\mathcal{O}(nmd/\kappa)$, where κ denotes the number of available processing cores. As for the choice of T, the standard setting in simulations of stochastic gradient descent is 5-10 data passes. Thus, the presented approach is quite robust and can be applied to large scale datasets. In contrast to this, the cost of performing a gradient step in the hyperparameter optimization of à la carte is $\mathcal{O}\left(m^3 + m^2 n + nmd\right)$. In our empirical evaluation using an implementation with 10 random restarts, the approach needed at least 20 steps per restart to learn an accurate model. The required number of gradient steps and the cost of computing them hinder the application of à la carte to large scale datasets. In random kitchen sinks which also run in time $\mathcal{O}\left(m^3 + m^2 n + nmd\right)$, the main cost is the fitting of linear models – one for each pair of considered spectral and regularization parameters.

Related Approaches

Beside fast kernel learning approaches, the presented method is also related to neural networks parameterized with a single hidden layer. These approaches can be seen as feature construction methods jointly optimizing over the whole feature representation. A detailed study of the approximation properties of a hypothesis space of a single layer network with the sigmoid ridge function has been provided by Barron (1993). In contrast to these approaches, we construct features incrementally by fitting residuals and we do this with a set of non-monotone ridge functions as a dictionary of features. Regarding our generalization bound, we note that the past work on single layer neural networks contains similar results but in the context of monotone ridge functions (Anthony and Bartlett, 2009).

As the goal of our approach is to construct a feature space for which linear hypotheses will be of sufficient capacity, the presented method is also related to linear models working with low-rank kernel representations. For instance, Fine and Scheinberg (2002) investigate a training algorithm for SVMs using low-rank kernel representations. The difference between our approach and this method is in the fact that the low-rank decomposition is performed without considering the labels. Side knowledge and labels are considered by Kulis et al. (2006) and Bach and Jordan (2005) in their approaches to construct a low-rank kernel matrix. However, these approaches are not selecting features from a set of ridge functions, but find a subspace of a preselected kernel feature space with a good set of hypothesis.

From the perspective of the optimization problem considered in the greedy descent (Algorithm 3.1) our approach can be related to single index models (SIM) where the goal is to learn a regression function that can be represented as a single monotone ridge function (Kalai and Sastry, 2009; Kakade et al., 2011). In contrast to this, our approach learns target/regression functions from the closure of the convex hull of ridge functions. Typically, these target functions cannot be written as single ridge functions. Moreover, our ridge functions are not necessarily monotone and are more general than the ones considered in SIM models.

In addition to these approaches and considered baseline methods, the presented feature construction approach is also related to methods optimizing expected loss functions using functional gradient descent (Mason et al., 2000). However, while Mason et al. (2000) focus on classification problems and hypothesis spaces with finite VC dimension, we focus on the estimation of regression functions in spaces with infinite VC dimension (e.g., see Sections 3.1 and 3.2.2). In contrast to that work, we also provide a convergence rate for our approach. Similarly, Friedman (2000) has proposed a gradient boosting machine for greedy function estimation. In their approach, the empirical functional gradient is approximated by a weak learner which is then combined with previously constructed learners following a *stagewise* strategy. This is different from the *stepwise* strategy that is followed in our approach where previously constructed estimators are readjusted when new features are added. The approach in Friedman (2000) is investigated mainly in the context of regression trees, but it can be adapted to feature construction. To the best of our knowledge, theoretical and empirical properties of this approach in the context of feature construction and stationary reproducing kernel Hilbert spaces have not been considered so far.

Part II

Randomized Greedy Approaches

CHAPTER 4

Nyström Method with Kernel K-means++ Landmarks

Kernel methods are a powerful class of machine learning algorithms that can be used for solving classification and regression problems, clustering, anomaly detection, and dimensionality reduction (Schölkopf and Smola, 2002). For this class of methods, the learning problem can often be posed as a convex optimization problem for which the representer theorem (Wahba, 1990) guarantees that an optimal solution can be found in the subspace of the kernel feature space spanned by the instances. Typically, the algorithms from this class of methods first transform the data to a symmetric and positive definite matrix and then use an off-the-shelf matrix-based algorithm for solving the resulting convex optimization problem (Bach and Jordan, 2005). Computational and space complexities of these approaches are at least quadratic in the number of instances and in several algorithms, requiring a matrix inversion or eigendecomposition, the computational complexity is cubic. To overcome this computational shortcoming and scale kernel methods to large scale datasets, Williams and Seeger (2001) have proposed to use a variant of the Nyström method (Nyström, 1930) for low-rank approximation of kernel matrices. The approach is motivated by the fact that frequently used kernels have a fast decaying spectrum and that small eigenvalues can be removed without a significant effect on the precision (Schölkopf and Smola, 2002). For a learning problem with n instances and a given subset of l landmarks, the Nyström method finds a low-rank approximation in time $\mathcal{O}(l^2 n + l^3)$ and kernel methods with the low-rank approximation in place of the kernel matrix scale as $\mathcal{O}(l^3)$. In practice, $l \ll n$ and the approach can scale kernel methods to millions of instances.

The crucial step in the Nyström approximation of a symmetric and positive definite matrix is the choice of landmarks and an optimal choice is a difficult discrete/combinatorial problem directly influencing the quality of the approximation. A large part of the existing work has, therefore, focused on providing approximation guarantees for different landmark selection strategies. Following this line of research, we investigate the effectiveness of kernel K-means++ samples (Arthur and Vassilvitskii, 2007) as landmarks in the Nyström method for low-rank approximation of kernel matrices. Previous empirical studies (Zhang et al., 2008; Kumar et al., 2012) observe that the landmarks obtained using K-means clustering define a good low-rank approximation of kernel matrices. However, the existing work does not provide a theoretical guarantee on the approximation error for this approach to landmark selection. We close this gap and provide the first bound on the relative approximation error in the Frobenius norm for this landmark selection strategy. An important part of our theoretical contribution is the first complete proof of a claim by Ding and He (2004) on the relation between the subspace spanned by optimal K-means centroids and left singular vectors of the feature space. While our results (Propositions 4.3 and 4.5) cover the general case, that of Ding and He (2004) is restricted to data matrices with piecewise constant right singular vectors.

In Section 4.1, we provide a brief review of the Nyström method for low-rank approximation of kernel matrices with two different perspectives on the approach. Following this, we review K-means clustering and express the corresponding optimization problem as a constrained low-rank approximation problem (Section 4.2.1). In Section 4.2.2, we build on the constrained low-rank formulation of K-means clustering and give the first complete proof of a claim by Ding and He (2004) on the relation between the subspace spanned by optimal K-means centroids and left singular vectors of the feature space. Having established that the claim by Ding and He (2004) does not hold for general data matrices, we then review the K-means++ sampling scheme for cluster seeding (Section 4.3.1) and give a pseudo-code description of our landmark selection strategy (Section 4.3.2), which is a kernelized variant of the K-means++ algorithm. In Section 4.3.3, we analyze the theoretical properties of the kernel K-means++ landmark selection strategy and give the first bound on the relative approximation error in the Frobenius norm for this strategy. Having given a bound on the approximation error for the proposed landmark selection strategy, we provide a brief overview of the existing landmark selection algorithms and discuss our work in relation to approaches directly comparable to ours (Section 4.4). For the frequently used Gaussian kernel, we also theoretically motivate the instance space Lloyd refinements (Lloyd, 1982) of kernel K-means++ landmarks. The results of our empirical study are presented in Section 4.5 and indicate a superior performance of the proposed approach over competing methods. This is also in agreement with the previous studies on K-means centroids as landmarks by Zhang et al. (2008) and Kumar et al. (2012).

4.1 Nyström Method

In this section, we review the Nyström method for low-rank approximation of kernel matrices. The method was originally proposed for the approximation of integral eigenfunctions (Nyström, 1930) and later adapted to low-rank approximation of kernel matrices by Williams and Seeger (2001). In the literature, however, the latter adaptation of the original approach is known as the Nyström method. We review the original approach in Section 4.1.1 and its adaptation to low-rank approximation of kernel matrices in Section 4.1.2. Following this, we provide an alternative derivation of the approach relating it to subspace approximations (Smola and Schölkopf, 2000; Schölkopf and Smola, 2002). The section concludes with a characterization of an optimal low-rank approximation of a kernel matrix (Section 4.1.4).

4.1.1 Nyström Method for Approximation of Integral Eigenfunctions

Let $\mathcal{X}$ be an instance space, $X = \{x_1, x_2, \cdots, x_n\}$ an independent sample from a Borel probability measure ρ defined on $\mathcal{X}$, and $\mathcal{H}$ the reproducing kernel Hilbert space with a Mercer kernel $h \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Such kernels can be decomposed in terms of the eigenfunctions and eigenvalues of the corresponding integral operator. More precisely, we have that the following theorem holds (note that in the original formulation, ρ is a bounded Borel measure).

Theorem 4.1. *(Mercer, 1909; Schölkopf and Smola, 2002) Suppose $h \in L_\infty(\mathcal{X} \times \mathcal{X})$ is a symmetric real-valued function such that the integral operator $T_h \colon L_2(\mathcal{X}) \to L_2(\mathcal{X})$,*

$$(T_h f)(x) := \int_{\mathcal{X}} h(x, x') f(x') d\rho(x') \ ,$$

is positive definite; that is, for all $f \in L_2(\mathcal{X})$, we have

$$\int_{\mathcal{X} \times \mathcal{X}} h(x, x') f(x) f(x') d\rho(x) d\rho(x') \geq 0 \ .$$

Let $\psi_i \in L_2(\mathcal{X})$ be the normalized orthogonal eigenfunctions of T_h associated with eigenvalues $\lambda_i > 0$, sorted in non-increasing order. Then, $\{\lambda_i\}_{i \geq 1} \in \ell_1$ and

$$h(x, x') = \sum_{i=1}^{N_{\mathcal{H}}} \lambda_i \psi_i(x) \psi_i(x')$$

for almost all $(x, x') \in \mathcal{X} \times \mathcal{X}$. Either $N_{\mathcal{H}} \in \mathbb{N}$ or $N_{\mathcal{H}} = \infty$; in the latter case, the series converges absolutely and uniformly for almost all $(x, x') \in \mathcal{X} \times \mathcal{X}$.

As the eigenfunctions are normalized and mutually orthogonal, it follows that

$$(T_h \psi_i)(x) = \int_{\mathcal{X}} h(x, x') \psi_i(x') d\rho(x') = \sum_{j=1}^{N_{\mathcal{H}}} \lambda_j \psi_j(x) \int_{\mathcal{X}} \psi_j(x') \psi_i(x') d\rho(x') = \lambda_i \psi_i(x) \ .$$

The original Nyström method (Nyström, 1930) considers the problem of approximating eigenfunctions ψ_i given an independent sample X from the probability measure ρ. In order to approximate an eigenfunction ψ_i using the sample X, the eigenfunction is matched with the Monte–Carlo estimate of the corresponding integral operator, i.e.,

$$\frac{1}{n} \sum_{l=1}^{n} h(x, x_l) \psi_i(x_l) \approx \lambda_i \psi_i(x) \ . \tag{4.1}$$

On the other hand, the empirical version of the orthogonality constraint implies that

$$\frac{1}{n} \sum_{l=1}^{n} \psi_i(x_l) \psi_j(x_l) \approx 0 \ .$$

Now, plugging x_j in place of x in Eq. (4.1) we get

$$\frac{1}{n} \sum_{l=1}^{n} h\left(x_j, x_l\right) \psi_i(x_l) \approx \lambda_i \psi_i\left(x_j\right) \ .$$

As the latter equation holds for all $1 \leq i, j \leq n$, then we can write it in a matrix form as

$$H^{(n)} U^{(n)} = \Lambda^{(n)} U^{(n)} \ , \tag{4.2}$$

where $H^{(n)}$ denotes the kernel matrix with $h_{ij}^{(n)} = h\left(x_i, x_j\right)$, $U^{(n)}$ the column orthonormal matrix with $u_{ij}^{(n)} \approx \frac{1}{\sqrt{n}} \psi_j(x_i)$, and $\Lambda^{(n)}$ the diagonal matrix with $\lambda_i^{(n)} \approx n\lambda_i$ $(1 \leq i, j \leq n)$. Hence, combining the eigendecomposition of the kernel matrix $H^{(n)}$ with Eq. (4.1) we obtain an empirical estimate of the eigenfunction ψ_i,

$$\psi_i(x) \approx \frac{1}{n\lambda_i} \sum_{l=1}^{n} h(x, x_l) \psi_i(x_l) \approx \frac{\sqrt{n}}{\lambda_i^{(n)}} \sum_{l=1}^{n} h(x, x_l) u_{li}^{(n)} = \frac{\sqrt{n}}{\lambda_i^{(n)}} h_x^\top U_i^{(n)},$$

where $h_x = \operatorname{vec}(h(x, x_1), \ldots, h(x, x_n))$ and $U_i^{(n)}$ is the i-th column in the matrix $U^{(n)}$.

4.1.2 Application of the Nyström Method to Low-Rank Approximation of Kernel Matrices

The Nyström method approximates integral eigenfunctions and eigenvalues using an independent and identically distributed sample of instances. Thus, the approach can also be applied to a uniformly selected subsample of instances and the resulting approximation of integral eigenfunctions and eigenvalues should be approximately equal to that obtained using the full sample. In other words, the eigendecomposition of the kernel matrix defined by the subsample can be used to make a low-rank approximation of the full kernel matrix.

To see this, let us assume that $Z = \{z_1, \dots, z_m\}$ are landmarks sampled uniformly from X. Then, from Section 4.1.1 we have that

$$\lambda_i^{(n)} := n\lambda_i \approx \frac{n}{m}\lambda_i^{(m)} \quad \wedge \quad u_{ji}^{(n)} := \frac{1}{\sqrt{n}}\psi_i\left(x_j\right) \approx \sqrt{\frac{m}{n}}\frac{1}{\lambda_i^{(m)}} h_{x_j \times Z}^\top U_i^{(m)} ,$$

where $H_Z = U^{(m)}\Lambda^{(m)}U^{(m)\top}$ is an eigendecomposition of the kernel matrix defined by a subsample Z, $h_{x \times Z} = \operatorname{vec}(h(x,z_1),\dots,h(x,z_m))$, $1 \le i \le m$, and $1 \le j \le n$. Thus, the i-th column in the column orthonormal matrix $U^{(n)}$ is given by

$$U_i^{(n)} = \sqrt{\frac{m}{n}}\frac{1}{\lambda_i^{(m)}} H_{X\times Z} U_i^{(m)} ,$$

where $H_{X\times Z}$ is the block in the kernel matrix H corresponding to kernel values between the instances from X and Z, respectively. From here it then follows that $H^{(n)} = U^{(n)}\Lambda^{(n)}U^{(n)\top}$ is an approximate eigendecomposition of the kernel matrix H. More precisely, we have that

$$\begin{aligned} H^{(n)} &= \sum_{i=1}^{m} \lambda_i^{(n)} U_i^{(n)} U_i^{(n)\top} = H_{X\times Z}\left(\sum_{i=1}^{m}\frac{1}{\lambda_i^{(m)}} U_i^{(m)} U_i^{(m)\top}\right) H_{Z\times X} \\ &= H_{X\times Z} H_{Z\times Z}^{-1} H_{Z\times X} . \end{aligned} \tag{4.3}$$

The matrix $H^{(n)}$ is a rank m approximation of the kernel matrix H and in the relevant literature it is known as the Nyström approximation of the kernel matrix.

4.1.3 Alternative Derivation of the Nyström Method for Low-Rank Approximation of Kernel Matrices

Let us now describe an alternative method for derivation of the Nyström low-rank approximation of a kernel matrix. For that, assume we are given a set of landmark instances $Z = \{z_1, \cdots z_m\}$ (not necessarily a subset of the sample) and that the goal is to approximate the evaluation functionals $h(x_i,\cdot)$ for all $1 \le i \le n$ using linear combinations of evaluation functionals defined by the landmarks. This goal can be formally stated as

$$\min_{\alpha\in\mathbb{R}^{m\times n}} \sum_{i=1}^{n} \left\| h(x_i,\cdot) - \sum_{j=1}^{m} \alpha_{ji} h\left(z_j,\cdot\right) \right\|_{\mathcal{H}}^2 . \tag{4.4}$$

After expanding the norm, the problem is transformed into

$$\min_{\alpha\in\mathbb{R}^{m\times n}} \sum_{i=1}^{n} h_{ii} - 2h_{x_i}^\top \alpha_i + \alpha_i^\top H_{Z\times Z}\alpha_i , \tag{4.5}$$

where α_i denotes the i-th column of α. Each summand in the optimization objective is a convex function depending only on one column of α. Hence, the optimal solution is

$$\alpha^* = H_{Z\times Z}^{-1} H_{Z\times X} \,.$$

From here it then follows that given landmarks Z, the optimal low-rank approximation $\tilde{H}$ of kernel matrix H can be written as

$$\tilde{H} = H_{X\times Z} H_{Z\times Z}^{-1} H_{Z\times X} \,.$$

The latter is precisely the Nyström low-rank approximation of kernel matrix H, as defined in Eq. (4.3). Thus, the approach computes the optimal embedding of instances to a subspace of the kernel feature space spanned by the maps of the landmarks from the instance space.

4.1.4 Optimal Low-Rank Approximation of Kernel Matrix

While the problem of computing the optimal embedding of instances to a subspace spanned by the kernel functions corresponding to landmarks is convex and solvable in closed form (Section 4.1.3), the problem of choosing the best set of landmarks is a combinatorial problem that is difficult to solve. To evaluate the effectiveness of the subspace spanned by a given set of landmarks it is standard to use the Schatten matrix norms (Weidmann, 1980). The *Schatten p-norm* of a symmetric and positive definite matrix H is defined as

$$\|H\|_p = \left(\sum_{i=1}^{n} \lambda_i^p\right)^{\frac{1}{p}} \,,$$

where $\lambda_i \geq 0$ are eigenvalues of H and $p \geq 1$. For $p = \infty$ the Schatten p-norm is equal to the operator norm and for $p = 2$ it is equal to the Frobenius norm. The three most frequently used Schatten norms are $p = 1, 2, \infty$ and for these norms the following inequalities hold

$$\|H\|_\infty = \max_i \lambda_i \leq \sqrt{\sum_i \lambda_i^2} = \sqrt{\operatorname{tr}(H^\top H)} = \|H\|_2 \leq \sum_i \lambda_i = \operatorname{tr}(H) = \|H\|_1 \,.$$

From Eq. (4.4) and (4.5) it follows that for a subspace of the kernel feature space spanned by kernel functions corresponding to a given set of landmarks Z, the Schatten 1-norm approximation error of the optimal embedding into this subspace is given by

$$L(\alpha^*) = \operatorname{tr}(H) - \operatorname{tr}(\tilde{H}) = \left\|H - \tilde{H}\right\|_1 \,.$$

The latter equation follows from the properties of trace and the fact that $\Xi = H - \tilde{H}$ is a symmetric and positive definite matrix with $\xi_{ij} = \left\langle \xi(x_i, \cdot), \xi\left(x_j, \cdot\right)\right\rangle_{\mathcal{H}}$ and $\xi(x_i, \cdot) = h(x_i, \cdot) - \sum_{k=1}^{m} \alpha_{ki}^* h(z_k, \cdot)$. Thus, for a good Nyström approximation of a kernel matrix it is crucial to select the landmarks to reduce the error in one of the frequently used Schatten p-norms, i.e.,

$$Z^* = \operatorname*{argmin}_{Z\subset X \,\wedge\, |Z|=m} \left\|H - H_{X\times Z} H_{Z\times Z}^{-1} H_{Z\times X}\right\|_p \,.$$

Having characterized an optimal set of landmarks, let us now characterize the optimal low-rank approximation of the kernel matrix H. The following proposition is a special case of the Eckart–Young–Mirsky theorem (Eckart and Young, 1936; Mirsky, 1960) and it characterizes the optimal low-rank approximation of a symmetric and positive definite matrix.

Proposition 4.2. *(Eckart and Young, 1936; Mirsky, 1960) Suppose U_m and Λ_m are the top m eigenvectors and eigenvalues from an eigendecomposition of the kernel matrix H. Then, at the low-rank approximation $\tilde{H}^* = U_m \Lambda_m U_m^\top$, the Schatten p-norm error of a rank m approximation of the matrix H attains its minimal value.*

Proof. While the Eckart–Young–Mirsky theorem holds for all Schatten p-norms (i.e., $1 \leq p \leq \infty$), the focus of the proof will be on the cases with $p = 1$, $p = 2$, and $p = \infty$. If we let $H_m = AB^\top$ with $A, B \in \mathbb{R}^{n \times m}$ be a rank m approximation of H, then we need to show that

$$\|H - H_m\|_p \geq \left\|H - \tilde{H}^*\right\|_p .$$

Let us begin by showing the claim for $p = \infty$. For that, let $H = U\Lambda U^\top$ be an eigendecomposition of the matrix H with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n \geq 0$ and corresponding eigenvectors $\{u_i\}_{i=1}^n$. As the low-rank approximation $H_m = AB^\top$ is of rank m, then there exists a vector $g \in \operatorname{span}\{u_1, \ldots, u_{m+1}\}$ such that $B^\top g = 0$. To see this, first note that the null space of the matrix B is of dimension $\dim(\mathcal{N}(B)) = n - m$. From here it then follows that

$$\dim(\mathcal{N}(B)) + \dim(\operatorname{span}\{u_1, \ldots, u_{m+1}\}) = n + 1 ,$$

and there exists $g \in \mathcal{N}(B) \cap \operatorname{span}\{u_1, \ldots, u_{m+1}\}$ such that $\|g\| = 1$. Hence,

$$\begin{aligned} \|H - H_m\|_\infty \geq \quad & \|(H - H_m)g\|_2 = \|Hg\|_2 = \left\|\sum_{i=1}^n \lambda_i u_i u_i^\top g\right\|_2 = \left\|\sum_{i=1}^{m+1} \lambda_i u_i u_i^\top g\right\|_2 = \\ & \left\|\sum_{i=1}^{m+1} \lambda_i g_i u_i\right\|_2 = \|U\Lambda_{m+1} g\|_2 \geq \lambda_{m+1} = \left\|H - \tilde{H}^*\right\|_\infty , \end{aligned}$$

where the first inequality follows from the operator norm and the latter one can be obtained using the Rayleigh–Ritz quotient (Lütkepohl, 1997).

Having shown the claim for case $p = \infty$, we now proceed with the proof for cases $p = 1$ and $p = 2$. For that, we need to first introduce the Weyl's spectral inequality (Weyl, 1912). For matrices $P_1, P_2 \in \mathbb{C}^{r \times n}$ and $1 \leq r \leq n$, the inequality states that

$$\lambda_{i+j-1}(P_1 + P_2) \leq \lambda_i(P_1) + \lambda_j(P_2) ,$$

where $\lambda_i(P_1)$ denotes the i-th eigenvalue of matrix P_1 and $1 \leq i, j, i + j - 1 \leq r$.

Now, setting $j = m + 1$, $P_1 = H - H_m$, and $P_2 = H_m$ into the Weyl's inequality we obtain

$$\lambda_{i+m}(H) \leq \lambda_i(H - H_m) + \lambda_{m+1}(H_m) = \lambda_i(H - H_m) . \tag{4.6}$$

From the latter inequality it follows that

$$\|H - H_m\|_p^p = \sum_{i=1}^n \lambda_i (H - H_m)^p \geq \sum_{i=1}^{n-m} \lambda_i (H - H_m)^p \geq \sum_{i=m+1}^n \lambda_i (H)^p = \left\|H - \tilde{H}^*\right\|_p^p ,$$

where $1 \leq p < \infty$ and the latter inequality follows by applying the Weyl's inequality from Eq. (4.6) for $1 \leq i \leq n - m$. □

4.2 K-means Clustering

As our landmark selection strategy for the Nyström low-rank approximation of a kernel matrix is based on K-means clustering (Lloyd, 1982), we first provide a brief review of that algorithm from the perspective of low-rank approximation of Gram matrices and then discuss its relation to Proposition 4.2. In particular, we express the optimization problem for K-means clustering as a constrained low-rank approximation problem (Section 4.2.1) and then give a result which relates the subspace spanned by the top $(K-1)$ left singular vectors of the data matrix and that spanned by optimal K-means centroids (Section 4.2.2). This result (formulated in Proposition 4.5) is the first complete proof of a claim first considered by Ding and He (2004) and one of the main contributions of this chapter.

4.2.1 Optimization Problem

Let the instance space $\mathcal{X} \subset \mathbb{R}^d$ and let K denote the number of clusters. In K-means clustering the goal is to choose a set of centroids $C = \{c_1, \cdots, c_K\}$ minimizing the clustering potential

$$\phi(C) = \sum_{x \in X} \min_{c \in C} \|x - c\|^2 = \sum_{k=1}^{K} \sum_{x \in \mathcal{P}_k} \|x - c_k\|^2 \ ,$$

where $\mathcal{P}_k = \{x \in X \mid \mathcal{P}(x) = c_k\}$ is a clustering cell and $\mathcal{P} \colon \mathcal{X} \to C$ denotes the centroid assignment function. For a clustering cell $\mathcal{P}_k$ the centroid is computed as $\frac{1}{|\mathcal{P}_k|} \sum_{x \in \mathcal{P}_k} x$. In the remainder of the section, we denote with $P \in \mathbb{R}^{n \times K}$ the *cluster indicator matrix* of clustering C such that $p_{ij} = 1/\sqrt{n_j}$ when instance x_i is assigned to centroid c_j, and $p_{ij} = 0$ otherwise. Here n_j denotes the number of instances assigned to centroid c_j.

Without loss of generality, we can assume that the columns of data matrix $X \in \mathbb{R}^{d \times n}$ are centered instances (i.e., $\sum_{i=1}^{n} x_i/n = 0$). Now, using the introduced notation we can write the clustering potential as (Ding and He, 2004; Boutsidis et al., 2009)

$$\phi(C) = \left\| X - XPP^\top \right\|_2^2 \ .$$

Denoting with p_i the i-th column in P we have that it holds $p_i^\top p_j = \delta_{ij}$, where $\delta_{ij} = 1$ if $i = j$ and otherwise $\delta_{ij} = 0$. Hence, it holds that $P^\top P = \mathbb{I}_K$ and P is an orthogonal projection matrix with rank K. Let $\mathcal{C}$ denote the family of all possible clustering indicator matrices of rank K. Then, the K-means optimization problem is equivalent to the constrained low-rank approximation problem

$$P^* = \operatorname*{argmin}_{P \in \mathcal{C}} \ \left\| X - XPP^\top \right\|_2^2 \ .$$

From here, using the relation between the Schatten 2-norm and the matrix trace we obtain

$$P^* = \operatorname*{argmin}_{P \in \mathcal{C}} \ \operatorname{tr}\left(X^\top X\right) - \operatorname{tr}\left(P^\top X^\top X P\right) = \operatorname*{argmax}_{P \in \mathcal{C}} \ \operatorname{tr}\left(P^\top X^\top X P\right) \ . \tag{4.7}$$

In the remainder of the section, we refer to the constrained optimization objective from Eq. (4.7) as the *discrete* problem. For this problem, Ding and He (2004) observe that the set of vectors $\{p_1, \cdots, p_K, \mathbf{e}/\sqrt{n}\}$ is linearly dependent ($\mathbf{e}$ is a vector of ones) and that the rank of the optimization problem can be reduced. As $\sum_{i=1}^{K} \sqrt{n_i} p_i = \mathbf{e}$, there exists a linear orthonormal transformation of the subspace basis given by the columns of P such that one of the vectors in the new basis of the subspace spanned by P is $\mathbf{e}/\sqrt{n}$. Such transformations are equivalent to a rotation of the subspace. Let $R \in \mathbb{R}^{K \times K}$ denote an orthonormal transformation matrix such that the vectors $\{p_i\}_{i=1}^{K}$ map to $\{q_i\}_{i=1}^{K}$ with $q_K = \frac{1}{\sqrt{n}}\mathbf{e}$. This is equivalent to requiring that the K-th column in R is $r_K = \left(\sqrt{n_1/n}, \cdots, \sqrt{n_K/n}\right)^\top$ and $q_i^\top \mathbf{e} = 0$ for $1 \le i \le K-1$. Moreover, from $Q = PR$ and $R^\top R = \mathbb{I}_K$ it follows that

$$Q^\top Q = R^\top P^\top P R = R^\top R = \mathbb{I}_K \ .$$

Hence, if we denote with Q_{K-1} the matrix-block with the first $(K-1)$ columns of Q then the problem from Eq. (4.7) can be written as (Ding and He, 2004; Xu et al., 2015)

$$\begin{aligned} Q_{K-1}^* = \underset{Q_{K-1} \in \mathbb{R}^{n \times (K-1)}}{\operatorname{argmax}} \quad & \operatorname{tr}\left(Q_{K-1}^\top X^\top X Q_{K-1}\right) \\ s.t. \quad & Q_{K-1}^\top Q_{K-1} = \mathbb{I}_{K-1} \\ & Q = PR \ \wedge \ q_K = \frac{1}{\sqrt{n}}\mathbf{e} \ . \end{aligned} \tag{4.8}$$

While P is an orthonormal indicator/sparse matrix of rank K, Q is a piecewise constant and in general non-sparse orthonormal matrix of the same rank. The latter optimization problem can be relaxed by not adding the structural constraints $Q = PR$ and $q_K = \mathbf{e}/\sqrt{n}$. The resulting optimization problem is known as the Rayleigh–Ritz quotient (e.g., see Lütkepohl, 1997) and in the remainder of the section we refer to it as the *continuous* problem. The optimal solution to the continuous problem is (up to a rotation of the basis) defined by the top $(K-1)$ eigenvectors from the eigendecomposition of the positive definite matrix $X^\top X$ and the optimal value of the relaxed optimization objective is the sum of the eigenvalues corresponding to this solution. As the continuous solution is (in general) not sparse, the discrete problem can be better described with the non-sparse piecewise constant matrix Q than with the sparse indicator matrix P.

4.2.2 Relation to Optimal Low-Rank Approximation of Kernel Matrices

Ding and He (2004) and Xu et al. (2015) have formulated a theorem which claims that the subspace spanned by optimal K-means centroids is in fact the subspace spanned by the top $(K-1)$ left singular vectors of X. The proofs provided in these works are, however, restricted to the case when the discrete and continuous/relaxed version of the optimization problem match. We address here this claim without that restriction and amend their formulation accordingly. For this purpose, let $C^* = \{c_1, \cdots, c_K\}$ be K centroids specifying an optimal K-means clustering (i.e., minimizing the potential). The between cluster scatter matrix $S = \sum_{i=1}^{K} n_i c_i c_i^\top$ projects any vector $x \in \mathcal{X}$ to a subspace spanned by the centroid vectors, i.e., $Sx = \sum_{i=1}^{K} n_i \left(c_i^\top x\right) c_i \in \operatorname{span}(\{c_1, \cdots, c_K\})$. Let also λ_K denote the K-th eigenvalue of $H = X^\top X$ and assume the eigenvalues are listed in descending order.

Proposition 4.3. *Suppose that the subspace spanned by optimal K-means centroids has a basis that consists of left singular vectors of X. If the gap between the eigenvalues λ_{K-1} and λ_K is sufficiently large (see the proof for explicit definition), then the optimal K-means centroids and the top $(K-1)$ left singular vectors of X span the same subspace.*

In our proof of the latter proposition, we will need an auxiliary result that allows us to express the clustering potential in terms of the squared norms of centroids. Let us now give this auxiliary claim before we proceed with a proof of Proposition 4.3.

Lemma 4.4. *(Kanungo et al., 2002) Let c be the centroid of a set C with n instances and let z be an arbitrary point from $\mathbb{R}^d$. Then, it holds*

$$\sum_{x \in C} \|x - z\|^2 - \sum_{x \in C} \|x - c\|^2 = n \|c - z\|^2 \ .$$

Proof. After expanding the sums we obtain

$$\sum_{x \in C} \|x\|^2 + n \|z\|^2 - 2 \sum_{x \in C} \langle x, z \rangle - \sum_{x \in C} \|x\|^2 - n \|c\|^2 + 2 \sum_{x \in C} \langle c, x \rangle = n \|c\|^2 + n \|z\|^2 - 2n \langle c, z \rangle \ .$$

We can now rewrite the latter equation as

$$-2n \langle c, z \rangle - n \|c\|^2 + 2n \|c\|^2 = n \|c\|^2 - 2n \langle c, z \rangle \ ,$$

and the claim follows from here. □

Having provided a proof for Lemma 4.4, we are now ready to prove Proposition 4.3.

Proof of Proposition 4.3. Let $M \in \mathbb{R}^{d \times K}$ be a matrix with centroids $\{c_1, c_2, \ldots, c_K\}$ as columns and let $N = \operatorname{diag}\{n_1, n_2, \cdots, n_K\}$, where n_i denotes the number of instances assigned to centroid c_i. Now, observe that $M = XPN^{-1/2}$ and that we can write the non-constant term from Eq. (4.7) as

$$\operatorname{tr}\left(P^\top X^\top X P\right) = \operatorname{tr}\left(N^{\frac{1}{2}} M^\top M N^{\frac{1}{2}}\right) = \operatorname{tr}\left(M N M^\top\right) = \operatorname{tr}\left(\sum_{i=1}^{K} n_i c_i c_i^\top\right) \ . \tag{4.9}$$

An optimal solution to K-means clustering places centroids to maximize this objective. From the relaxed version of the problem, defined in Eq. (4.8), we know that it holds

$$\operatorname{tr}\left(\sum_{i=1}^{K} n_i c_i c_i^\top\right) \leq \sum_{i=1}^{K-1} \lambda_i \ , \tag{4.10}$$

where $\{\lambda_i\}_{i=1}^{K-1}$ are the top eigenvalues of the eigendecomposition $XX^\top = U \Lambda U^\top$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$. Moreover, from the singular value decomposition, $X = U \Sigma V^\top$, it follows that $X = \sum_{i=1}^{r} \sigma_i u_i v_i^\top$, where r is the rank of X and $r \leq \min(d-1, n-1)$. The latter follows from the assumption that X is a centered data matrix. Hence, $U \in \mathbb{R}^{d \times r}$ is an orthonormal basis of the data span and we can express the centroids in this basis as $M = U\Gamma$, where $\Gamma = [\gamma_{\cdot 1} \cdots \gamma_{\cdot K}]$ and $\gamma_{\cdot i} \in \mathbb{R}^r$ for $i = 1, \ldots, K$.

Having expressed the centroids in the U-basis it is now possible to rewrite the optimization objective in this basis, as well. In particular, it holds

$$\begin{aligned} \operatorname{tr}\left(M N M^\top\right) &= \operatorname{tr}\left(U \Gamma N \Gamma^\top U^\top\right) = \operatorname{tr}\left(\Gamma N \Gamma^\top\right) = \operatorname{tr}\left(\sum_{i=1}^{K} n_i \gamma_i \gamma_i^\top\right) \\ &= \sum_{i=1}^{K} n_i \sum_{j=1}^{r} \gamma_{ji}^2 = \sum_{j=1}^{r} \sum_{i=1}^{K} n_i \gamma_{ji}^2 \ . \end{aligned} \tag{4.11}$$

From the singular value decomposition of X it is possible to compute the projections of instances onto the left singular vectors, and thus retrieve the coefficient matrix Γ. For instance, projecting the data over a left singular vector u_j we get

$$u_j^\top X = \sigma_j v_j^\top \ ,$$

where $v_j \in \mathbb{R}^n$ is the j-th right singular vector of X. The centroid c_i of a cluster cell $\mathcal{P}_i$ can be expressed in the U-basis by setting

$$\gamma_{ji} = \frac{\sigma_j}{n_i} \sum_{l \in \mathcal{P}_i} v_{lj} \; .$$

From here, we then have that

$$n_i \gamma_{ji}^2 = \lambda_j n_i \left(\frac{\sum_{l \in \mathcal{P}_i} v_{lj}}{n_i} \right)^2 = \lambda_j n_i \delta_{ji}^2 \; ,$$

where $\delta_{ji} = \sum_{l \in \mathcal{P}_i} v_{lj} / n_i$. The latter equation, on the other hand, allows us to write Eq. (4.11) as

$$\operatorname{tr}\left(M N M^\top \right) = \sum_{j=1}^{r} \lambda_j \sum_{i=1}^{K} n_i \delta_{ji}^2 \; . \tag{4.12}$$

From the Cauchy–Schwartz inequality and the fact that right singular vectors of X are orthonormal vectors it follows that

$$n_i \delta_{ji}^2 \leq \sum_{l \in \mathcal{P}_i} v_{lj}^2 \leq 1 \; . \tag{4.13}$$

On the one hand, from this inequality we can conclude that $\delta_{ji} \leq 1/\sqrt{n_i}$. On the other hand, summing the first part of the inequality over $1 \leq i \leq K$ we obtain

$$\sum_{i=1}^{K} n_i \delta_{ji}^2 \leq \sum_{i=1}^{K} \sum_{l \in \mathcal{P}_i} v_{lj}^2 = \left\| v_j \right\|^2 = 1 \; .$$

As the data matrix X is centered, i.e., $\frac{1}{n} \sum_{i=1}^{n} x_i = 0$, it follows that the columns of the matrix M are linearly dependent. In particular, we have that it holds

$$\sum_{i=1}^{K} \frac{n_i}{n} c_i = \frac{1}{n} \sum_{i=1}^{n} x_i = 0 \; .$$

From here it then follows that we can express one column (e.g., the centroid c_K) as a linear combination of the others. Thus, the rank of M is at most $K - 1 \leq r$. As the rank of $\operatorname{span}\{c_1, c_2, \cdots, c_K\}$ is at most $K - 1$, then by the assumption of the proposition there are at least $r - K + 1$ columns of U that are orthogonal to the span. Consequently, in matrix $\Gamma \in \mathbb{R}^{r \times K}$ there are at least $(r - K + 1)$ rows with all entries equal to zero.

Now, the problem of minimizing the objective in Eq. (4.7) is equivalent to that of maximizing the objective in Eq. (4.11). By the assumption of the proposition, the latter is equivalent to setting the rows in Γ corresponding to low value terms in Eq. (4.12) to zero vectors. As

$$\lambda_j \sum_{i=1}^{K} n_i \delta_{ji}^2 \leq \lambda_j \; ,$$

the optimization with respect to upper bounds on terms in Eq. (4.12) is equivalent to setting to zero the rows in Γ that correspond to eigenvalues λ_j with $j \geq K$, i.e., $\gamma_{ji} = \delta_{ji} = 0$ for $K \leq j \leq r$ and $1 \leq i \leq K$. Let us now check whether and under what conditions the optimal value of the relaxed objective, $\sum_{j=1}^{K-1}\lambda_j$, is attained for this choice of centroids. By applying Lemma 4.4 with $z = 0$ to Eq. (4.12) we obtain

$$\sum_{j=1}^{r}\lambda_j\sum_{i=1}^{K}n_i\delta_{ji}^2 = \sum_{j=1}^{r}\lambda_j\sum_{i=1}^{K}\sum_{l\in\mathcal{P}_i}v_{lj}^2 - \left(v_{lj}-\delta_{ji}\right)^2 = \sum_{j=1}^{r}\lambda_j\left(1-\sum_{i=1}^{K}\sum_{l\in\mathcal{P}_i}\left(v_{lj}-\delta_{ji}\right)^2\right).$$

The maximal value of this objective is attained if the top $(K-1)$ right singular vectors V are piecewise constant over clusters. In particular, for $v_{lj} = \delta_{ji}$ with $l \in \mathcal{P}_i$, $1 \leq i \leq K$, and $1 \leq j \leq K-1$, the expression attains the maximal value of the continuous version of the problem, $\sum_{j=1}^{K-1}\lambda_j$. Thus, if the top $(K-1)$ right singular vectors are piecewise constant the solutions to the discrete and continuous version of the K-means optimization problem match.

However, right singular vectors of X are not necessarily piecewise constant and a solution based on the optimization with respect to upper bounds on terms in Eq. (4.12) might not be optimal. To establish under what conditions the subspace spanned by the top $(K-1)$ left singular vectors is identical to that spanned by optimal K-means centroids, we consider two clusterings $C_K^{(1)}$ and $C_K^{(2)}$. According to the assumption of the proposition, the subspace spanned by optimal K-means centroids has a basis consisting of left singular vectors. Thus, we assume that $\{u_1,\cdots,u_{K-2},u_{K-1}\}$ and $\{u_1,\cdots,u_{K-2},u_K\}$ are subspaces spanned by centroids in $C_K^{(1)}$ and $C_K^{(2)}$, respectively. Let us also denote with V_{K-1} the matrix with top $(K-1)$ right singular vectors of X and let $\tilde{V}_1$ and $\tilde{V}_2$ be the piecewise constant approximations to right singular vectors corresponding to the subspaces spanned by sets of centroids $C_K^{(1)}$ and $C_K^{(2)}$.

Taking $\tilde{v}_j$ to be a column vector given by $\tilde{v}_{lj} = \delta_{ji}$ with $l \in \mathcal{P}_i$ and $1 \leq i \leq K$, we can write Eq. (4.11) as

$$\operatorname{tr}\left(MNM^\top\right) = \sum_{j=1}^{r}\lambda_j\left(1-\left\|v_j-\tilde{v}_j\right\|^2\right).$$

Now, if $\tilde{V}_1 \neq V_{K-1}$ and the gap between eigenvalues λ_{K-1} and λ_K is sufficiently large then the choice of coefficients $\gamma_{ji} \neq 0$ with $1 \leq j \leq K-1$ corresponds to an optimal K-means clustering and the corresponding centroid subspace is spanned by the top $(K-1)$ left singular vectors of X. More specifically, the latter claim holds if the gap between the eigenvalues λ_{K-1} and λ_K satisfies

$$\lambda_{K-1}(1-\|v_{K-1}-\tilde{v}_{K-1}^{(1)}\|^2) > \lambda_K(1-\|v_K-\tilde{v}_K^{(2)}\|^2),$$

where v_j and $\tilde{v}_j^{(\cdot)}$ denote corresponding columns in matrices V_{K-1} and $\tilde{V}_\cdot$, respectively. If $\left\|v_K-\tilde{v}_K^{(2)}\right\| < 1$, the latter inequality is equivalent to

$$\frac{\lambda_{K-1}-\lambda_K}{\lambda_{K-1}} > \frac{\left\|v_{K-1}-\tilde{v}_{K-1}^{(1)}\right\|^2-\left\|v_K-\tilde{v}_K^{(2)}\right\|^2}{1-\left\|v_K-\tilde{v}_K^{(2)}\right\|^2}.$$

To see that the condition $\left\|v_K-\tilde{v}_K^{(2)}\right\| < 1$ is satisfied note that

$$0 < \sum_{i=1}^{K}n_i\delta_{ji}^2 = \sum_{i=1}^{K}\sum_{l\in\mathcal{P}_i}v_{lj}^2-\left(v_{lj}-\delta_{ji}\right)^2 = 1-\left\|v_K-\tilde{v}_K^{(2)}\right\|^2.$$

Having established a condition on the gap between the eigenvalues λ_{K-1} and λ_K, let us now check whether the upper bound from Eq. (4.10) is attained in cases when right singular vectors are not piecewise constant. From the Cauchy-Schwarz inequality in Eq. (4.13), it follows that the equality is attained when v_{lj} = const. for all $l \in \mathcal{P}_i$ and $1 \le i \le K$. However, we have assumed that right singular vectors are not piecewise constant and this implies that the strict inequality holds in Eq. (4.13). Consequently, for non-constant right singular vectors we have that the optimal value of the relaxed problem is not attained, i.e.,

$$\operatorname{tr}\left(MNM^\top\right) = \sum_{j=1}^{K-1} \lambda_j \sum_{i=1}^{K} n_i \delta_{ji}^2 < \sum_{j=1}^{K-1} \lambda_j \, .$$

□

Let us now relate Proposition 4.3 to the Eckart–Young–Mirsky theorem, reviewed in Section 4.1. The theorem implies that an optimal set of landmarks for the Nyström approximation of a kernel matrix spans the subspace of the kernel feature space which preserves most of the variation present in the dataset (e.g., see Chapter 2). Assuming that the conditions from Proposition 4.3 are satisfied, then the Nyström approximation using optimal kernel K-means centroids as landmarks projects the data to a subspace which preserves the maximal possible amount of variation in the dataset. Hence, under these conditions optimal kernel K-means landmarks provide an optimal rank $(K-1)$ reconstruction of the kernel matrix.

Having established this relation, we now proceed to a general case for which the assumption on the basis of the subspace spanned by optimal K-means centroids does not hold. In this more realistic case, we show that the claim by Ding and He (2004) and Xu et al. (2015) on the relation between the subspace spanned by the top $(K-1)$ left singular vectors of X and that spanned by optimal K-means centroids does not hold for all data matrices.

Proposition 4.5. *In contrast to the claim by Ding and He (2004) and Xu et al. (2015), it is possible that no basis of the subspace spanned by optimal K-means centroids consists of left singular vectors of X. In that case, the subspace spanned by the top $(K-1)$ left singular vectors is different from that spanned by optimal K-means centroids.*

Proof. If no basis of $\operatorname{span}\{c_1, c_2, \ldots, c_K\}$ is given by a subset of left singular vectors, then (using the notation from the proof of Proposition 4.3) there are at least K rows with non-zero entries in matrix Γ. Let us now show that this is indeed possible. The fact that a left singular vector u_i is orthogonal to the span is equivalent to

$$\left(\forall \beta \in \mathbb{R}^K\right) \colon \; 0 = u_i^\top \left(\sum_{j=1}^{K} \beta_j c_j \right) = u_i^\top \left(\sum_{j=1}^{K} \sum_{l=1}^{r} \beta_j \delta_{lj} \sigma_l u_l \right) = \sum_{j=1}^{K} \beta_j \sigma_i \delta_{ij} = \sum_{j=1}^{K} \beta_j \sigma_i \frac{1}{n_j} \sum_{l \in \mathcal{P}_j} v_{li} \, ,$$

where v_i is the i-th right singular vector. As the latter equation holds for all vectors $\beta = (\beta_1, \ldots, \beta_K) \in \mathbb{R}^K$, the claim $u_i \perp \operatorname{span}\{c_1, \ldots, c_K\}$ is equivalent to

$$\sum_{l \in \mathcal{P}_j} v_{li} = 0 \quad (\forall j = 1, \ldots, K) \, . \tag{4.14}$$

Moreover, as the data matrix is centered the vector v_i also satisfies $v_i^\top \mathbf{e} = 0$.

To construct a problem instance where no basis of the subspace spanned by optimal K-means centroids consists of left singular vectors, we take a unit vector v_r such that, for any cluster in any clustering, none of the conditions from Eq. (4.14) is satisfied. Then, we can construct a basis of right singular vectors using the Gram–Schmidt orthogonalization method. For instance, we can take $\tilde{v}$ with $\tilde{v}_i = -2^i$ for $1 \leq i < n$ and $\tilde{v}_n = 2^n - 2$, and then set $v_r = \tilde{v}/\|\tilde{v}\|$, where r is the rank of the problem. Once we have constructed a right singular basis that contains the vector v_r, we pick a small positive real value as the singular value corresponding to the vector v_r and select the remaining singular values so that there are sufficiently large gaps between them (e.g., see the proof of Proposition 4.3). By choosing a left singular basis of rank r, we form a data matrix X and the subspace spanned by optimal K-means centroids in this problem instance is not the one spanned by the top $(K-1)$ left singular vectors. To see this, note that from Eq. (4.14) and the definition of v_r it follows that $u_r \not\perp \operatorname{span}\{c_1, \dots, c_K\}$.

Having shown that an optimal centroid subspace of data matrix X is not the one spanned by the top $(K-1)$ left singular vectors, let us now show that there is no basis for this subspace consisting of left singular vectors. For simplicity, let us take $K = 2$. According to our assumption $\sigma_1 \gg \sigma_2 \gg \sigma_{r-1} \gg \sigma_r$. Now, from Eq. (4.12) it follows that the largest reduction in the clustering potential is obtained by partitioning data so that the centroids for the top components are far away from the zero-vector. As the basis of $\operatorname{span}\{c_1, c_2\}$ consists of one vector and as $u_r \not\perp \operatorname{span}\{c_1, c_2\}$ it then follows that the basis vector is given by $\sum_{j=1}^{r} \beta_j u_j$ with $\beta_j \in \mathbb{R}$ and at least $\beta_1, \beta_r \neq 0$. Hence, for $K = 2$ and data matrix X there is no basis of $\operatorname{span}\{c_1, c_2\}$ that consists of a left singular vector. □

Thus, there are K-means clustering problems where optimal K-means centroids span a subspace different from the one spanned by the top $(K-1)$ left singular vectors. In such cases, similar to Proposition 4.3, an optimal clustering partitions the data so that the components of the centroids on the top left singular vectors are not zero. For some data distributions, the latter amounts to selecting optimal centroids so that the corresponding centroid subspace is close to that spanned by the top $(K-1)$ left singular vectors.

4.3 Nyström Method with Kernel K-means++ Landmarks

In Section 4.2.2, we have seen that instances which map to optimal kernel K-means centroids can be effective landmarks for the Nyström low-rank approximation of a kernel matrix. However, for a kernel K-means centroid there does not necessarily exist a point in the instance space that maps to it (Burges, 1999). To account for this and the hardness of the kernel K-means clustering problem (Aloise et al., 2009), as well as the computational complexity of the Lloyd refinements (Lloyd, 1982), we propose to approximate the centroids with kernel K-means++ samples (Arthur and Vassilvitskii, 2007). We start with a brief overview of the K-means++ sampling scheme (Arthur and Vassilvitskii, 2007) for seeding of centroids in K-means clustering (Section 4.3.1). Following this, we give a pseudo-code description of a kernelized version of this sampling scheme together with an analysis of its computational and space complexities (Section 4.3.2). The section concludes with a bound on the relative approximation error of the Nyström method for low-rank approximation of kernel matrices with kernel K-means++ samples as landmarks (Section 4.3.3).

4.3.1 K-means++ Sampling Scheme

In this section, we review the K-means++ sampling scheme (Arthur and Vassilvitskii, 2007) proposed for seeding of initial clusters in the Lloyd's algorithm for K-means clustering (Lloyd, 1982). The main idea behind the sampling scheme is to approximate an optimal clustering with a randomized greedy algorithm that in each iteration selects an instance with probability proportional to its contribution to the clustering potential in which previously selected instances act as cluster centroids. We describe this sampling scheme below and review a relative error bound on its clustering potential given by Arthur and Vassilvitskii (2007).

Let us begin by denoting the intermediate clustering solution constructed using the K-means++ sampling scheme at step t with $C_t = \{c_1, \dots, c_t\}$, where $1 \leq t \leq K$ and K is the desired number of clusters. For an instance $x \in X$, the contribution to the clustering potential at the step $t > 1$ is given by

$$D_t(x) = \min_{c \in C_{t-1}} \|x - c\|^2 \ . \tag{4.15}$$

Thus, the centroid c_t with $t > 1$ is selected by sampling an instance $x \in X$ with probability

$$p_t(x) = \frac{D_t(x)}{\phi(C_{t-1})} \ .$$

In the step $t = 1$, there are no previously selected instances that act as centroids and the centroid c_1 is selected by sampling an instance $x \in X$ using the uniform distribution

$$p_1(x) = \frac{1}{n} \ .$$

Hence, the probability of selecting a set of centroids $C_K = \{c_1, \dots, c_K\}$ is then given by

$$p(C_K) = p(c_K \mid C_{K-1}) p(C_{K-1}) = p_K(c_K) p(C_{K-1}) = \prod_{i=1}^{K} p_i(c_i) \ .$$

Having reviewed the K-means++ sampling scheme, we now provide a bound on the relative error for the approximation of the optimal clustering using this sampling scheme.

Theorem 4.6. *(Arthur and Vassilvitskii, 2007) If a clustering C is constructed using the K-means++ sampling scheme then the corresponding clustering potential $\phi(C)$ satisfies*

$$\mathbb{E}_{C \sim p(\cdot)} \left[\frac{\phi(C)}{\phi(C^*)} \right] \leq 8 (\ln K + 2) \ ,$$

where C^* is an optimal clustering with K centroids.

The computation complexity of the sampling scheme is $\mathcal{O}(Knd)$, where K denotes the number of clusters and d is the dimension of the problem. To see this, first observe that in each iteration the K-means++ sampling scheme computes the contribution of each instance to the current clustering potential. An efficient implementation of the scheme stores this vector with individual contributions to the clustering potential in the memory and updates the vector in each iteration to account for a newly added centroid (i.e., the instance sampled at that iteration). In this way, the computational complexity of an iteration is $\mathcal{O}(nd)$, where n arises from the number of instances and d from the computation of the squared distance to the selected centroid. As there are in total K such iterations, the algorithm constructs a clustering with K centroids in time $\mathcal{O}(Knd)$.

Algorithm 4.1 Kernel K-means++ Sampling Scheme

```
Input: sample $X = \{x_1, \dots, x_n\}$, kernel function $h \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, number of clusters $K \in \mathbb{N}$
Output: cluster centroids $\{c_1, \dots, c_K\}$
 1: $D_i \leftarrow \infty$ for $i = 1, \dots, n$
 2: $i_1 \sim \mathcal{U}_{[1,\dots,n]}$ and $c_1 \leftarrow x_{i_1}$
 3: for $k = 2, \dots, K$ do
 4:     $\phi \leftarrow 0$
 5:     for $i = 1, \dots, n$ do
 6:         $d_i \leftarrow h(x_i, x_i) + h(c_{k-1}, c_{k-1}) - 2h(x_i, c_{k-1})$
 7:         if $D_i > d_i$ then $D_i \leftarrow d_i$ end if
 8:         $\phi \leftarrow \phi + D_i$
 9:     end for
10:     $p(i) \leftarrow D_i/\phi$ for $i = 1, \dots, n$
11:     $i_k \sim p(\cdot)$ and $c_k \leftarrow x_{i_k}$
12: end for
```

4.3.2 Kernel K-means++ Landmarks

Algorithm 4.1 provides a pseudo-code description of the kernel K-means++ sampling scheme for landmark selection in the Nyström method for low-rank approximation of kernel matrices. The algorithm takes as input a set of n instances sampled independently from a Borel probability measure defined on the instance space $\mathcal{X}$, together with a Mercer kernel function $h \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and a number of clusters $K \in \mathbb{N}$ with $1 < K < n$.

The constructive process starts by initializing the contributions of instances to the clustering potential to a very large number (line 1). Then, an instance is sampled uniformly at random from the set of available instances as the first centroid (line 2). Following this, the algorithm starts iterating until the specified number of centroids is selected (lines 3–12). In the first step of each iteration, the clustering potential ϕ is set to zero (line 4). Then, for each instance the algorithm computes the squared distance to the centroid selected in the previous iteration (line 6). If the computed distance is smaller than the current contribution of the instance to the clustering potential, the algorithm updates the corresponding contribution to the clustering potential (lines 7). To account for the updates in the contributions of instances to the clustering potential, the algorithm recomputes the clustering potential in each iteration (line 8). Having updated the clustering potential and individual contributions of instances to it, the algorithm defines a discrete distribution over the set of available instances so that the probability of selecting an instance is proportional to its contribution to the clustering potential (line 10). At the last step of each iteration, the algorithm samples a new centroid from the set of available instances using the previously defined discrete distribution (line 11).

From the pseudo-code description, we can observe that the algorithm does not need to store the whole matrix in order to sample the centroids. More specifically, if the diagonal entries of the kernel matrix required for the computation of the squared distances in line 6 are precomputed/cached, then in each iteration of the algorithm only one row of the kernel matrix needs to be computed (i.e., the row corresponding to the instance c_{k-1}). Thus, the space complexity of the approach is $\mathcal{O}(nd)$, where d is the dimension of the data and it originates from storing the instances that are given as input to the algorithm.

From the computational perspective, the most time-consuming step is the computation of the squared distances (line 6). This step depends on the provided kernel function and it can be typically computed in $\mathcal{O}(d)$ time. As this computation is repeated for each instance, the computational complexity of that sequence of steps is $\mathcal{O}(nd)$. Thus, the computational complexity of selecting K centroids using the kernel K-means++ sampling scheme is $\mathcal{O}(Knd)$.

To further improve the estimate of optimal centroids, it is possible to implement the kernel K-means++ scheme with *local restarts*. What this means is that in line 11 instead of sampling a single index, the algorithm samples q candidate indices (not necessarily unique). Then, for each of the corresponding candidate instances the algorithm computes the reduction in the current clustering potential that comes as a result of adding a new centroid. A candidate instance corresponding to the largest reduction in the clustering potential is then selected as the new centroid (ties are broken arbitrarily). Local restarts add a computational overhead of $\mathcal{O}(qnd)$ to each iteration of the kernel K-means++ sampling scheme. Thus, by following Arthur and Vassilvitskii (2007) and setting $q = 2 + \ln K$ the computational complexity of the kernel K-means++ sampling scheme with local restarts is $\mathcal{O}(ndK \ln K)$.

4.3.3 Theoretical Analysis

Let us begin by relating the optimal clustering potential to the optimal low-rank approximation error of the Nyström method. For that, we perform a singular value decomposition of the data matrix, $X = U\Sigma V^\top$, and denote with U_K the top K left singular vectors from that decomposition. Let also $U_K^\perp$ denote the dual matrix of U_K and $\phi(C^* \mid U_K)$ the clustering potential given by the projections of X and C^* onto the subspace U_K (see the proof of Proposition 4.7 for the explicit definition). The following proposition gives an upper and lower bound on the optimal clustering potential in terms of the optimal low-rank approximation error of the Nyström method, expressed in the Schatten 1-norm.

Proposition 4.7. *Let H_K denote the optimal rank K approximation of the Gram matrix $H = X^\top X$ and let C^* be an optimal K-means clustering of X. Then, it holds*

$$\|H - H_{K-1}\|_1 \leq \phi(C^*) \leq \|H - H_{K-1}\|_1 + \phi(C^* \mid U_{K-1}).$$

Proof. We first prove the left-hand side inequality. For that, let us consider the optimization problem in Eq. (4.8). The clustering potential attains its minimal value $\phi(C^*)$ when the optimization objective from Eq. (4.8) is maximized. The upper bound on the maximal value in that optimization problem is given by the optimal value of the corresponding relaxed version of the problem. As already stated in Section 4.2.1, the relaxed version of that optimization problem is known as the Rayleigh–Ritz quotient and its maximal value is $\sum_{j=1}^{K-1} \lambda_j$. Thus, we have that it holds

$$\phi(C^*) \geq \operatorname{tr}\left(X^\top X\right) - \sum_{j=1}^{K-1} \lambda_j = \sum_{j=K}^{r} \lambda_j = \|H - H_{K-1}\|_1 \ .$$

Having shown the left-hand side inequality, let us now turn our attention to proving the inequality on the right-hand side. From the proof of Proposition 4.3, we have that

$$\phi(C^*) = \sum_{j=1}^{r} \lambda_j - \sum_{j=1}^{r} \lambda_j \left(1 - \sum_{i=1}^{K} \sum_{l \in \mathcal{P}_i} \left(v_{lj} - \delta_{ji}\right)^2\right) = \sum_{j=1}^{r} \lambda_j \sum_{i=1}^{K} \sum_{l \in \mathcal{P}_i} \left(v_{lj} - \delta_{ji}\right)^2 \ .$$

Now, observe that

$$0 \leq \delta_{ji}^2 \quad \Longleftrightarrow \quad n_i \delta_{ji}^2 \leq 2\delta_{ji} \sum_{l \in \mathcal{P}_i} v_{lj} \quad \Longleftrightarrow \quad \sum_{l \in \mathcal{P}_i} \left(v_{lj} - \delta_{ji}\right)^2 \leq \sum_{l \in \mathcal{P}_i} v_{lj}^2 \ .$$

Thus, we have that it holds

$$\phi\left(C^* \mid U_{K-1}^{\perp}\right) = \sum_{j=K}^{r} \lambda_j \sum_{i=1}^{K} \sum_{l \in \mathcal{P}_i} \left(v_{lj} - \delta_{ji}\right)^2 \leq \sum_{j=K}^{r} \lambda_j \sum_{i=1}^{K} \sum_{l \in \mathcal{P}_i} v_{lj}^2 = \sum_{j=K}^{r} \lambda_j = \|H - H_{K-1}\|_1 \ .$$

The claim follows by combining the latter inequality with the fact that

$$\phi\left(C^* \mid U_{K-1}\right) = \sum_{j=1}^{K-1} \lambda_j \sum_{i=1}^{K} \sum_{l \in \mathcal{P}_i} \left(v_{lj} - \delta_{ji}\right)^2 \ .$$

□

Having presented all the relevant results, we now give a bound on the approximation error of the Nyström method with kernel K-means++ samples as landmarks.

Theorem 4.8. *Let H be a kernel matrix with a finite rank factorization $H = \Phi(X)^\top \Phi(X)$. Denote with H_K the optimal rank K approximation of H and let $\tilde{H}_K$ be the Nyström approximation of the same rank obtained using kernel K-means++ samples as landmarks. Then,*

$$\mathbb{E}\left[\frac{\|H - \tilde{H}_K\|_2}{\|H - H_K\|_2}\right] \leq 8(\ln(K+1) + 2)(\sqrt{n-K} + \Theta_K),$$

where $\Theta_K = \phi(C^* | U_K)/\|H - H_K\|_2$, U_K denotes the top K left singular vectors of $\Phi(X)$, and C^* optimal kernel K-means clustering with $(K+1)$ clusters.

Proof. Let us assume that $(K+1)$ landmarks, $Z \subset X$, are selected using the kernel K-means++ sampling scheme. Then, for the clustering potential defined with Z we have that it holds

$$\phi(Z) = \sum_{i=1}^{n} \min_{z \in Z} \|\Phi(x_i) - \Phi(z)\|^2 \geq \min_{\alpha \in \mathbb{R}^{(K+1)\times n}} \sum_{i=1}^{n} \left\|\Phi(x_i) - \sum_{j=1}^{K+1} \alpha_{ji} \Phi\left(z_j\right)\right\|^2 = \left\|H - \tilde{H}_K\right\|_1 \ ,$$

where $\tilde{H}_K$ is the Nyström approximation matrix (e.g., see Section 4.1.3) of rank K defined with landmarks $Z = \{z_1, \ldots, z_{K+1}\}$ and $\Phi(x)$ is the image of instance x in the factorization space. The latter inequality follows from the fact that the distance of a point to its orthogonal projection onto span $\{\Phi(z_1), \ldots, \Phi(z_{K+1})\}$ is not greater than the distance between that point and the closest landmark from $\{\Phi(z_1), \ldots, \Phi(z_{K+1})\}$.

Now, combining this result with Theorem 4.6 and Proposition 4.7 we deduce

$$\mathbb{E}\left[\|H - \tilde{H}_K\|_1\right] \leq \mathbb{E}[\phi(Z)] \leq 8(\ln(K+1) + 2)(\|H - H_K\|_1 + \phi(C^* \mid U_K)) \ .$$

From this and the Schatten p-norm inequalities (Weidmann, 1980),

$$\|H - H_K\|_1 \leq \sqrt{n-K}\, \|H - H_K\|_2 \quad \wedge \quad \|H\|_2 \leq \|H\|_1 \ ,$$

we obtain the following bound

$$\mathbb{E}\left[\|H - \tilde{H}_K\|_2\right] \leq 8(\ln(K+1) + 2)\left(\sqrt{n-K}\, \|H - H_K\|_2 + \phi(C^* \mid U_K)\right) \ .$$

The result follows after division by $\|H - H_K\|_2$. □

Corollary 4.9. *If $\phi(C^* \mid U_K) \leq \sqrt{n-K}\,||H - H_K||_2$, then the additive term $\Theta_K \leq \sqrt{n-K}$ and*

$$\mathbb{E}\left[\frac{||H-\tilde{H}_K||_2}{||H-H_K||_2}\right] \in \mathcal{O}\left(\ln K\sqrt{n-K}\right). \tag{4.16}$$

The given bound for low-rank approximation of symmetric and positive definite matrices holds for the Nyström method with kernel K-means++ samples as landmarks *without any Lloyd iterations* (Lloyd, 1982). To obtain even better landmarks, it is possible to first sample candidates using the kernel K-means++ sampling scheme and then attempt a Lloyd refinement in the instance space (motivation for this is provided in Section 4.4.3). If the clustering potential is decreased as a result of this, the iteration is considered successful and the landmarks are updated. Otherwise, the refinement is rejected and current candidates are selected as landmarks. This is one of the landmark selection strategies we analyze in our experiments (e.g., see Section 4.6).

We conclude the section with an insight into the properties of our bound with respect to the rank of the approximation. From Corollary 4.9 it follows that the bound on the relative approximation error increases initially (for small K) with $\ln K$ and then decreases as K approaches n. This is to be expected as a larger K means we are trying to find a higher dimensional subspace and initially this results in having to solve a more difficult problem. The bound on the low-rank approximation error is, on the other hand, obtained by multiplying with $||H - H_K||_2$ which depends on the spectrum of the kernel matrix and decreases with K. In order to be able to generalize at all, one has to assume that the spectrum falls rather sharply and typical assumptions are $\lambda_i \in \mathcal{O}(i^{-a})$ with $a > 1$ or $\lambda_i \in \mathcal{O}(e^{-bi})$ with $b > 0$ (e.g., see Section 4.3, Bach, 2013). The following corollary shows that for $a \geq 2$, $K > 1$, and $\lambda_i \in \mathcal{O}(i^{-a})$ such falls are sharper than $\ln K$.

Corollary 4.10. *Assume that the eigenvalues of the kernel matrix H satisfy $\lambda_i \in \mathcal{O}(i^{-a})$ with $a \geq 2$. The low-rank approximation error in the Frobenius norm of the Nyström method with kernel K-means++ samples as landmarks decreases with $K > 1$ as*

$$\mathcal{O}\left(\frac{\sqrt{n-K}\,(\ln(K+1)+1)}{(K+1)^{a-1}}\right).$$

Proof. First observe that

$$\begin{aligned}\sum_{l=K}^{n}\frac{1}{l^{2a}} &= \frac{1}{K^{2a}}\sum_{l=K}^{n}\frac{1}{(l/K)^{2a}} = \frac{1}{K^{2a}}\sum_{l=0}^{n-K}\frac{1}{(1+l/K)^{2a}} < \frac{1}{K^{2a}}\sum_{l=0}^{n-K}\frac{1}{(1+l/K)^{2}} \\ &< \frac{1}{K^{2(a-1)}}\sum_{l=0}^{n-K}\frac{1}{(1+l)^{2}} < \frac{1}{K^{2(a-1)}}\sum_{l\geq 0}\frac{1}{(1+l)^{2}} \in \mathcal{O}\left(\frac{1}{K^{2(a-1)}}\right).\end{aligned}$$

Hence, we deduce that the approximation error in the Frobenius norm of the optimal rank K subspace satisfies

$$||H - H_K||_2 \in \mathcal{O}\left(\frac{1}{(K+1)^{a-1}}\right).$$

From here it then follows that the low-rank approximation error in Frobenius norm of the Nyström method with kernel K-means++ samples as landmarks satisfies

$$\left\|H - \tilde{H}_K\right\|_2 \in \mathcal{O}\left(\frac{\sqrt{n-K}\,(\ln(K+1)+1)}{(K+1)^{a-1}}\right).$$

The claim follows by observing that for $a \geq 2$ the function $\frac{\ln(K+1)}{(K+1)^{a-1}}$ decreases with $K > 1$. □

We note here that a similar state-of-the-art bound (discussed subsequently) on the relative approximation error by Li et al. (2016) exhibits worse behavior and grows linearly with K.

4.4 Discussion

We start with a brief overview of alternative approaches to landmark selection in the Nyström method for low-rank approximation of kernel matrices. Following this, we focus on a bound that is the most similar to ours, that of K-DPP-Nyström (Li et al., 2016). Then, for the frequently used Gaussian kernel, we provide a theoretically sound motivation for performing the Lloyd refinements of kernel K-means++ landmarks in the instance space instead of the kernel feature space. These refinements are computationally cheaper than the ones performed in the kernel feature space and can only improve the positioning of the landmarks.

4.4.1 Related Approaches

As pointed in Section 4.1.4, the choice of landmarks is crucial for the quality of the Nyström low-rank approximations. For this reason, the existing work on the Nyström method has focused mainly on landmark selection techniques with theoretical guarantees. These approaches can be divided into four groups: *i*) random sampling, *ii*) greedy methods, *iii*) methods based on the Cholesky decomposition, *iv*) vector quantization (e.g., K-means clustering).

The simplest strategy for choosing the landmarks is by uniformly sampling them from a given set of instances. This was the strategy that was proposed by Williams and Seeger (2001) in the first paper on the Nyström method for low-rank approximation of kernel matrices. Following this, more sophisticated non-uniform sampling schemes were proposed. The schemes that received a lot of attention over the past years are the selection of landmarks by sampling proportional to column norms of the kernel matrix (Drineas et al., 2006), diagonal entries of the kernel matrix (Drineas and Mahoney, 2005), approximate leverage scores (Alaoui and Mahoney, 2015; Gittens and Mahoney, 2016), and submatrix determinants (Belabbas and Wolfe, 2009; Li et al., 2016). From this group of methods, the approximate leverage score sampling and the K-DPP Nyström method (see Section 4.4.2) are considered state-of-the-art methods in low-rank approximation of kernel matrices.

The second group of landmark selection techniques are greedy methods. A well-performing representative from this group is a method for sparse approximations proposed by Smola and Schölkopf (2000) for which it was later independently established (Kumar et al., 2012) that it performs very well in practice—second only to K-means clustering.

The third group of methods relies on the incomplete Cholesky decomposition to construct a low-rank approximation of a kernel matrix (Fine and Scheinberg, 2002; Bach and Jordan, 2005; Kulis et al., 2006). An interesting aspect of the work by Bach and Jordan (2005) and that of Kulis et al. (2006) is the incorporation of side information/labels into the process of finding a good low-rank approximations of a given kernel matrix.

Beside these approaches, an influential ensemble method for low-rank approximation of kernel matrices was proposed by Kumar et al. (2012). This work also contains an empirical study with a number of approaches to landmark selection. Kumar et al. (2012) also note that the landmarks obtained using instance space K-means clustering perform the best among non-ensemble methods.

4.4.2 K-DPP Nyström Method

The first bound on the Nyström approximation with landmarks sampled proportional to submatrix determinants was given by Belabbas and Wolfe (2009). Li et al. (2016) recognize this sampling scheme as a determinantal point process and extend the bound to account for the case when l landmarks are selected to make an approximation of rank $K \leq l$. That bound can be formally specified as (Li et al., 2016)

$$\mathbb{E}\left[\frac{\|H - \tilde{H}_K\|_2}{\|H - H_K\|_2}\right] \leq \frac{l+1}{l+1-K}\sqrt{n-K}\ . \tag{4.17}$$

For $l = K$, the bound can be derived from that of Belabbas and Wolfe (Theorem 1, 2009) by applying the inequalities between the corresponding Schatten p-norms.

The bounds obtained by Belabbas and Wolfe (2009) and Li et al. (2016) can be directly compared to the bound from Corollary 4.9. From Eq. (4.17), for $l = K + 1$, we get that the expected relative approximation error of the K-DPP Nyström method scales like $\mathcal{O}\left(K\sqrt{n-K}\right)$. For a good worst case guarantee on the generalization error of learning with the Nyström approximations (see, e.g., Yang et al., 2012), the parameter K scales as $\sqrt{n}$. Plugging this estimate into Eq. (4.16), we see that the upper bound on the expected error with kernel K-means++ landmarks scales like $\mathcal{O}\left(\sqrt{n}\ln n\right)$ and that with K-DPP landmarks as $\mathcal{O}(n)$.

Having compared our bound to that of the K-DPP landmark selection, we now discuss some specifics of the empirical study performed by Li et al. (2016). The crucial step of that landmark selection strategy is the ability to efficiently sample from a K-DPP. To achieve this, the authors have proposed to use a Markov chain with a worst case mixing time linear in the number of instances. The mixing bound holds provided that a data-dependent parameter satisfies a condition which is computationally difficult to verify (Section 5, Li et al., 2016). Moreover, there are cases when this condition is not satisfied and for which the mixing bound does not hold. In their empirical evaluation of the K-DPP Nyström method, Li et al. (2016) have chosen the initial state of the Markov chain by sampling it using the K-means++ scheme and then run the chain for 100-300 iterations. While the choice of the initial state is not discussed by the authors, one reason that this could be a good choice is because it starts the chain from a high density region. To verify this hypothesis, we simulate the K-DPP Nyström method by choosing the initial state uniformly at random and run the chain for 1 000 and 10 000 steps (Section 4.5). Our empirical results indicate that starting the K-DPP chain with K-means++ samples is instrumental for performing well with this method in terms of runtime and accuracy (Figure 6, Li et al., 2016). Moreover, for the case when the initial state is sampled uniformly at random, our study indicates that the chain might need at least one pass through the data to reach a region with good landmarks. The latter is computationally inefficient already on datasets with more than 10 000 instances.

4.4.3 Instance Space K-means Centroids as Landmarks

We first address the approach to landmark selection based on K-means clustering in the instance space (Zhang et al., 2008) and then give a theoretically sound motivation for why these landmarks work well with the frequently used Gaussian kernel. The outlined reasoning motivates the instance space Lloyd refinements of kernel K-means++ samples and it can be extended to other kernel feature spaces by following the derivations from Burges (1999).

The only existing bound for the instance space K-means landmarks was provided by Zhang et al. (2008). However, this bound only works for kernel functions that satisfy

$$(h(a,b) - h(c,d))^2 \leq \eta(h, \mathcal{X})\left(\|a-c\|^2 - \|b-d\|^2\right)\ ,$$

for all $a, b, c, d \in \mathcal{X}$ and a data- and kernel-dependent constant $\eta(h, \mathcal{X})$. In contrast to this, our bound holds for all positive definite kernels. The bound given by Zhang et al. (2008) is also a worst case bound, while ours is a bound in the expectation. The type of the error itself is also different, as we bound the relative error and Zhang et al. (2008) bound the error in the Frobenius norm. The disadvantage of the latter is in the sensitivity to scaling and such bounds become loose even if a single entry of the matrix is large (Li et al., 2016). Having established the difference in the type of the bounds, it cannot be claimed that one is sharper than the other. However, it is important to note that the bound by Zhang et al. (Proposition 3, 2008) contains the full clustering potential $\phi(C^*)$ multiplied by ${}^{n\sqrt{n}}/_{K}$ as a term and this is significantly larger than the rank dependent term from our bound (e.g., see Theorem 4.8).

Burges (1999) has investigated the geometry of kernel feature spaces and a part of that work refers to the Gaussian kernel. We review the results related to this kernel feature space and give an intuition for why K-means clustering in the instance space provides a good set of landmarks for the Nyström approximation of the Gaussian kernel matrix. The reasoning can be extended to other kernel feature spaces as long as the manifold onto which the data is projected in the kernel feature space is a flat Riemannian manifold with the geodesic distance between the points expressed in terms of the Euclidean distance between instances (e.g., see Riemmannian metric tensors in Burges, 1999).

The frequently used Gaussian kernel is given by

$$h(x, y) = \langle \Phi(x), \Phi(y) \rangle = \exp\left({}^{\|x-y\|^2}/_{2\sigma^2} \right),$$

where the feature map $\Phi(x)$ is infinite dimensional and for a subset X of the instance space $\mathcal{X} \in \mathbb{R}^d$ also infinitely continuously differentiable on X. As in Burges (1999), we denote with $\mathcal{S}$ the image of X in the reproducing kernel Hilbert space of h. The image $\mathcal{S}$ is a $r \leq d$ dimensional surface in this Hilbert space. As noted by Burges (1999), the image $\mathcal{S}$ is a Hausdorff space (Hilbert space is a metric space and, thus, a Hausdorff space) and has a countable basis of open sets (the reproducing kernel Hilbert space of the Gaussian kernel is separable). So, for $\mathcal{S}$ to be a differentiable manifold (Boothby, 1986) the image $\mathcal{S}$ needs to be locally Euclidean of dimension $r \leq d$. We assume that our set of instances X is mapped to a differentiable manifold in the reproducing kernel Hilbert space $\mathcal{H}$. On this manifold a Riemannian metric can be defined and, thus, the set X is mapped to a Riemannian manifold $\mathcal{S}$. Burges (1999) has showed that the Riemannian metric tensor induced by this kernel feature map is approximately $g_{ij} = \frac{\delta_{ij}}{\sigma^2}$, where $\delta_{ij} = 1$ if $i = j$ and zero otherwise ($1 \leq i, j \leq d$). This form of the tensor implies that the manifold is flat.

From the obtained metric tensor, it follows that the squared geodesic distance between two points $\Phi(x)$ and $\Phi(y)$ on $\mathcal{S}$ is equal to the σ-scaled Euclidean distance between x and y in the instance space, i.e., $d_{\mathcal{S}}(\Phi(x), \Phi(y))^2 = {}^{\|x-y\|^2}/_{\sigma^2}$. For a cluster $\mathcal{P}_k$, the geodesic centroid is a point on $\mathcal{S}$ that minimizes the distance to other cluster points (centroid in the K-means sense). For our instance space, we have that

$$c_k^* = \underset{c \in \mathbb{R}^d}{\operatorname{argmin}} \sum_{x \in \mathcal{P}_k} \|x - c\|^2 \quad \Rightarrow \quad c_k^* = \frac{1}{|\mathcal{P}_k|} \sum_{x \in \mathcal{P}_k} x\,.$$

Thus, by doing K-means clustering in the instance space we are performing approximate geodesic clustering on the manifold onto which the data is embedded in the Gaussian kernel feature space. It is important to note here that a centroid from the instance space is only an approximation to the geodesic centroid from the kernel feature space—the preimage of

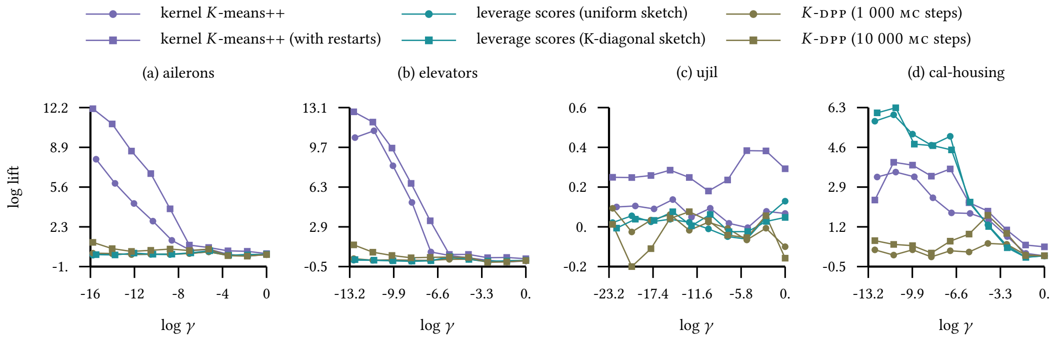


Figure 4.1: The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using that strategy compared to the uniformly sampled ones.

the kernel feature space centroid does not necessarily exist. As the manifold is flat, geodesic centroids are 'good' approximations to kernel K-means centroids. Hence, by selecting centroids obtained using K-means clustering in the instance space we are making a good estimate of the kernel K-means centroids. For the latter centroids, we know that under the conditions of Proposition 4.3 they span the same subspace as the top $(K-1)$ left singular vectors of a finite rank factorization of the kernel matrix and, thus, define a good low-rank approximation of the kernel matrix.

4.5 Experiments

Having reviewed the state-of-the-art methods in selecting landmarks for the Nyström low-rank approximation of kernel matrices, we perform a series of experiments to demonstrate the effectiveness of the proposed approach and substantiate our claims from Sections 4.3 and 4.4. We achieve this by comparing our approach to the state-of-the-art in landmark selection – approximate leverage score sampling (Gittens and Mahoney, 2016) and the K-DPP Nyström method (Belabbas and Wolfe, 2009; Li et al., 2016).

Before we present and discuss our empirical results, we provide a brief summary of the experimental setup. The experiments were performed on 13 real-world datasets available at the UCI and LIACC repositories. Each of the selected datasets consists of more than 5 000 instances. Prior to running the experiments, the datasets were standardized to have zero mean and unit variance. We measure the goodness of a landmark selection strategy with the lift of the approximation error in the Frobenius norm and the time needed to select the landmarks. The lift of the approximation error of a given strategy is computed by dividing the error obtained by sampling landmarks uniformly without replacement (Williams and Seeger, 2001) with the error of the given strategy. In contrast to the empirical study by Li et al. (2016), we do not perform any sub-sampling of the datasets with less than 25 000 instances and compute the Frobenius norm error using full kernel matrices. On one larger dataset with more than 25 000 instances the memory requirements were hindering our parallel implementation and we, therefore, subsampled it to 25 000 instances (*ct-slice* dataset, Section 4.6). By performing our empirical study on full datasets, we are avoiding a potentially negative influence of the sub-sampling on the effectiveness of the compared landmark selection strategies, time consumed, and the accuracy of the approximation error. Following previous empirical studies (Drineas and Mahoney, 2005; Kumar et al., 2012; Li et al., 2016), we evaluate the goodness of landmark selection strategies using the Gaussian kernel and repeat all experiments 10 times to account

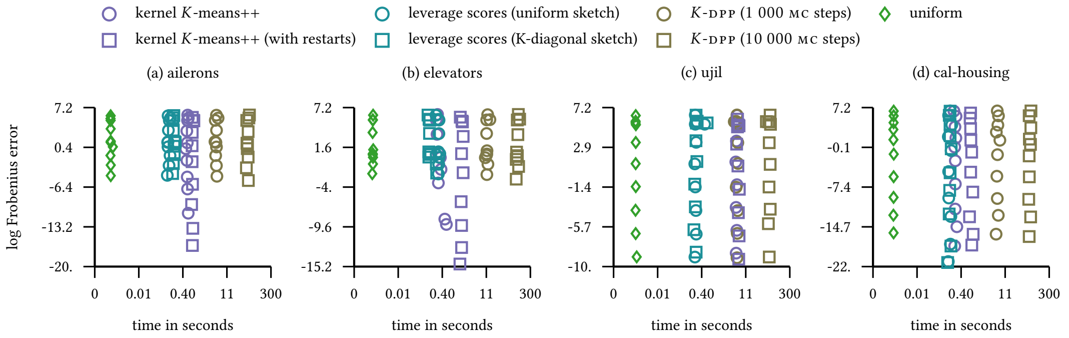


Figure 4.2: The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

for their non-deterministic nature. We refer to $\gamma = 1/\sigma^2$ as the bandwidth of the Gaussian kernel and in order to determine the bandwidth interval we sample 5 000 instances and compute their squared pairwise distances. From these distances we take the inverse of 1 and 99 percentile values as the right and left endpoints. To force the kernel matrix to have a large number of significant spectral components (i.e., the Gaussian kernel matrix approaches to the identity matrix), we require the right bandwidth endpoint to be at least 1. From the logspace of the determined interval we choose 10 evenly spaced values as bandwidth parameters. In the remainder of the section, we summarize our findings with 5 datasets and provide the complete empirical study in Section 4.6.

In the first set of experiments, we fix the approximation rank and evaluate the performance of the landmark selection strategies while varying the bandwidth of the Gaussian kernel. Similar to Kumar et al. (2012), we observe that for most datasets at a standard choice of bandwidth – inverse median squared pairwise distance between instances – the principal part of the spectral mass is concentrated at the top 100 eigenvalues and we set the approximation rank $K = 100$. Figure 4.1 demonstrates the effectiveness of evaluated selection strategies as the bandwidth varies. More precisely, as the log value of the bandwidth parameter approaches to zero the kernel matrix is close to being the identity matrix, thus, hindering low-rank approximations. In contrast to this, as the log-bandwidth value gets smaller the spectrum mass becomes concentrated in a small number of eigenvalues and low-rank approximations are more accurate. Overall, the kernel K-means++ sampling scheme performs the best across all 13 datasets. It is the best performing method on 10 of the considered datasets and a competitive alternative on the remaining ones. The improvement over alternative approaches is especially evident on datasets *ailerons* and *elevators*. The approximate leverage score sampling is on most datasets competitive and achieves a significantly better approximation than alternatives on the dataset *cal-housing*. The approximations for the K-DPP Nyström method with 10 000 MC steps are more accurate than the ones with 1 000 steps. The low lift values for that method seem to indicate that the approach moves rather slowly away from the initial state sampled uniformly at random. This choice of the initial state is the main difference in the experimental setup compared to the study by Li et al. (2016) where the K-DPP chain was initialized with K-means++ sampling scheme.

Figure 4.2 depicts the runtime costs incurred by each of the sampling schemes. It is evident that compared to other methods the cost of running the K-DPP chain with uniformly chosen initial state for more than 1 000 steps results in a huge runtime cost without an appropriate reward in the accuracy. From this figure it is also evident that the approximate leverage score and kernel K-means++ sampling are efficient and run in approximately the same time apart

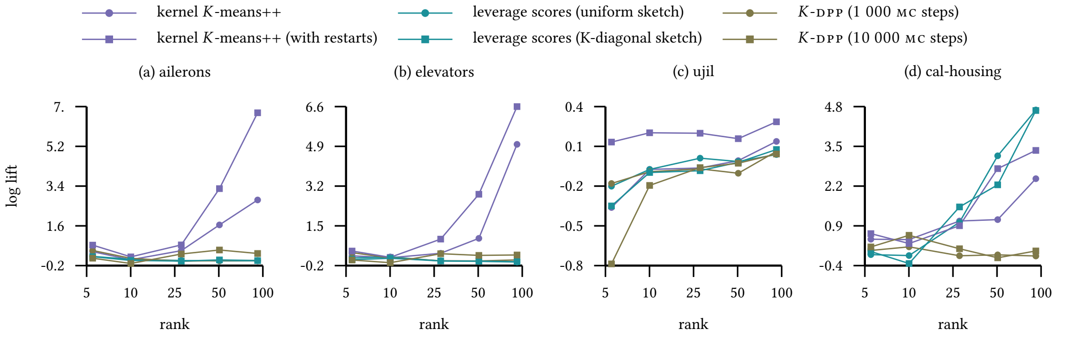


Figure 4.3: The figure shows the improvement in the lift of the approximation error measured in the Frobenius norm that comes as a result of the increase in the rank of the approximation. The bandwidth parameter of the Gaussian kernel is set to the inverse of the squared median pairwise distance between the samples.

from the dataset *ujil* (see also *ct-slice*, Section 4.6). This dataset has more than 500 attributes and it is time consuming for the kernel K-means++ sampling scheme (our implementation does not cache/pre-compute the kernel matrix). While on such large dimensional datasets the kernel K-means++ sampling scheme is not as fast as the approximate leverage score sampling, it is still the best performing landmark selection technique in terms of the accuracy.

In Figure 4.3 we summarize the results of the second experiment where we compare the improvement in the approximation achieved by each of the methods as the rank of the approximation is increased from 5 to 100. The results indicate that the kernel K-means++ sampling achieves the best increase in the lift of the approximation error. On most of the datasets the approximate leverage score sampling is competitive. That method also performs much better than the K-DPP Nyström approach initialized via uniform sampling.

As the landmark subspace captured by our approach depends on the gap between the eigenvalues and that of the approximate leverage score sampling on the size of the sketch matrix, we also evaluate the strategies in a setting where l landmarks are selected in order to make a rank $K < l$ approximation of the kernel matrix. Similar to the first experiment, we fix the rank to $K = 100$ and in addition to the already discussed case with $l = K$ we consider cases with $l = K \ln n$ and $l = K \ln K$. The detailed results of this experiment are provided in Section 4.6 and indicate that there is barely any difference between the lift curves for the kernel K-means++ sampling with $l = K \ln K$ and $l = K \ln n$ landmarks. In their empirical study, Gittens and Mahoney (2016) have observed that for uniformly selected landmarks, $\varepsilon \in [0, 1]$, and $l \in \mathcal{O}(K \ln n)$, the average rank K approximation errors are within $(1 + \varepsilon)$ of the optimal rank K approximation errors. Thus, based on that and our empirical results it seems sufficient to take $K \ln K$ landmarks for an accurate rank K approximation of the kernel matrix. Moreover, the gain in the accuracy for our approach with $l = K \ln K$ landmarks comes with only a slight increase in the time taken to select the landmarks. Across all the datasets, the proposed sampling scheme is the best performing landmark selection technique.

4.6 Appendix: Additional Figures

In this appendix, we provide the detailed results of our empirical study. The appendix is organized such that the results are presented by datasets that are listed in ascending order with respect to the number of instances and dimension. The empirical study provided below compares the following approaches:

i) uniform sampling of landmarks,

ii) approximate leverage score sampling with uniform sketch matrix,

iii) approximate leverage score sampling with the sketch matrix selected by sampling instances proportional to the diagonal entries in the kernel matrix,

iv) K-DPP Nyström with 1 000 and 10 000 MC steps and the initial state chosen by sampling landmarks uniformly at random,

v) K-means clustering in the input space (Lloyd $\oplus$ K-means++),

vi) kernel K-means++ sampling,

vii) kernel K-means++ sampling with local restarts,

viii) kernel K-means++ sampling with local restarts and the Lloyd refinements in the instance space (Lloyd $\oplus$ kernel K-means++).

4.6.1 Parkinsons

The number of instances in this dataset is $n = 5\ 875$ and the dimension is $d = 21$.

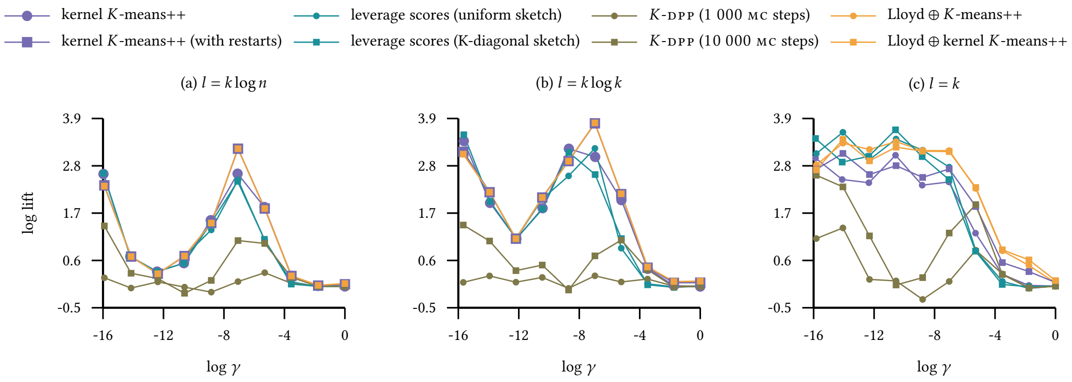


Figure 4.4: Parkinsons Dataset. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.

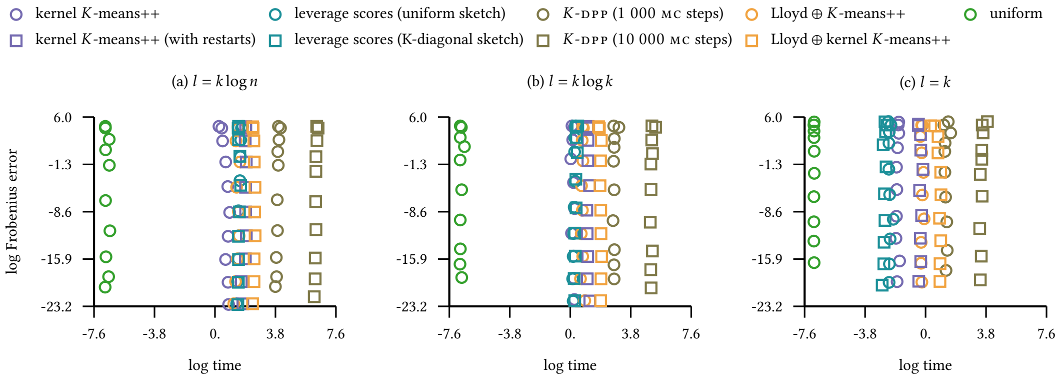


Figure 4.5: Parkinsons Dataset. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

4.6.2 Delta-Ailerons

The number of instances in this dataset is $n = 7\ 129$ and the dimension is $d = 5$.

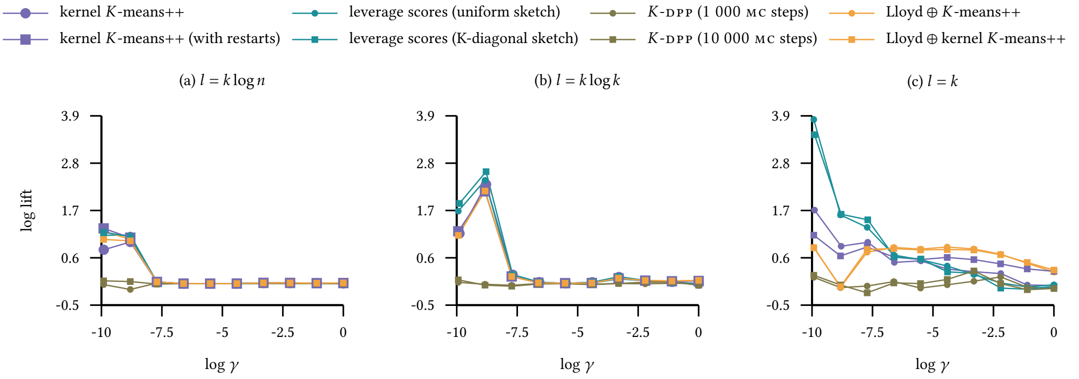


Figure 4.6: Delta-Ailerons Dataset. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.

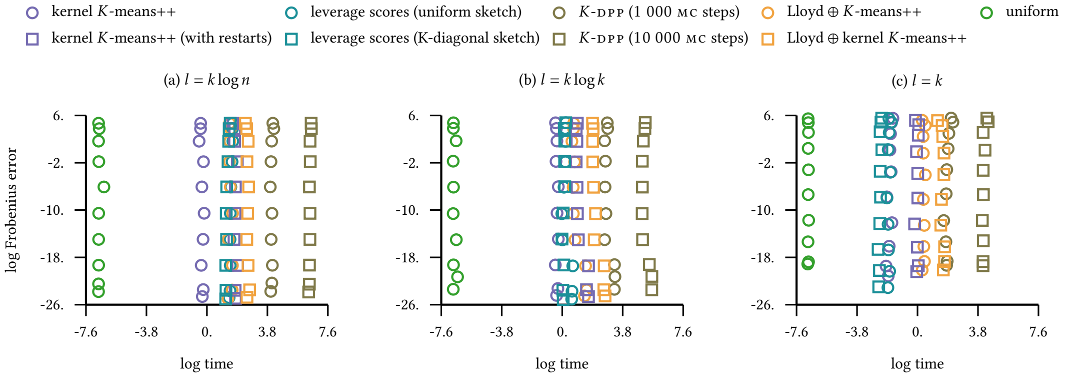


Figure 4.7: Delta-Ailerons Dataset. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

4.6.3 Kinematics

The number of instances in this dataset is $n = 8\,192$ and the dimension is $d = 8$.

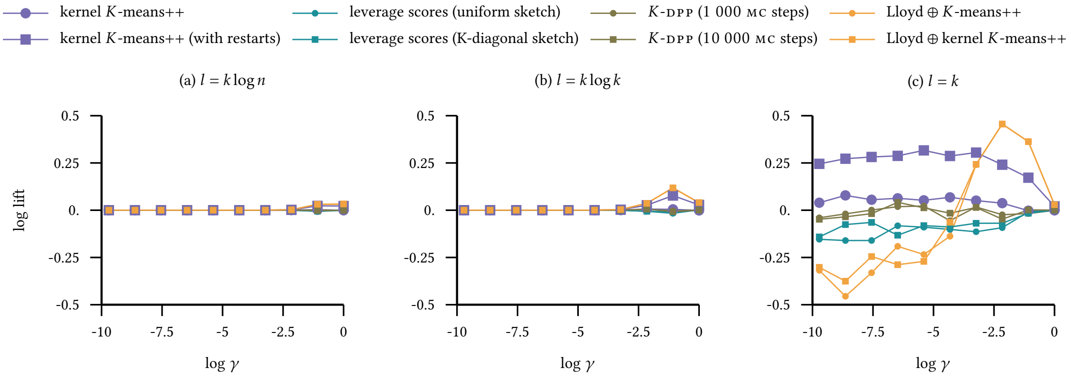


Figure 4.8: Kinematics Dataset. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.

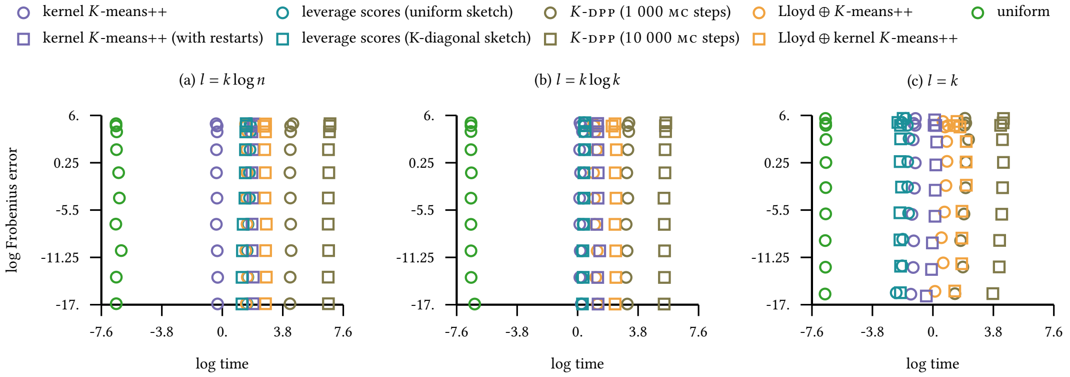


Figure 4.9: Kinematics Dataset. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

4.6.4 CPU-Activity

The number of instances in this dataset is $n = 8\ 192$ and the dimension is $d = 21$.

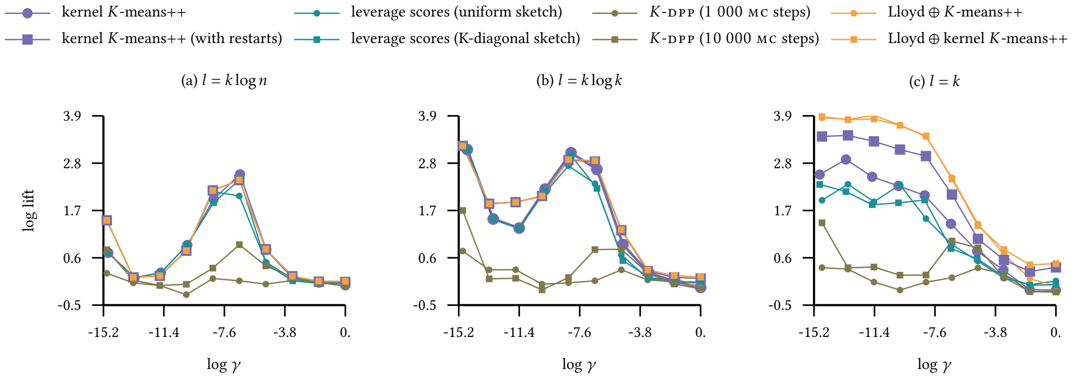


Figure 4.10: CPU-Activity Dataset. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.

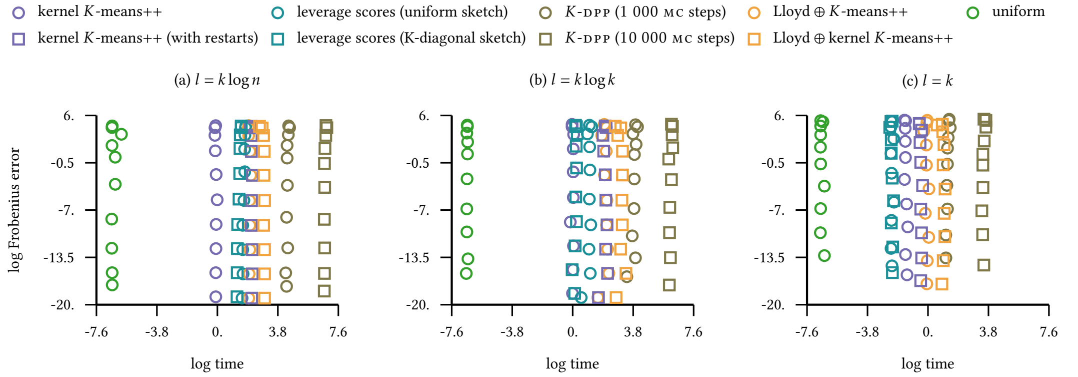


Figure 4.11: CPU-Activity Dataset. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

4.6.5 Bank

The number of instances in this dataset is $n = 8\ 192$ and the dimension is $d = 32$.

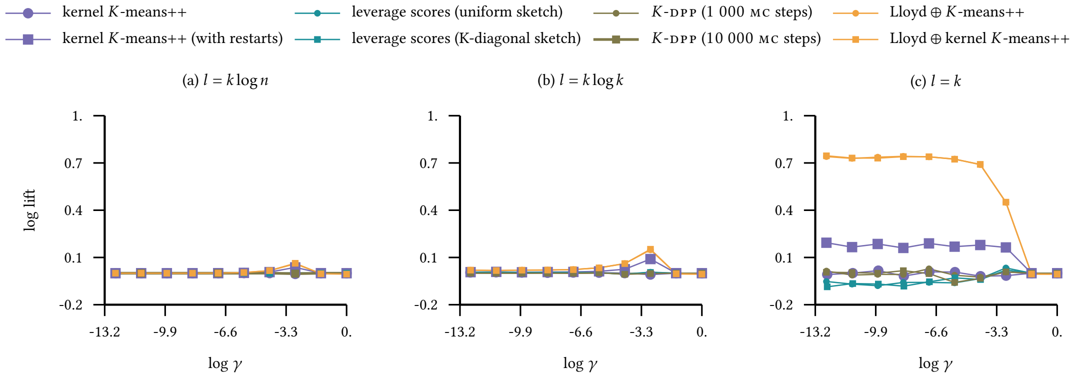


Figure 4.12: Bank Dataset. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.

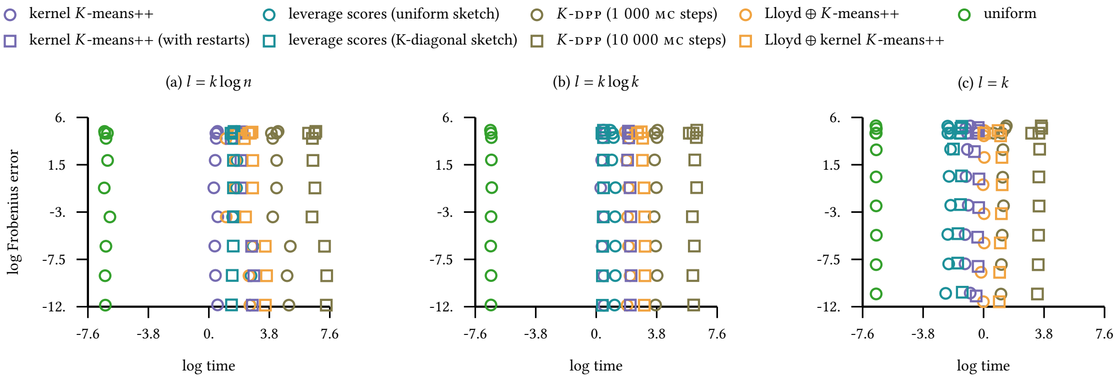


Figure 4.13: Bank Dataset. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

4.6.6 Pumadyn

The number of instances in this dataset is $n = 8\,192$ and the dimension is $d = 32$.

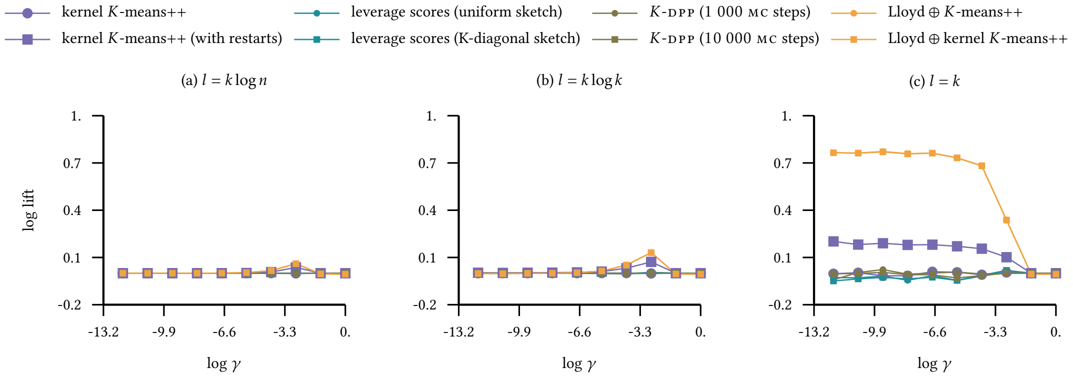


Figure 4.14: Pumadyn Dataset. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.

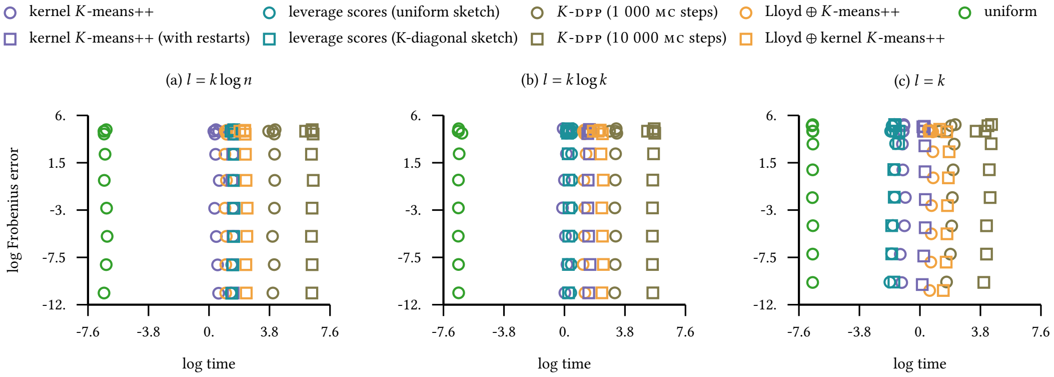
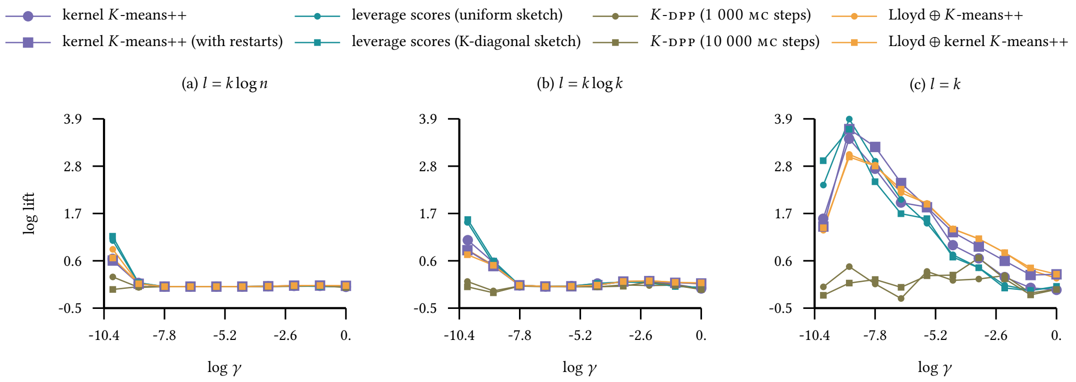


Figure 4.15: Pumadyn Dataset. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

4.6.7 Delta-Elevators

The number of instances in this dataset is $n = 9\ 517$ and the dimension is $d = 6$.

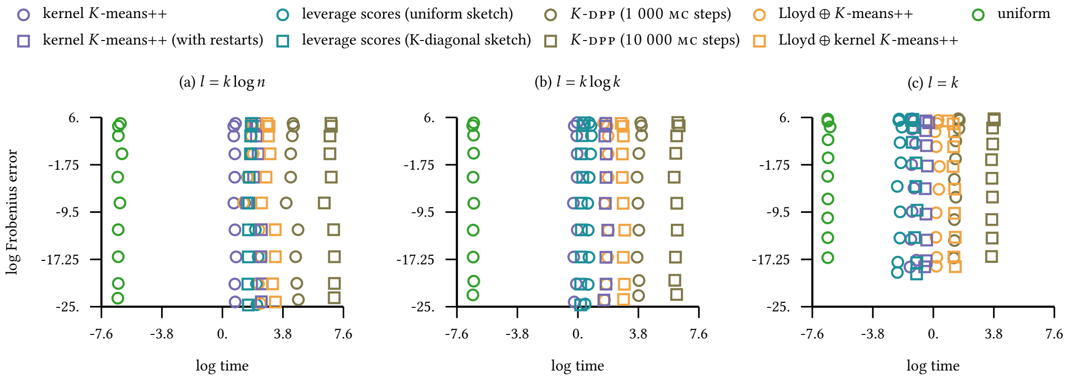


(a) $l = k \log n$

(b) $l = k \log k$

(c) $l = k$

Figure 4.16: Delta-Elevators Dataset. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.



(a) $l = k \log n$

(b) $l = k \log k$

(c) $l = k$

Figure 4.17: Delta-Elevators Dataset. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

4.6.8 Ailerons

The number of instances in this dataset is $n = 13\,750$ and the dimension is $d = 40$.

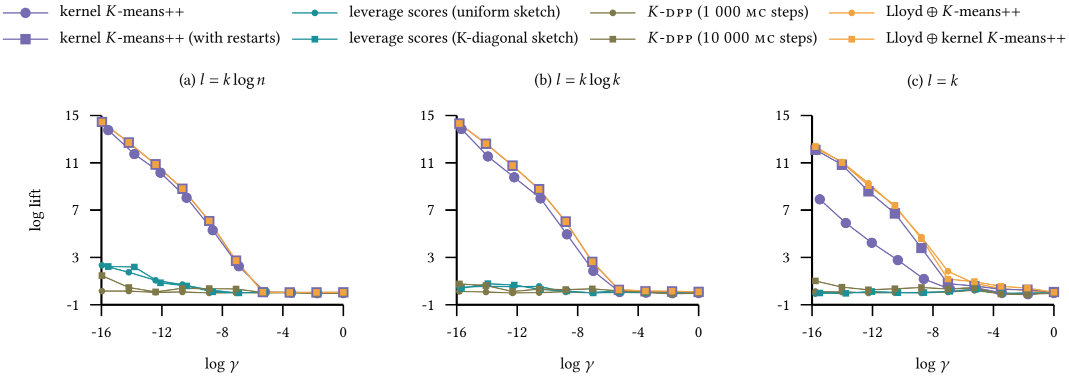


Figure 4.18: AILERONS DATASET. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.

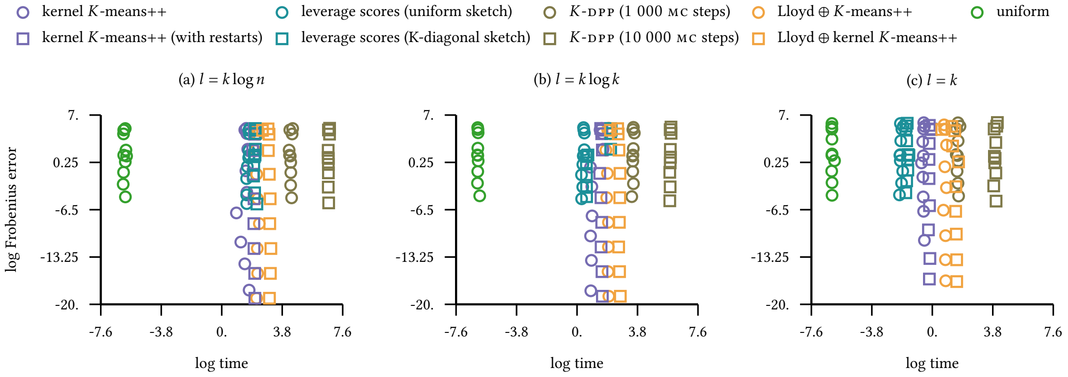


Figure 4.19: AILERONS DATASET. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

4.6.9 Pole-Telecom

The number of instances in this dataset is $n = 15\,000$ and the dimension is $d = 26$.

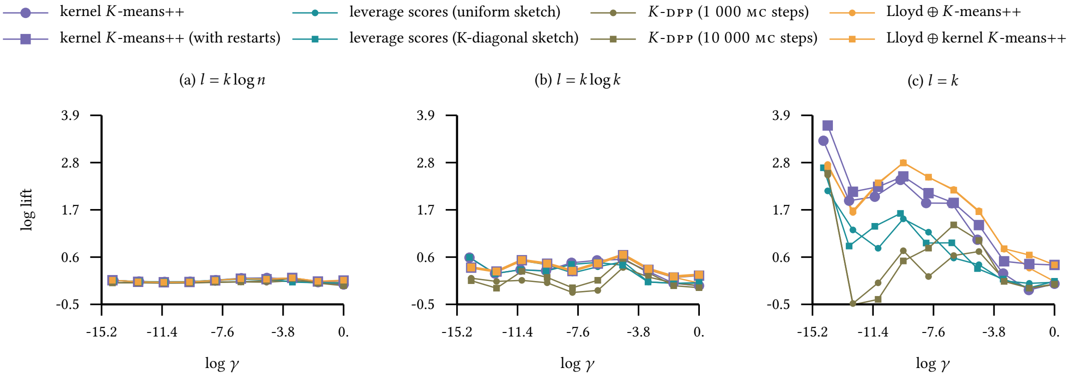


Figure 4.20: Pole-Telecom Dataset. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.

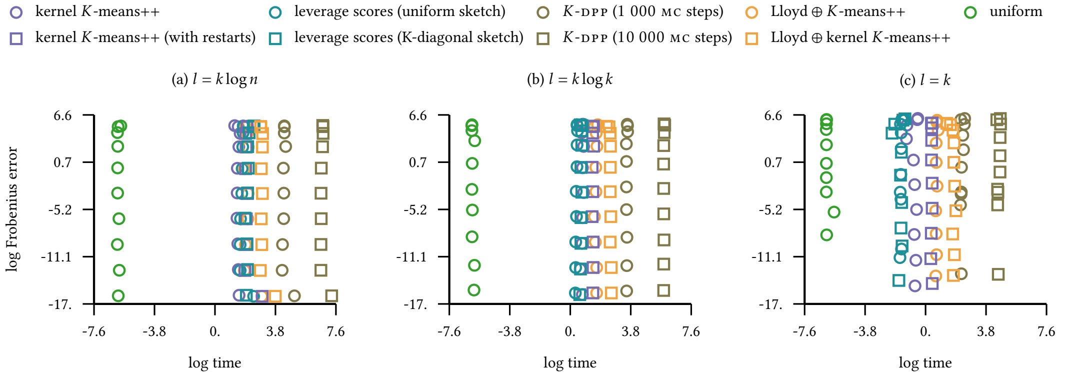


Figure 4.21: Pole-Telecom Dataset. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

4.6.10 Elevators

The number of instances in this dataset is $n = 16\,599$ and the dimension is $d = 18$.

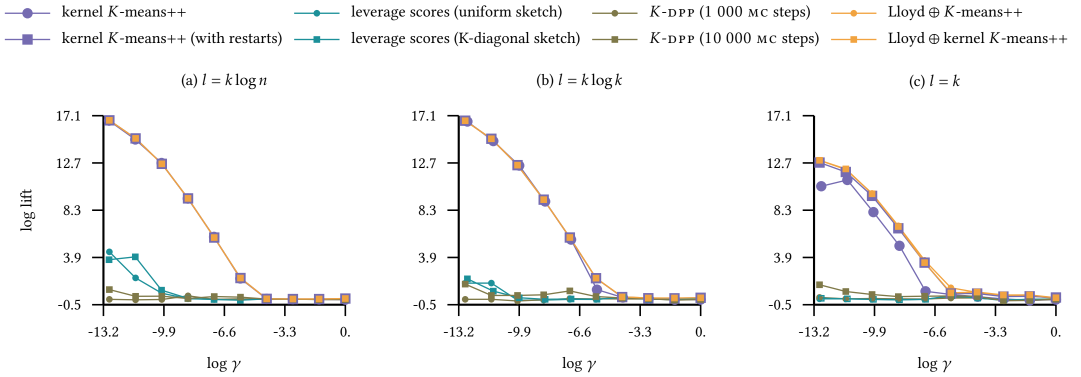


Figure 4.22: Elevators Dataset. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.

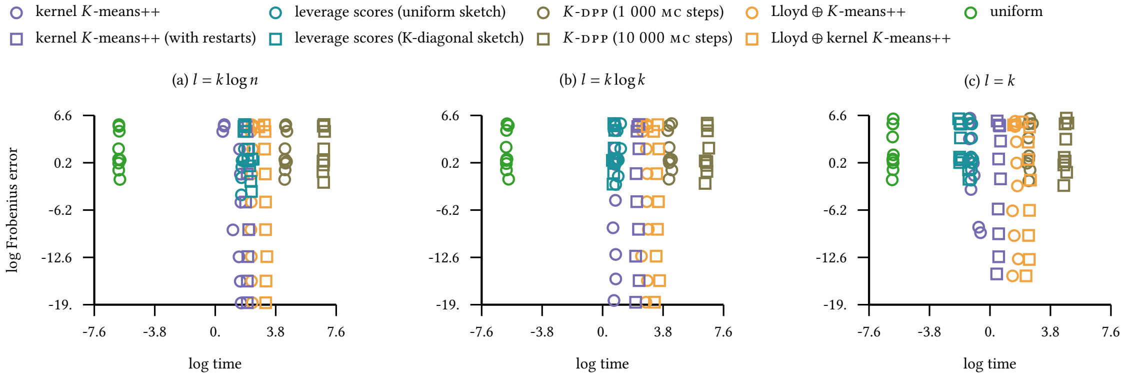
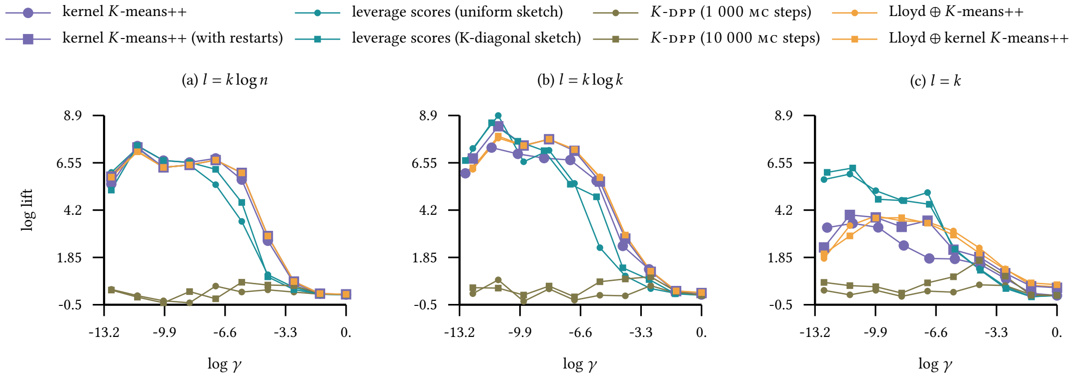


Figure 4.23: Elevators Dataset. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

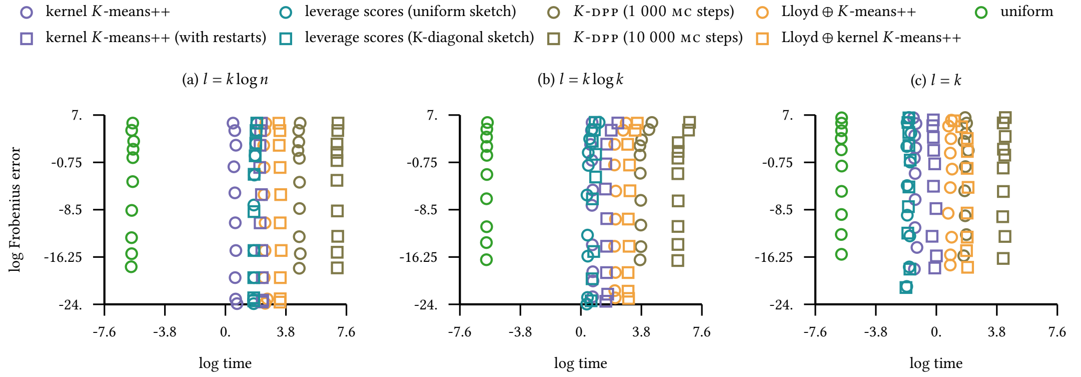
4.6.11 Cal-Housing

The number of instances in this dataset is $n = 20\ 640$ and the dimension is $d = 8$.



(a) $l = k \log n$ (b) $l = k \log k$ (c) $l = k$

Figure 4.24: Cal-Housing Dataset. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.



(a) $l = k \log n$ (b) $l = k \log k$ (c) $l = k$

Figure 4.25: Cal-Housing Dataset. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

4.6.12 UJIL

The number of instances in this dataset is $n = 21\ 048$ and the dimension is $d = 527$.

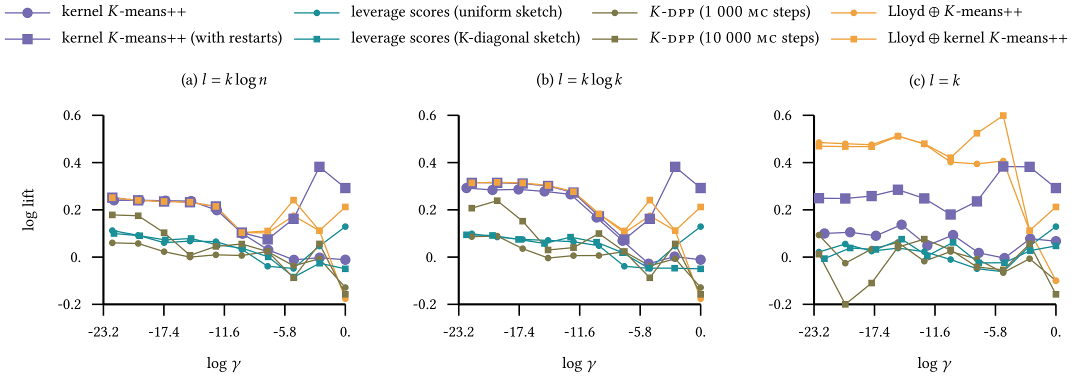


Figure 4.26: UJIL DATASET. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.

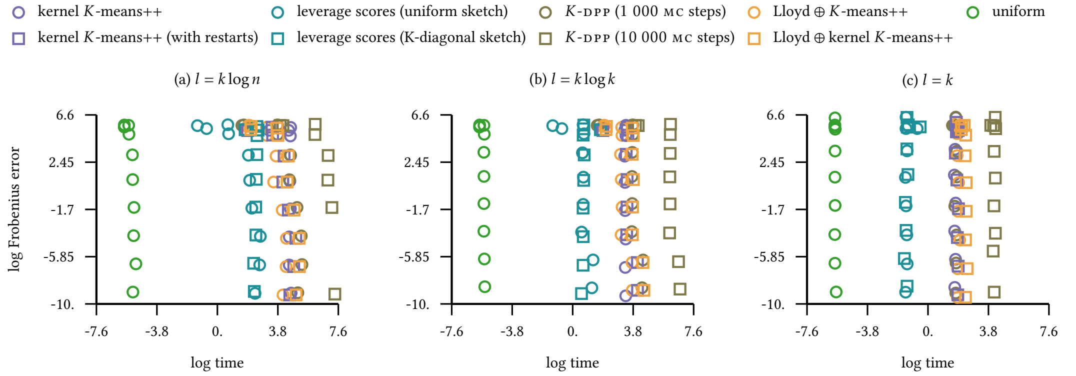


Figure 4.27: UJIL DATASET. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

4.6.13 CT-Slice

The number of instances in this dataset is $n = 53\ 500$ and the dimension is $d = 380$. Due to the memory requirements imposed onto our parallel implementation this dataset was sub-sampled to $n = 25\ 000$ instances.

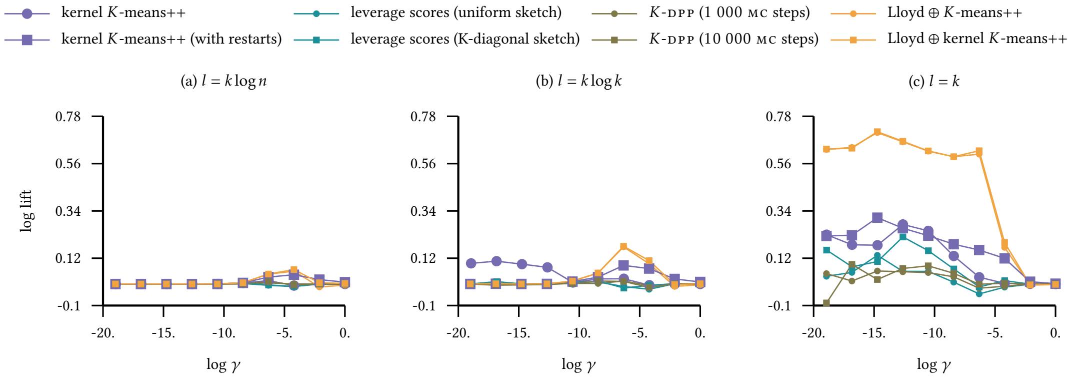


Figure 4.28: CT-Slice Dataset. The figure shows the lift of the approximation error in the Frobenius norm as the bandwidth parameter of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$. The lift of a landmark selection strategy indicates how much better it is to approximate the kernel matrix with landmarks obtained using this strategy compared to the uniformly sampled ones.

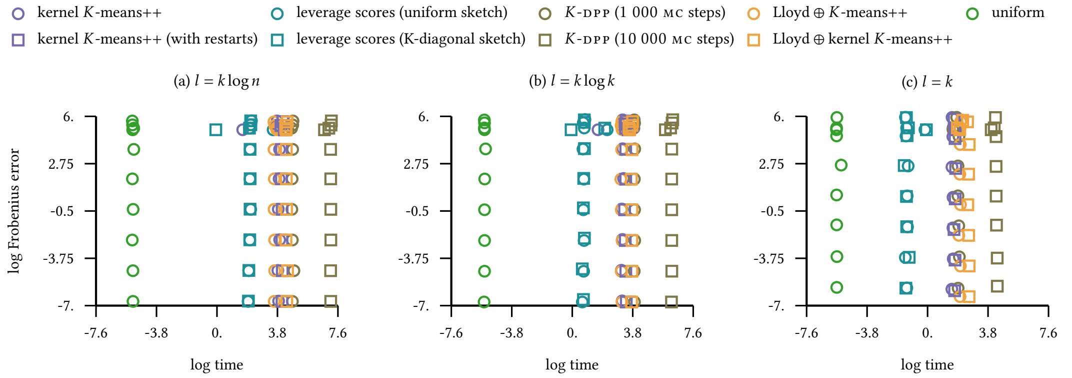


Figure 4.29: CT-Slice Dataset. The figure shows the time it takes to select landmarks via different schemes together with the corresponding error in the Frobenius norm while the bandwidth of the Gaussian kernel varies and the approximation rank is fixed to $K = 100$.

CHAPTER 5

Active Search for Computer-Aided Cyclic Discovery Processes

We consider the class of cyclic discovery processes in which the aim is to discover novel representatives of a fixed but unknown target concept defined over a structured space. Examples where such discovery processes are aided by computer programs include the design of aircrafts (Woltosz, 2012), pharmaceutical drugs (Schneider and Fechner, 2005), and cooking recipes (Varshney et al., 2013). Typically, there are four phases characterizing such a process (Andersson et al., 2009): *design, make, test*, and *analyze.* While in aircraft design, for instance, algorithmic support is mainly by means of simulated *make* and *test* phases, in other discovery processes from this class the underlying systems, such as the biochemistry of human body, are often too complex and not yet well enough understood to merit sufficiently precise simulation. In these cases, the *make* and *test* phases are implemented in a lab (*in vitro* or *in vivo*) and algorithmic support is mainly focused on the *design* and *analysis* phases.

Taking drug discovery as our main motivating example, several problems (Scannell et al., 2012; Schneider and Schneider, 2016) have been identified as the cause for the huge cost associated with attrition (i.e., drug candidates failing later stages of the development process) and increased use of algorithmic support has been proposed as a remedy (Woltosz, 2012). In particular, (i) the *chemspace* (i.e., the space of potentially synthesizable compounds) is huge—estimates are often larger than 10^{60}; (ii) there are many *activity cliffs* (i.e., small changes in structure can have large effects on pharmaceutical activity), and (iii) existing compound libraries focus on a very restricted area of the chemspace. *De novo design* approaches (Schneider and Fechner, 2005) have a potential to overcome these problems by finding desired molecular structures from an intensionally specified chemical space of interest. An intensional description specifies a set of structures with necessary and sufficient conditions for any structure to be in that set. This is in contrast to an extensional definition which simply lists all elements of the set. The intensional specification is often much smaller than the extensional one. The difference in the size of specification is important when considering the runtime and space complexities of algorithms. In particular, extensional definitions of significant parts of the chemspace cannot be stored on disk nor enumerated in feasible time. Sampling from intensionally defined parts, on the other hand, is by no means impossible. In drug discovery, a chemical space of interest is specified only implicitly by the binding affinity to a target protein site or an *in silico* proxy of it. More specifically, a molecule is considered interesting if an expensive to evaluate black-box function (e.g., binding affinity) assigns a

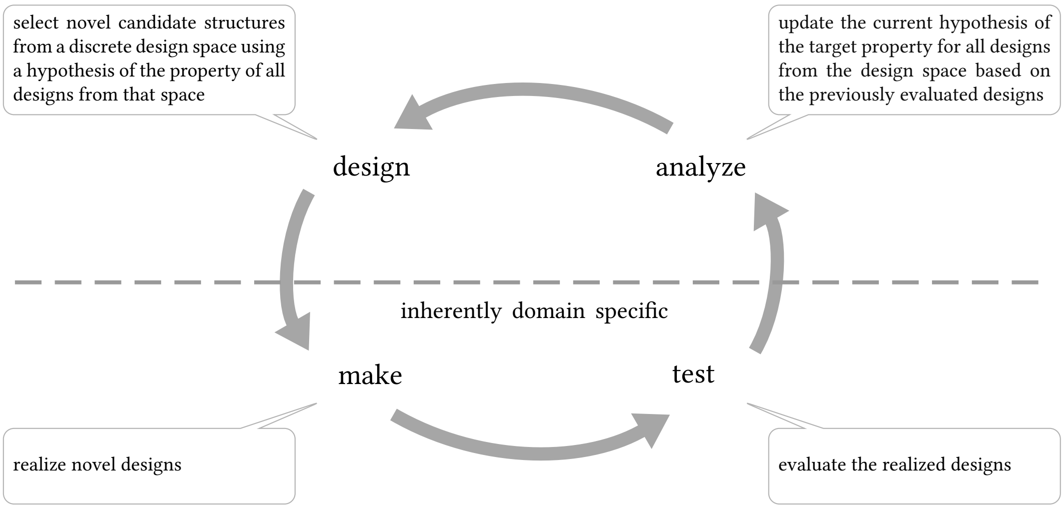

select novel candidate structures from a discrete design space using a hypothesis of the property of all designs from that space
update the current hypothesis of the target property for all designs from the design space based on the previously evaluated designs
design
analyze
inherently domain specific
make
test
realize novel designs
evaluate the realized designs

Figure 5.1: Schematic of a hypothesis-driven cyclic discovery process.

sufficiently high value to it. Faced with such an implicit specification, medicinal chemists hope/expect to devise an intensional description which either covers the whole chemical space of interest or a part of it. Access to any such specification can be provided by a *proposal generator*, which randomly samples compounds from the intensional specification.

In the past 20 years, several Monte Carlo search heuristics have been developed for de novo design of drug-like molecules (Schneider and Fechner, 2005). A common property of these heuristics is the generation of molecular structures using Markov chains. Several search heuristics incorporate an additional scoring step in which the generated structures are accepted/rejected with a probability based on a hand-crafted energy-based scoring function (e.g., see Nishibata and Itai, 1991). The whole process can be seen as Metropolis–Hastings sampling from an expert-designed distribution. Throughout the constructive process this designed distribution is either kept static or manually updated as the process evolves. Motivated by these hand-crafted search heuristics, we propose a data-driven approach that adapts its hypothesis on the target class of structures as it observes the results of new experiments. Figure 5.1 illustrates a hypothesis-driven discovery cycle characteristic to our approach. To deal with the intensionally specified search space, we assume that a proposal generator can be constructed which can be specific to a representation of structures (e.g., a space of graphs, strings, sparse vectors etc.) and has support on (all) parts of the space that contain targets. Similar to the described Monte Carlo search heuristics, we model this proposal generator with a Markov chain given by its transition kernel. As the target structures are typically rare and expensive to evaluate, the cost per discovered structure would be prohibitively high for plain Monte Carlo search performed by evaluating each proposed structure. In particular, proposal generators are typically designed using a small sample of active compounds and are not very good at approximating the binding affinity to a target protein site. Moreover, these samplers are kept static throughout the constructive process and do not exploit the information from the evaluation of the previously designed compounds. To overcome this, our approach relies on a max-entropy conditional model that acts as a probabilistic surrogate for the evaluations in the *test* phase. This conditional model is updated in each iteration after the evaluation of a selected structure and by that the distribution of the Metropolis–Hastings sampler changes in the following *design* step. As a result of this, we cannot assume that the sampled structures are drawn independently from identical distribution. A formal description of the problem setting and a pseudo-code description of our approach are provided in Section 5.1. Following this, we review conditional exponential family models and adapt a result by Altun and Smola (2006) to demonstrate that the maximum a posteriori estimator from this family is a conditional density function which maximizes the conditional entropy subject to constraints on the first moments of the sample (Section 5.2). In Section 5.3, we provide a review of Markov chains and the Metropolis–Hastings algorithm which is used in the design phase of our approach to draw samples from the posterior distribution of structures conditioned on them having the target property. The theoretical properties of the proposed algorithm are analyzed in Section 5.4 where we show the consistency of the approach and bound the mixing time of the Metropolis–Hastings chain with an independent proposal generator. Having presented and analyzed our approach, we describe means to adapt it to different cyclic discovery processes (Section 5.5). In particular, we devise adaptations for a focused drug design problem and discovery of novel cocktails with desired flavors. To study the empirical performance *in silico*, i.e., without conducting lab experiments, we also design synthetic testbeds that share many characteristics with drug design. In particular, instead of the chemspace, we consider the space of all graphs of a given size and aim at constructing graphs with rare and structurally non-smooth properties such as having a Hamiltonian cycle or being connected and planar. We then compare our approach to relevant active learning baselines (described in Section 5.6.1) on these synthetic testbeds and the problem of cocktail discovery. In these experiments, the approach achieves a large structural variety of designed targets compared to the baselines (Section 5.6.2). Following this comparison, we apply a variant of our approach to a focused drug design problem in which as an in silico proxy of the binding affinity we use the molecular docking score to an experimentally determined $\alpha_v\beta_6$ protein structure (Section 5.6.3). The approach again achieves a large structural variety of designed molecular structures for which the docking score is better than the desired threshold. Some novel molecules, suggested to be active by the surrogate model, provoke a significant interest from the perspective of medicinal chemistry and warrant prioritization for synthesis. Moreover, the approach discovered 19 out of the 24 active compounds which are known to be active from previous biological assays (e.g., see Adams et al., 2014). The chapter concludes with a discussion (Section 5.7) where we contrast our method to related approaches.

5.1 Problem Setting and Algorithm

Active learning is broadly defined as the learning setting in which a learning algorithm is allowed to select instances from an instance space and ask for the properties/labels of any of these objects (Cohn et al., 1994; Settles, 2012). The goal of active learning is to generate an accurate hypothesis with as few such queries as possible. This learning setting is different from the standard passive model of supervised learning where the algorithm only receives a set of labeled instances. The typical success measure for active learning is the quality of the found hypotheses. In drug design and several other applications of active learning the goal is, however, to discover objects with desired properties and the algorithms should, therefore, be rewarded for the quality of the discovered objects, rather than the quality of the formed hypotheses. Several extensions of active learning have been developed to address this disparity, active search and active optimization being the most relevant to our work. Active search (Garnett et al., 2012) is a variant of active learning with binary feedback where the goal is to discover as many objects as possible from an unknown property class that can be expensive to evaluate, given a (small) budget of such evaluations. Active optimization (e.g., see Shahriari et al., 2016) focuses on finding a single high-quality item from an instance space rather than a (diverse) set of objects exhibiting a desired property. Active optimization has been investigated extensively in the context of learning with real-valued and binary feedback (Shahriari et al., 2016; Tesch et al., 2013). The real-valued case (Shahriari et al., 2016) can often be cast as global optimization of a black-box function that is expensive to evaluate. The binary case (Tesch et al., 2013) is a variant of active classification where the goal is to discover an item with the highest conditional probability of being a target. Thus, while active search does not discriminate between targets with respect to the corresponding conditional probabilities and focuses on discovering such objects from the whole search space, active optimization with binary feedback focuses on potentially small regions of the design space consisting of objects with high conditional probability of being a target.

We investigate a variant of active search in structured spaces where the goal is to discover as many diverse targets as possible as early as possible, given a fixed budget of potentially expensive black-box property evaluations. In the applications we consider, the search space is specified only intensionally and its cardinality is (at least) exponential in the size of its combinatorial objects (e.g., number of edges in a graph). Thus, the extension of the search space can neither be completely stored on disk nor enumerated in feasible time. The structures we aim to discover can typically be characterized by a target property that is a priori not known for any structure and is expensive to evaluate on each structure. The evaluation process can be noisy and it is simulated with an oracle. The structures exhibiting the target property are typically rare and not concentrated in a small region of the search space. We are thus interested in finding a diverse set of candidates that spans the whole space and is likely to exhibit the target property. Typically, the effectiveness of an active search approach is evaluated using its correct-construction curve and the cumulative number of discovered targets as a function of budget expended (Garnett et al., 2012). More formally, for an instance space $\mathcal{X}$, an evaluation oracle $\mathcal{O}$, a target property $y^* \in \mathcal{Y}$, and a budget of B oracle evaluations the correct-construction curve of a query strategy $q \colon \mathcal{X} \mapsto \{x_1, \ldots, x_B\} \subset \mathcal{X}$ is given by

$$\mathcal{C}(q, y^*) = \left\{ (i, c_i) \mid 1 \leq i \leq B \ \land \ c_i = \left| \left\{ x_j \mid 1 \leq j \leq i \ \land \ y^* = \mathcal{O}\left(x_j\right) \right\} \right| \right\} .$$

While this definition of effectiveness captures the number of successful discoveries made by an active search algorithm within a given budget, it does not reflect the diversity of the discovered items exhibiting a target property. For problems in which the goal is to discover a diverse set of targets, the definition of correct-construction curve can be modified to account for the dispersion of discovered targets. Assuming that oracle queries are not too expensive (e.g., in synthetic testbeds), it is possible to first extract a set of targets $\mathcal{T}$ from a sample of instances from the proposal generator. Then, we can circumscribe a ball of radius κ around each of the extracted targets and define the diversity based correct-construction curve as

$$\mathcal{C}(q, y^*, \mathcal{T}, r) = \left\{ (i, c_i) \mid 1 \leq i \leq B \ \land \ c_i = \left| \left\{ t \in \mathcal{T} \mid 1 \leq j \leq i \ \land \ \left\| x_j - t \right\| < \kappa \right\} \right| \right\} .$$

In preliminary experiments, the average pairwise distance between targets in the observed sample from the proposal generator emerged as an informative choice of the radius κ. In particular, for such a radius the rate of increase in the number of hits $t \in \mathcal{T}$ such that $\left\| x_j - t \right\| < \kappa$ proved to be a good indicator of sample dispersion (see Section 5.6).

Algorithm 5.1 gives a pseudo-code description of our approach. To model the evaluation of the target property, our algorithm takes as input an oracle which outputs a label for a given structure. To reflect the expensiveness of these evaluations, the oracle can be accessed a number of times that is limited by a budget. Other parameters of the algorithm are the proposal generator, target property, and parameters specifying a set of models from the

Algorithm 5.1 DE-NOVO-DESIGN

```
Input: target property $y^* \in \mathcal{Y}$, conditional exponential family model $p\,(y \mid x, \theta)$ with a regularization
    parameter $\lambda > 0$, proposal generator $\mathcal{G}$, evaluation oracle $\mathcal{O}$, and budget $B \in \mathbb{N}$
Output: list of structures $x_1, x_2, \ldots, x_B \in \mathcal{X}^B$
 1: $\theta_1 \leftarrow \mathbf{0}$
 2: for $t = 1, 2, \ldots, B$ do
 3:     $x_t \sim \mathcal{G}$
 4:     repeat
 5:         $x \sim \mathcal{G}$ and $u \sim \mathcal{U}[0,1]$
 6:         if $u < \frac{p(y^* \mid x,\, \theta_t)}{p(y^* \mid x_t,\, \theta_t)}$ then $x_t \leftarrow x$ end if
 7:     until CHAIN MIXED
 8:     $y_t \leftarrow \mathcal{O}(x_t)$ and $w_t \leftarrow {}^1\!/_{p(y^* \mid x_t, \theta_t)}$
 9:     $\theta_{t+1} \leftarrow \operatorname{argmin}_\theta -\frac{1}{t} \sum_{i=1}^{t} w_i \ln p\,(y_i \mid x_i, \theta) + \lambda \|\theta\|_{\mathcal{H}}^2$
10: end for
```

conditional exponential family (reviewed in Section 5.2). In Section 5.2.2, we show that for this choice of a conditional model the probabilistic surrogate for the oracle evaluations is a max-entropy model subject to constraints on the first moments of the sample. Denote the space of candidate structures $\mathcal{X}$, the space of properties $\mathcal{Y}$, and a Hilbert space $\mathcal{H}$ with inner product $\langle \cdot, \cdot \rangle$. The parameter set $\Theta \subseteq \mathcal{H}$ is usually a compact subset of the Hilbert space and together with the sufficient statistics $\phi \colon \mathcal{X} \times \mathcal{Y} \to \mathcal{H}$ of $y \mid x$ specifies the family of conditional exponential models (Altun et al., 2004)

$$p\,(y \mid x, \theta) = \exp\Big(\langle \phi\,(x, y), \theta \rangle - A\,(\theta \mid x)\Big)\,, \tag{5.1}$$

where $A\,(\theta \mid x) = \ln \sum_{y \in \mathcal{Y}} \exp\,(\langle \phi\,(x, y), \theta \rangle)$ is the log-partition function and $\theta \in \Theta$. In practice, we do not directly specify the parameter set Θ but instead simply regularize the importance weighted negative log-likelihood of the sample by adding the term $\|\theta\|_{\mathcal{H}}^2$. To account for this, the algorithm takes as input a hyperparameter which controls the regularization.

The constructive process is initialized by setting the parameter vector of the conditional exponential family to zero (line 1). This implies that the first sample is unbiased and uninformed. Then, the algorithm starts iterating until we deplete the oracle budget B (line 2). In the initial steps of each iteration (lines 3–7), the Metropolis–Hastings algorithm (e.g., see Section 5.3.2 or Metropolis et al., 1953) is used to sample from the posterior

$$p\,(x \mid y^*, \theta_t) = \frac{p\,(y^* \mid x, \theta_t)\,\rho\,(x)}{p\,(y^*)}\,,$$

where $p\,(y^*)$ is the marginal probability of $y^* \in \mathcal{Y}$ and $\rho\,(x)$ is the stationary distribution of the proposal generator $\mathcal{G}$ defined with a transition kernel g satisfying the detailed balance condition (e.g., see Section 5.3 or Andrieu et al., 2003). Thus, to obtain samples from the posterior $p\,(x \mid y^*, \theta_t)$, the Metropolis–Hastings acceptance criterion is

$$\frac{p\,(y^* \mid x', \theta_t)}{p\,(y^* \mid x_t, \theta_t)} \cdot \frac{\rho\,(x') \cdot g\,(x' \to x_t)}{\rho\,(x_t) \cdot g\,(x_t \to x')} = \frac{p\,(y^* \mid x', \theta_t)}{p\,(y^* \mid x_t, \theta_t)}\,, \tag{5.2}$$

where x' is the proposed candidate, x_t is the last accepted state, θ_t is the parameter vector of the conditional exponential family model, and $g\,(x_t \to x')$ denotes the probability that the proposal generator transitions from state x_t to state x'. After the Metropolis–Hastings chain has mixed (line 7), the algorithm outputs its last accepted state x_t as a candidate structure and presents it to an evaluation oracle (line 8). The oracle evaluates it providing feedback y_t to the algorithm. The labeled pair (x_t, y_t) is then added to the training sample and an importance weight is assigned to it (line 8). The importance weighting is needed for the consistency of the algorithm because the samples are neither independent nor identically distributed. Finally, the conditional exponential family model is updated by optimizing the weighted negative-log likelihood of the sample (line 9). This model is then used by the algorithm to sample a candidate structure in the next iteration. The optimization problem in line 9 is convex in θ and the representer theorem (Wahba, 1990) guarantees that it is possible to express the solution θ_{t+1} as a linear combination of sufficient statistics, i.e., $\theta_{t+1} = \sum_{i=1}^{t} \sum_{c \in \mathcal{Y}} \alpha_{ic} \phi(x_i, c)$ for some $\alpha_{ic} \in \mathbb{R}$. Hence, a globally optimal solution can be found and a set of conditional exponential family models can be specified using only a joint input–output kernel and a regularization parameter.

5.2 Conditional Exponential Family Models

In this section, we review conditional exponential family models that act as a probabilistic surrogate for the oracle evaluations in the active search approach presented in Section 5.1. The review follows closely the expositions by Altun et al. (2004) and Altun and Smola (2007; 2006). The conditional exponential family is a family of parameterized conditional density functions which defines a hypothesis class for learning a probabilistic conditional dependence between instances from an instance space $\mathcal{X}$ and properties from a property space $\mathcal{Y}$ using a set of examples $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^{n}$ from the space $\mathcal{X} \times \mathcal{Y}$. We assume that there is an unknown Borel probability measure ρ defined on the space $\mathcal{X} \times \mathcal{Y}$ and that the observed examples are independent samples from ρ. For such a sample of examples we want to estimate the conditional density function of $y \mid x$ such that, for all $(x, y) \in \mathbf{z}$, the expectation of the features of $y \mid x$ with respect to the estimated density closely matches an empirical expectation of the features with respect to samples from ρ. As outlined by Altun and Smola (2006), this estimation problem can be posed as a linearly constrained max-entropy problem and an optimal solution to its dual can be represented as a conditional exponential family model. Consequently, conditional exponential family models are also called conditional max-entropy models. In the remainder of the section, we formally introduce the conditional exponential family of models (Section 5.2.1) and relate the maximum likelihood and maximum a posteriori estimation with these models to the maximization of conditional entropy subject to constraints on the first moments of the sample (Section 5.2.2). This result is a minor adaptation of the work by Altun and Smola (2006). The section concludes with an overview relating conditional exponential family models to Gaussian processes (Section 5.2.3).

5.2.1 Basic Notions

Let $\mathcal{H}$ be a Hilbert space with inner product $\langle \cdot, \cdot \rangle$ and $\phi \colon \mathcal{X} \times \mathcal{Y} \to \mathcal{H}$ a sufficient statistics (i.e., feature vector) of $y \mid x$. A parameter set $\Theta \subseteq \mathcal{H}$ and the sufficient statistics of $y \mid x$ specify a conditional exponential family of models as (Altun et al., 2004)

$$p(y \mid x, \theta) = \exp\Big(\langle \phi(x, y), \theta \rangle - A(\theta \mid x)\Big), \tag{5.3}$$

where $A(\theta \mid x) = \ln \int_{\mathcal{Y}} \exp(\langle \phi(x, y), \theta \rangle) \, dv$ is the log-partition function, v is a base measure on $\mathcal{Y}$ (the counting measure for discrete $\mathcal{Y}$), $\theta \in \Theta$ is a parameter vector, and $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is an example. Beside being the normalization constant of the conditional exponential family model, the log-partition function $A(\theta \mid x)$ also gives the moments of that density function.

Proposition 5.1. *(Jaynes, 1957; Altun et al., 2004) The log-partition function, $A(\theta \mid x)$, is a convex and infinitely continuously differentiable function. Moreover, the derivatives of A satisfy*

$$\nabla_\theta A(\theta \mid x) = \mathbb{E}_{p(y|x,\theta)}[\phi(x,y) \mid x] \quad \wedge \quad \nabla_\theta^2 A(\theta \mid x) = \mathrm{Var}_{p(y|x,\theta)}[\phi(x,y) \mid x] \ . \tag{5.4}$$

For a prior distribution $p(\theta)$ and a conditional exponential family model as the likelihood function, the posterior distribution of the parameter vector θ given an independent and identically distributed sample $\{(x_i, y_i)\}_{i=1}^n$ satisfies

$$p\left(\theta \mid \{(x_i, y_i)\}_{i=1}^n\right) \propto \prod_{i=1}^n p(y_i \mid x_i, \theta) p(\theta) \ .$$

The maximum a posteriori estimator of the parameter θ is then given as

$$\theta^* = \underset{\theta \in \Theta}{\mathrm{argmax}} \ \frac{1}{n} \sum_{i=1}^n \ln p(y_i \mid x_i, \theta) + \ln p(\theta) \ .$$

Having formally introduced conditional exponential family of models, we now proceed to relate the maximum likelihood and maximum a posteriori estimation using these models to the estimation of conditional models via maximization of conditional entropy subject to a linear constraint on the first moment of the sample.

5.2.2 Relation to Conditional Max-Entropy Models

A max-entropy model is an element of a hypothesis class of probability density functions which satisfies a linear constraint on the first moment of the sample with the maximal entropy. The aim of such an estimation procedure is to objectively encode the information from the sample into the model. Jaynes (1957) showed that exponential family models are max-entropy models subject to a linear constraint on the first moment of the sample. This result was extended to conditional exponential family models and generalized to support different ‚*divergence measures*' (Altun and Smola, 2006). We present here an instance of the latter work adapted to the maximization of conditional entropy subject to a constrain on the first moment of the sample. For that, let $\mathcal{P}$ denote the set of all conditional distributions that have square integrable densities with respect to some base measure μ defined on $\mathcal{X} \times \mathcal{Y}$ and support on the entire domain of a sufficient statistics $\phi(x,y)$. More formally, $\mathcal{P} \subset \mathcal{L}_\mu^2(\mathcal{X} \times \mathcal{Y})$ where $\mathcal{L}_\mu^2(\mathcal{X} \times \mathcal{Y})$ is the Hilbert space of square integrable functions defined on $\mathcal{X} \times \mathcal{Y}$. Thus, the inner product between $f, g \in \mathcal{P}$ can be defined as $\langle f, g \rangle_{\mathcal{L}_\mu^2} = \int_{\mathcal{X} \times \mathcal{Y}} f(y \mid x) g(y \mid x) d\mu$. In the remainder of the chapter, we will use $\mathrm{supp}(\cdot)$ to denote the support of a probability distribution and $\mu_\mathcal{X}$ to denote the marginal distribution of a measure μ defined on $\mathcal{X} \times \mathcal{Y}$.

Definition 5.1. *The conditional entropy of a conditional density function $p \in \mathcal{P}$ with respect to a marginal probability density function q defined on* $\mathrm{supp}(\mu_\mathcal{X}) \subseteq \mathcal{X}$ *is defined as*

$$H(p \mid q) = -\int_\mathcal{X} q(x) \int_\mathcal{Y} p(y \mid x) \ln p(y \mid x) d\mu \ . \tag{5.5}$$

Denote the empirical expectation of a feature vector $\phi(x,y)$ with respect to an independent sample $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^n$ from the probability distribution ρ by $\overline{\phi} = \frac{1}{n} \sum_{i=1}^n \phi(x_i, y_i)$.

Then, a max-entropy distribution from $\mathcal{P}$ satisfying a linear constraint on the first moment of the sample is a solution of the following optimization problem

$$\begin{aligned} &\underset{p \in \mathcal{P}}{\operatorname{argmax}} \ H(p \mid \rho_{\mathcal{X}}) \\ &\text{s.t.} \quad \left\| \mathbb{E}_{\rho_{\mathcal{X}}} \mathbb{E}_p [\phi(x,y) \mid x] - \overline{\phi} \right\|_{\mathcal{H}} \leq \varepsilon \, , \end{aligned} \tag{5.6}$$

where $\varepsilon \geq 0$ is a hyperparameter chosen such that there exists a non-empty subset of $\mathcal{P}$ satisfying the linear constraint. The hyperparameter also controls the quality of the matching between the first moment of the sample and that of the conditional probability density function p. As the following proposition will show, if the optimal solution to this problem exists then it can be represented as a conditional exponential family model. If the solution exists for $\varepsilon = 0$, then it is sufficient to find the maximum likelihood estimator from the conditional exponential family. On the other hand, if the optimal solution exists for $\varepsilon > 0$ then it can be found by performing the maximum a posteriori estimation with a zero-mean Laplace prior on the parameter vector of the model. Before formally stating these results, we review the required terminology from convex analysis.

Definition 5.2. *Let $\mathcal{V}$ be a Hilbert space. The convex conjugate of a function $f \colon \mathcal{V} \to \mathbb{R}$ is $f^* \colon \mathcal{V} \to \mathbb{R}$, where f^* is defined as*

$$f^*(\xi) = \sup_{v \in \mathcal{V}} \langle \xi, v \rangle_{\mathcal{V}} - f(v) \, .$$

The effective domain of f is the set $\operatorname{dom}(f) = \{v \in \mathcal{V} \mid f(v) < \infty\}$. The epigraph of f is the set of points above its graph, $\operatorname{epi}(f) = \{(v,r) \in \mathcal{V} \times \mathbb{R} \mid f(v) \leq r\}$. If the epigraph of a function is a convex/closed set (with respect to the Hilbert topology on $\mathcal{V} \times \mathbb{R}$) then the function is called convex/closed. A convex function f is called proper if there exists $v \in \mathcal{V}$ such that $f(v) < \infty$. A proper convex function is closed if and only if it is lower-semicontinuous, i.e., $\liminf_{v' \to v} f(v') \geq f(v)$. The algebraic interior of a set $A \subseteq \mathcal{V}$ is defined as

$$\operatorname{core}(A) = \{a \in A \mid (\forall v \in \mathcal{V})(\exists t_v > 0) : \forall t \in [0, t_v] \ a + tv \in A\} \, .$$

If A is a convex set with non-empty interior then $\operatorname{core}(A) = \operatorname{int}(A)$, where $\operatorname{int}(A)$ is the set of all interior points of A. Having reviewed the relevant terminology from convex analysis, we now state a variant of Fenchel's duality theorem that relates a convex minimization problem to the concave maximization using conjugates. Fenchel's theorem will then be used to relate the optimization problem in Eq. (5.6) to the maximum a posteriori estimation with a conditional exponential family model as the likelihood function. While the following variant of the theorem is for clarity reasons restricted to Hilbert spaces, the general result holds for Banach spaces (e.g., see Rockafellar, 1966).

Theorem 5.2. *(Fenchel's Duality Theorem, Rockafellar, 1966; Altun and Smola, 2006) Let $L \colon \mathcal{V}_1 \to \mathcal{V}_2$ be a bounded linear operator between Hilbert spaces $\mathcal{V}_1$ and $\mathcal{V}_2$. Suppose that $f \colon \mathcal{V}_1 \to \mathbb{R}$ and $g \colon \mathcal{V}_2 \to \mathbb{R}$ are proper convex functions. Denote with*

$$m = \inf_{v \in \mathcal{V}_1} f(v) + g(Lv) \quad \wedge \quad M = \sup_{\xi \in \mathcal{V}_2} -f^*(L^* \xi) - g^*(-\xi) \, .$$

Assume that f, g, and L satisfy one of the following two conditions:

i) $0 \in \operatorname{core}(\operatorname{dom}(g) - L \operatorname{dom}(f))$ *and both, f and g, are lower-semicontinuous,* *ii)* $L\,\mathrm{dom}(f) \cap \mathrm{cont}(g) \neq \emptyset$,

where $\mathrm{cont}(g)$ *denotes the subset of the domain where* g *is continuous. Then, it holds that* $m = M$*, where the dual solution* M *is attainable if it is finite.*

Having reviewed Fenchel's duality theorem, we are now ready to formally state a result relating the max-entropy estimation subject to empirical constrains on the first moments of the sample to the maximum a posteriori estimation using a conditional exponential family model as the likelihood function. The following proposition is a minor adaptation of a result by Altun and Smola (2006) that shows the equivalence of the two estimation problems.

Theorem 5.3. *Suppose there exists a constant* $r > 0$ *such that* $\left\|\phi(x,y)\right\|_{\mathcal{H}} < r$ *for all* $(x,y) \in \mathcal{X} \times \mathcal{Y}$ *and let* $\varepsilon \geq 0$ *be a hyperparameter such that there exists an optimal solution in the interior of* $\mathcal{P}$ *for the optimization problem in Eq. (5.6). Then, we have*

$$\begin{aligned}&\left\{\min_{p \in \mathcal{P}} -H\left(p \mid \rho_{\mathrm{emp}}\right) \textit{ subject to } \left\|\mathbb{E}_{\rho_{\mathrm{emp}}}\mathbb{E}_p\left[\phi(x,y) \mid x\right] - \overline{\phi}\right\|_{\mathcal{H}} \leq \varepsilon\right\} = \\ &\left\{\max_{\theta \in \mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \langle \phi(x_i,y_i), \theta\rangle - \ln \int_{\mathcal{Y}} \exp\left(\langle \phi(x_i,y), \theta\rangle\right) d\nu - \varepsilon \left\|\theta\right\|_{\mathcal{H}}\right\}\end{aligned} \tag{5.7}$$

where $\rho_{\mathrm{emp}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{x=x_i}$ *and* $\mathbb{I}_{x=x_i}$ *is one when* $x = x_i$ *and zero otherwise.*

Proof. The proof follows along the lines of the work by Altun and Smola (2006) with the main difference being the use of conditional entropy instead of the Kullback–Leibler divergence. Beside this, the theorem defines the first moment constraints slightly differently compared to Altun and Smola (2006), where the expectation is not taken over a density function on $\mathcal{X}$.

Denote with $\mathcal{B} = \{h \in \mathcal{H} \mid \|h\|_{\mathcal{H}} \leq 1\}$ the unit ball centered at the origin of $\mathcal{H}$ and let $L_\phi \colon \mathcal{L}^2_\mu(\mathcal{X} \times \mathcal{Y}) \to \mathcal{H}$ with $L_\phi(p) = \mathbb{E}_{\rho_{\mathcal{X}}}\mathbb{E}_p\left[\phi(x,y) \mid x\right]$ be an operator mapping $\mathcal{P}$ to the Hilbert space $\mathcal{H}$. This operator is linear because the expectation operator is linear and it is bounded because $\left\|\phi(x,y)\right\| < r$ for all $(x,y) \in \mathcal{X} \times \mathcal{Y}$. For all $\varepsilon \geq 0$ let

$$\overline{\phi} + \varepsilon\mathcal{B} = \left\{h \in \mathcal{H} \mid h = \overline{\phi} + \varepsilon b \ \wedge \ \|b\|_{\mathcal{H}} \leq 1\right\} .$$

Then, we have that

$$\left\{L_\phi(p) \in \mathcal{H} \mid \left\|L_\phi(p) - \overline{\phi}\right\|_{\mathcal{H}} \leq \varepsilon \ \wedge \ p \in \mathcal{P}\right\} = \left\{L_\phi(p) \in \mathcal{H} \mid L_\phi(p) \in \overline{\phi} + \varepsilon\mathcal{B} \ \wedge \ p \in \mathcal{P}\right\} .$$

Now, in order to shift the hard constraint from the max-entropy optimization problem (see Eq. 5.6) into the objective let us introduce a characteristic function $g \colon \mathcal{H} \to \mathbb{R}$ such that

$$g(h) = \begin{cases} 0 & \text{if } h \in \overline{\phi} + \varepsilon\mathcal{B} \\ \infty & \text{otherwise} . \end{cases}$$

On the one hand, functions $f = -H(\cdot \mid \rho_{\mathcal{X}})$ and g are proper closed convex functions (e.g., see Boyd and Vandenberghe, 2004; Dudík and Schapire, 2006) and, thus, lower-semicontinuous. On the other hand, we have that $\mathrm{dom}(g) = \overline{\phi} + \varepsilon\mathcal{B}$, where $\varepsilon \geq 0$ is chosen so that $0 \in \mathrm{core}\left(\mathrm{dom}(g) - L_\phi(\mathrm{dom}(f))\right)$. Hence, the first condition from Theorem 5.2 is satisfied and we have that the primal and dual solutions are identical.

To complete the proof, we need to derive the convex conjugates for functions f and g. The convex conjugate of function g is given by

$$g^*(\theta) = \sup_{\hat{h}} \left\{\left\langle \theta, \hat{h}\right\rangle_{\mathcal{H}} \mid \hat{h} \in \overline{\phi} + \varepsilon\mathcal{B}\right\} = \left\langle \theta, \overline{\phi}\right\rangle_{\mathcal{H}} + \varepsilon \sup_{\|h\|_{\mathcal{H}} \leq 1} \langle h, \theta\rangle_{\mathcal{H}} = \left\langle \theta, \overline{\phi}\right\rangle_{\mathcal{H}} + \varepsilon \|\theta\|_{\mathcal{H}} .$$

Now, observe that for all $h \in \mathcal{H}$ and all $p \in \mathcal{L}^2_\mu(\mathcal{X} \times \mathcal{Y})$ we have that

$$\left\langle h, L_\phi p \right\rangle_\mathcal{H} = \int_\mathcal{X} \rho(x) \int_\mathcal{Y} \langle h, \phi(x,y) \rangle_\mathcal{H} \, p(y \mid x) \, d\mu = \left\langle L^*_\phi h, p \right\rangle_{\mathcal{L}^2_\mu} ,$$

where $L^*_\phi h = \rho(x) \langle h, \phi(x,y) \rangle_\mathcal{H}$ and L^*_ϕ is the adjoint of the linear operator L_ϕ. Denote with $\hat{f}(\theta) = f\left(L^*_\phi \theta\right)$, where $\theta \in \mathcal{H}$ and $L^*_\phi \theta \in \mathcal{P}$. Then, the convex conjugate of $\hat{f}$ is defined as

$$\hat{f}^*(\theta) = \sup_{p \in \mathcal{P}} \left\langle \theta, L_\phi p \right\rangle_\mathcal{H} - \int_\mathcal{X} \rho(x) \int_\mathcal{Y} p(y \mid x) \ln p(y \mid x) \, d\mu \, .$$

To solve this linearly constrained optimization problem over the Hilbert space $\mathcal{L}^2_\mu(\mathcal{X} \times \mathcal{Y})$ and compute $\hat{f}^*$ in closed form, we first form the Lagrange function

$$\mathcal{L}(p, \lambda) = -\left\langle \theta, L_\phi p \right\rangle_\mathcal{H} + \int_\mathcal{X} \rho(x) \int_\mathcal{Y} p(y \mid x) \ln p(y \mid x) \, d\mu + \int_\mathcal{X} \lambda(x) \int_\mathcal{Y} (p(y \mid x) - 1) \, d\mu \, ,$$

where $\lambda(x) \geq 0$ for all $x \in \operatorname{supp}(\mu_\mathcal{X})$. As the optimal solution to the primal problem exists, we can compute the functional gradient of this Lagrange functional and find its optimal solution by setting the gradient to zero. Before we proceed with this, let us introduce the notion of functional gradient in Hilbert spaces.

For a functional F defined on a Hilbert space and an element p from that space, the functional gradient $\nabla F(p)$ is the principal linear part of a change in F after it is perturbed in the direction of q, i.e.,

$$F(p + \eta q) = F(p) + \eta \langle \nabla F, q \rangle + \left(\eta^2\right) ,$$

where $\eta \to 0$ (e.g., see Secton 3.2 in Gelfand and Fomin, 1963). Applying the definition of functional gradient to $\mathcal{L}$ we obtain that

$$\nabla_p \mathcal{L} = -\rho(x) \Big(\langle \theta, \phi(x,y) \rangle_\mathcal{H} - \ln p(y \mid x) - 1 \Big) + \lambda(x) \, .$$

From here it then follows that

$$\nabla_p \mathcal{L} = 0 \quad \Longrightarrow \quad p^*(y \mid x) = \frac{\exp\Big(\langle \theta, \phi(x,y) \rangle_\mathcal{H} \Big)}{\exp\left({}^{\lambda(x)}/_{\rho(x)} + 1 \right)} \, .$$

Now, taking $\lambda(x) = \rho(x) \Big(\ln \int_\mathcal{Y} \exp\Big(\langle \theta, \phi(x, y') \rangle_\mathcal{H} \Big) d\nu - 1 \Big)$ the constraint $p^* \in \mathcal{P}$ is satisfied and the convex conjugate of $\hat{f}$ is given by

$$\hat{f}^*(\theta) = \mathbb{E}_{\rho_\mathcal{X}} \left[A(\theta \mid x) \right] \, .$$

Plugging the two convex conjugates into the dual problem from Theorem 5.2 and setting $\rho(x) = \rho_{\text{emp}}(x)$ we obtain

$$M = \max_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \langle \phi(x_i, y_i), \theta \rangle_\mathcal{H} - A(\theta \mid x_i) - \varepsilon \|\theta\|_\mathcal{H} \, .$$

□

Theorem 5.3 guarantees that conditional exponential family models are objectively encoding the information from the sample into the model. In fact, any other choice of the conditional model makes additional assumptions about the samples that reduce the entropy and introduces a potentially undesirable bias into the process. As already pointed out, the theorem shows that if a solution exists for $\varepsilon = 0$ then fitting of a max-entropy model is equivalent to the maximum likelihood estimation using conditional exponential family of models. Moreover, if a solution exists for $\varepsilon > 0$ then the max-entropy estimation is equivalent to the maximum a posteriori estimation with a conditional exponential family model as the likelihood function and the Laplace prior on the parameter vector. As the following corollary shows, the latter is equivalent to the maximum a posteriori estimation with the zero-mean Gaussian prior on the parameter vector.

Corollary 5.4. *(Altun and Smola, 2007) Let $\varepsilon > 0$ be a hyperparameter value such that there exists a solution to the problem in Eq. (5.6). Then, there exists $\lambda > 0$ such that*

$$\begin{aligned}&\operatorname*{argmax}_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \langle \phi(x_i, y_i), \theta \rangle_{\mathcal{H}} - A(\theta \mid x_i) - \varepsilon \|\theta\|_{\mathcal{H}} \\ &= \operatorname*{argmax}_{\theta \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \langle \phi(x_i, y_i), \theta \rangle_{\mathcal{H}} - A(\theta \mid x_i) - \lambda \|\theta\|_{\mathcal{H}}^2 \ .\end{aligned}$$

Corollary 5.4 allows us to relate conditional exponential family models and max-entropy estimation to Gaussian processes (Mackay, 1997; Rasmussen and Williams, 2005). The following section reviews this connection first observed by Altun et al. (2004).

5.2.3 Relation to Gaussian Processes

This section reviews the relationship between estimation using Gaussian processes and conditional exponential family models on classification and heteroscedastic regression tasks. We start with an overview of Gaussian processes and then provide sufficient statistics specifying a conditional exponential family of models such that the posterior distribution of the parameter vector given examples is identical for these two estimators. The exposition in this section follows the materials by Altun and Smola (2007) and Rasmussen and Williams (2005).

Gaussian processes are non-parametric Bayesian methods initially developed for regression and later adapted for classification tasks. This class of approaches works by taking a Gaussian prior over a function space and combining that prior with a likelihood function to define a posterior distribution over the space of functions. The inference of the posterior distribution is tractable when the likelihood function is Gaussian. While the latter is true for standard regression tasks with independent and identically distributed Gaussian noise, it does not hold for classification tasks where it is often assumed that the likelihood of observing a label $y \in \mathcal{Y} = \{1, -1\}$ given an instance $x \in \mathcal{X}$ follows the logistic distribution given by

$$\pi(y = 1 \mid x) = \frac{1}{1 + \exp(-\langle \theta, \psi(x) \rangle)} \ ,$$

with θ denoting a parameter vector and ψ a map from $\mathcal{X}$ to a Hilbert space. The latter likelihood function can be expressed using a conditional exponential family model with the sufficient statistics given by $\phi(x, y) = \frac{y\psi(x)}{2}$.

A Gaussian process can be specified with its mean and covariance function. More specifically, denoting a mean function with $m(x)$ and covariance function with $k(x, x')$, the

Gaussian process can be written as

$$f(x) \sim \mathcal{GP}(m(x), k(x,x')) \ .$$

Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ be a space of examples and let $\phi \colon \mathcal{Z} \to \mathcal{H}$ be a mapping from that space to the reproducing kernel Hilbert space given by a positive definite kernel $k(z,z') = \langle \phi(z), \phi(z') \rangle_{\mathcal{H}}$. Denote with $f(z) = \langle \theta, \phi(z) \rangle_{\mathcal{H}}$, where $z \in \mathcal{Z}$ and $\theta \sim \mathcal{N}(0, \Sigma)$. Then,

$$\mathbb{E}_{\theta}[f(z)] = 0 \quad \wedge \quad \mathbb{E}_{\theta}[f(z) f(z')] = \phi(z)^{\top} \Sigma \phi(z') = \hat{k}(z,z') \ .$$

Thus, we have that $f(z) \sim \mathcal{GP}\left(0, \hat{k}(z,z')\right)$ for $z, z' \in \mathcal{Z}$. Denoting the identity matrix with $\mathbb{I}$ and letting $\Sigma = \sigma^2 \mathbb{I}$ we obtain $f(z) \sim \mathcal{GP}\left(0, \sigma^2 k(z,z')\right)$. Taking the latter $\mathcal{GP}$-prior over the dual space $\mathcal{H}^*$ in combination with the sufficient statistics $\phi(x,y) = {}^{y\psi(x)}/_{2}$ and a conditional exponential family model as the likelihood function, the posterior distribution of the $\mathcal{GP}$-classifier satisfies

$$p(\mathbf{f} \mid \mathbf{z}) \propto p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f}) \ , \tag{5.8}$$

where $\mathbf{z} = \{z_i\}_{i=1}^{n}$ with $z_i = (x_i, y_i)$ denoting an independent sample from ρ, $\mathbf{y} = \{y_i\}_{i=1}^{n}$, and $p(\mathbf{f}) \sim \mathcal{N}\left(0, \sigma^2 K\right)$ with K denoting the kernel matrix given by the tuple kernel function $k(z,z')$ for $z, z' \in \{(x_i, y) \mid i = 1, \dots, n \wedge y = \pm 1\}$. The mode of the posterior distribution (5.8) is the maximum a posteriori estimator and for this particular model the estimator is given by

$$\theta^* = \underset{\theta \in \mathcal{H}}{\operatorname{argmin}} \ -\frac{1}{n} \sum_{i=1}^{n} \langle \phi(x_i, y_i), \theta \rangle + \ln \sum_{y \in \mathcal{Y}} \exp(\langle \phi(x_i, y), \theta \rangle) + \frac{\sigma^2}{2} \theta^{\top} K \theta \ . \tag{5.9}$$

Now, from Corollary 5.4 it follows that the maximum a posteriori estimator for the conditional exponential family model with sufficient statistics $\phi(x,y) = {}^{y\psi(x)}/_{2}$ and the Gaussian prior on the parameter vector is equivalent to the maximum a posteriori estimator of the $\mathcal{GP}$-classifier with the logistic likelihood function. While the two classifiers, the conditional exponential family model and the $\mathcal{GP}$-classier, are related they are not equivalent. In particular, the two models differ in the way they predict a label for a given instance. More specifically, the conditional exponential family model, for a given instance $x \in \mathcal{X}$, predicts the label

$$y^* = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \ p(y \mid x, \theta^*) \ ,$$

where $\theta^* \in \mathcal{H}$ defines the maximum a posteriori estimator. In contrast, the $\mathcal{GP}$-classifier with the Laplace approximation of the posterior first defines the distribution of $y \mid x$ as

$$\pi^*(y = 1 \mid x') = \int_{\mathbf{f}'} \pi(y = 1 \mid \mathbf{f}') \int_{\mathbf{f}} p(\mathbf{f}' \mid \mathbf{f}, \mathbf{z}, x') \hat{p}(\mathbf{f} \mid \mathbf{z}) d\mathbf{f} \, d\mathbf{f}' \ .$$

In the latter equation, $\hat{p}(\mathbf{f} \mid \mathbf{z}) = \mathcal{N}(\mathbf{f}^*, H)$ is the Laplace approximation of the posterior distribution $p(\mathbf{f} \mid \mathbf{z})$, H is the Hessian of the posterior at the maximum a posteriori estimator $\mathbf{f}^*$ given by θ^*, and $\pi(y = 1 \mid \mathbf{f}')$ is the logistic likelihood function with

$$\mathbf{f}' = \operatorname{vec}\left(\langle \theta, {}^{\psi(x')}/_{2} \rangle, \langle \theta, {}^{-\psi(x')}/_{2} \rangle\right) \ .$$

The conditional distribution $p(\mathbf{f}' \mid \mathbf{f}, \mathbf{z}, x')$ can be obtained by marginalizing $\mathbf{z}$ from the $\mathcal{GP}$-prior. Thus, the integral with respect to $\mathbf{f}$ can be computed in a closed form because it is a product of two Gaussian distributions. Having computed the relevant terms for the conditional label distribution, the $\mathcal{GP}$-classifier, for a given instance $x \in \mathcal{X}$, predicts the label

$$y^* = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \ \pi^* (y \mid x) \ .$$

For sufficient statistics $\phi(x, y) = \operatorname{vec}\left(\operatorname{vec}\left(y, y^2\right)^\top \psi(x)\right)$, with a derivation similar to that of $\mathcal{GP}$-classification, it is possible to relate the heteroscedastic $\mathcal{GP}$-regression (Le et al., 2005) and the maximum a posteriori estimation using conditional exponential family models.

5.3 Markov Chains and the Metropolis–Hastings Algorithm

The Metropolis–Hastings approach (Metropolis et al., 1953) is a Markov chain Monte Carlo method for the simulation of a probability distribution. The approach is used in Algorithm 5.1 to draw samples from the posterior distribution of structures $p(x \mid y^*, \theta_t)$, conditioned on them having the target property $y^* \in \mathcal{Y}$. This section introduces terminology required for the theoretical analysis of that random process, analyzed in Section 5.4.2. We start by providing a brief overview of Markov chains and the properties characterizing the sensitivity of these random processes to initial conditions such as irreducibility, aperiodicity, ergodicity, and detailed balance condition. Following this, we review the Metropolis–Hastings algorithm that can be used for simulation of any distribution specified up to a normalization constant. The review is based on the surveys of Markov chain Monte Carlo methods by Robert and Casella (2005) and Andrieu et al. (2003).

5.3.1 Markov Chains

For clarity reasons, we restrict our review to Markov chains on finite discrete spaces. A more general introduction to this class of algorithms can be found in Robert and Casella (2005). This restriction is also in line with the problem investigated in this chapter—discovery of combinatorial objects with desired properties. Thus, throughout this section we assume that the state-space $\mathcal{X}$ is a discrete set with finitely many combinatorial objects.

The stochastic process, $\{x_t\}_{t \in \mathbb{N}}$, is called a Markov chain if, for all $t \geq 1$, the conditional distribution of x_t given $x_{t-1}, \ldots, x_0$ is the same as the distribution of x_t given x_{t-1}, i.e.,

$$p(x_t \mid x_{t-1}, \ldots, x_0) = p(x_t \mid x_{t-1}) \ .$$

If the initial state of the chain x_0 is known, then the construction of the chain is completely determined by its transition probabilities, i.e., the conditional density function $p(x_t \mid x_{t-1})$. This density function is also known as the transition kernel of the chain and in the remainder of the chapter we will denote this kernel with $T(x_{t-1} \to x_t) := p(x_t \mid x_{t-1})$.

A σ-finite probability measure π defined on the state-space $\mathcal{X}$ is invariant for a transition kernel $T(\cdot \to \cdot)$ and the corresponding Markov chain if, for all $x' \in \mathcal{X}$, it holds

$$\pi(x') = \sum_{x \in \mathcal{X}} T(x \to x') \pi(x) \ .$$

A Markov chain with an invariant probability measure is stationary in distribution. To see this, observe that $x_0 \sim \pi$ implies $x_t \sim \pi$ for all $t \geq 1$. As a result of this, the invariant probability measure π is also called the *stationary distribution* of the chain. The existence of the stationary distribution is an important stability property of a Markov chain and one of the main reasons for the popularity of Markov chain Monte Carlo methods. More specifically, for distributions that are difficult to simulate (e.g., not analytically tractable) the property enables their simulation via a corresponding Markov chain, subject to additional stability properties such as ergodicity which is discussed subsequently.

Having introduced the notion of a stationary chain, we proceed to review the stability properties of the chain required for the existence of a unique stationary distribution. For that, let $A \subset \mathcal{X}$ and denote the first time step t in which the chain enters the set A with

$$\tau_A = \min\{t \geq 1 \mid t \in \mathbb{N} \, \wedge \, x_t \in A\} \, . \tag{5.10}$$

The time step τ_A is called the *stopping time in* A and $\tau_A = \infty$ if $x_t \notin A$ for all $t \geq 1$. For the set A denote the number of visits of the chain to A with

$$\eta_A = \sum_{t=1}^{\infty} \mathbb{I}_A(x_t) \, . \tag{5.11}$$

This quantity allows us to define the stability property measuring the *expected number of visits to* A given an initial state of the chain $x \in \mathcal{X}$, denoted with $\mathbb{E}[\eta_A \mid x_0 = x]$. This stability measure is needed to ensure that the trajectory of the chain will visit each state often enough. To further formalize this stability property, we need to introduce the notion of state recurrence. A state $x \in \mathcal{X}$ is called *recurrent* if the expected number of returns to x is infinite, i.e., $\mathbb{E}[\eta_x \mid x_0 = x] = \infty$, and *transient* otherwise. Thus, for chains with discrete state-spaces the recurrence property of a state is equivalent to the guarantee of return to that state. In other words, the recurrence of a state can be characterized with the *probability of return to* x *in a finite number of steps* given an initial state of the chain $x \in \mathcal{X}$, denoted with $P(\tau_x < \infty \mid x_0 = x)$. More specifically, a state $x \in \mathcal{X}$ is recurrent if $P(\tau_x < \infty \mid x_0 = x) = 1$. To see that these two definitions are equivalent note that for $P(\tau_x < \infty \mid x_0 = x) > 0$ we have

$$\begin{aligned}
\mathbb{E}[\eta_x \mid x_0 = x] = \quad & \sum_{t=1}^{\infty} P(x_t = x \mid x_0 = x) = \\
& \sum_{t=1}^{\infty} P(\tau_x = t \mid x_0 = x) + \sum_{k=1}^{t-1} P(\tau_x = k \mid x_0 = x) P(x_{t-k} = x \mid x_0 = x) = \\
& P(\tau_x < \infty \mid x_0 = x) + \sum_{t=2}^{\infty} \sum_{k=1}^{t-1} P(\tau_x = k \mid x_0 = x) P(x_{t-k} = x \mid x_0 = x) = \\
& \frac{P(\tau_x < \infty \mid x_0 = x)}{1 - P(\tau_x < \infty \mid x_0 = x)} \, .
\end{aligned}$$

Now, the claim follows by setting $P(\tau_x < \infty \mid x_0 = x) = 1$ or $\mathbb{E}[\eta_x \mid x_0 = x] = \infty$.

Having introduced the notion of a recurrent state, we now turn our attention to a stability property that quantifies the sensitivity of the chain to initial conditions. This property will turn out te be crucial for the existence of the stationary distribution of a chain with discrete state-space. A Markov chain is irreducible if starting from any state it is possible to reach all states from the state-space in a finite number of steps with positive probability. More formally, a chain is *irreducible* if, for all $x, x' \in \mathcal{X}$, it holds that

$$P(\tau_{x'} < \infty \mid x_0 = x) > 0 \, . \tag{5.12}$$

An equivalent definition of the irreducibility requires that the chain satisfies $\mathbb{E}[\eta_{x'} \mid x_0 = x] > 0$ for all $x, x' \in \mathcal{X}$. For a given measure ψ on the state-space $\mathcal{X}$, the Markov chain is ψ-irreducible if, for all $x' \in \mathcal{X}$ with $\psi(x') > 0$ and all $x \in \mathcal{X}$, $P(\tau_{x'} < \infty \mid x_0 = x) > 0$.

A Markov chain is called *recurrent* if there exists a measure ψ on $\mathcal{X}$ such that the chain is ψ-irreducible and if all the states from the support of ψ are recurrent. An irreducible Markov chain on a discrete state-space is guaranteed to have at least one recurrent state (the cardinality of the state-space is finite and there are infinitely many states in the chain). The following proposition establishes the connection between irreducibility and recurrence of a chain on a discrete state-space.

Proposition 5.5. *(Robert and Casella, 2005) An irreducible Markov chain defined on a discrete state-space $\mathcal{X}$ is recurrent.*

Proof. As the chain has at least one recurrent state $x^* \in \mathcal{X}$ we have that $P(\tau_{x^*} < \infty \mid x_0 = x^*) = 1$. Assume now there is a transient state $z \in \mathcal{X}$ with $P(\tau_z < \infty \mid x_0 = z) < 1$. From the irreducibility of the chain we have that there exist $m_1, m_2 \in \mathbb{N}$ such that $P(\tau_{x^*} = m_1 \mid x_0 = z) > 0$ and $P(\tau_z = m_2 \mid x_0 = x^*) > 0$. Thus, we have that it holds

$$P\left(x_{m_1+m_2+n} = z \mid x_0 = z\right) = \sum_{x \in \mathcal{X}} P\left(x_{m_1} = x \mid x_0 = z\right) P\left(x_n = x \mid x_0 = x\right) P\left(x_{m_2} = z \mid x_0 = x\right) \geq$$
$$P\left(x_{m_1} = x^* \mid x_0 = z\right) P\left(x_n = x^* \mid x_0 = x^*\right) P\left(x_{m_2} = z \mid x_0 = x^*\right) .$$

Now, summing the last inequality over $n \in \mathbb{N}$ we deduce

$$\sum_{n=0}^{\infty} P\left(x_{m_1+m_2+n} = z \mid x_0 = z\right) \geq P\left(x_{m_1} = x^* \mid x_0 = z\right) P\left(x_{m_2} = z \mid x_0 = x^*\right) \mathbb{E}\left[\eta_{x^*} \mid x_0 = x^*\right] .$$

As the state x^* is recurrent it must hold that $\mathbb{E}[\eta_{x^*} \mid x_0 = x^*] = \infty$. The latter inequality then implies that $\mathbb{E}[\eta_z \mid x_0 = z] = \infty$. As all the states from the state-space $\mathcal{X}$ are recurrent the chain is also recurrent. □

We can now relate the properties of irreducibility and recurrence of a chain to the existence of the unique stationary probability measure. In particular, as the following theorem will show, for any recurrent chain there exists a unique stationary probability measure. Thus, Proposition 5.5 together with the following theorem implies that an irreducible Markov chain defined on a finite discrete state-space has a unique stationary probability measure.

Theorem 5.6. *(Meyn and Tweedie, 2009; Robert and Casella, 2005) If a Markov chain is recurrent then there exists an invariant σ-finite measure which is unique up to a multiplicative factor.*

An alternative constraint can also be imposed on the transition kernel of a chain to ensure the existence of a stationary probability measure. More specifically, *detailed balance condition*, formally defined below, is a sufficient but not necessary condition for the existence of a unique stationary probability measure of a Markov chain. Moreover, when designing chains the condition is often easier to impose than the recurrence or irreducibility.

Definition 5.3. *A Markov chain with transition kernel $T(\cdot \longrightarrow \cdot)$ satisfies the detailed balance condition if there exists a function π satisfying*

$$T(x \longrightarrow x')\, \pi(x) = T(x' \longrightarrow x)\, \pi(x') . \tag{5.13}$$

The following theorem provides a guarantee that a unique stationary probability density function corresponds to a Markov chain satisfying the detailed balance condition.

Theorem 5.7. *(Robert and Casella, 2005, Theorem 6.46) Suppose that the transition kernel of a Markov chain satisfies the detailed balance condition with a probability density function π. Then, the density function π is the invariant density of the chain.*

Having presented the standard constraints imposed on a chain for the existence of a unique stationary probability measure, we now review the convergence properties of discrete state-space Markov chains. For that, we need to introduce another stability property of the chain ensuring that the chain does not get trapped in cycles as a result of the constraints imposed by the transition kernel. The *period of a state $x \in \mathcal{X}$* is defined as

$$d(x) = \text{GCD}\left(\{m \geq 1 \mid P(x_m = x \mid x_0 = x) > 0\}\right) ,$$

where $\text{GCD}(S)$ denotes the greatest common divisor of a set of positive integers S. A Markov chain is *aperiodic* if $d(x) = 1$ for all $x \in \mathcal{X}$. As we will demonstrate shortly, this is an important stability property for the convergence of the chain.

Let us denote the probability of being at time t in the state x with $p(x_t = x)$ and the corresponding probability distribution over the state-space with a row-vector p_t. A transition kernel defined on a discrete state-space can be represented with a non-negative matrix T such that $T_{ij} = T\left(x_i \to x_j\right)$ for $1 \leq i, j \leq |\mathcal{X}|$. Then, for $n \geq 1$ the chain evolves as

$$p_n = p_{n-1} T = p_0 T^n . \tag{5.14}$$

Now, if the transition matrix T is irreducible (i.e., the corresponding Markov chain is aperiodic) then the Perron–Frobenius theorem (Frobenius, 1912) guarantees the existence of the limit of the matrix power, i.e., $\lim_{n \to \infty} T^n < \infty$. The latter is an important condition for the ergodicity property of the chain, formally defined as follows.

Definition 5.4. *Let $\{x_t\}_{t \in \mathbb{N}}$ be a Markov chain on a discrete state-space $\mathcal{X}$ and let π be the corresponding stationary distribution. The chain is uniformly ergodic if*

$$\lim_{t \to \infty} \sup_{x \in \mathcal{X}} \|P(x_t \mid x_0 = x) - \pi\|_{TV} = 0 ,$$

where $\|\cdot\|_{TV}$ is the total variation norm.

Definition 5.4 introduces a strong notion of convergence for Markov chains. In particular, the uniform ergodicity property implies that the chain is independent of the initial conditions and that a *sample from the chain*, x_t, is asymptotically distributed according to the corresponding stationary distribution. The following theorem formally specifies conditions for the uniform ergodicity of a Markov chain defined on a discrete state-space.

Theorem 5.8. *(Meyn and Tweedie, 2009; Robert and Casella, 2005) For any starting point $x_0 \in \mathcal{X}$, the Markov chain with a transition kernel defined on a discrete state-space $\mathcal{X}$ is uniformly ergodic if the transition kernel is irreducible and aperiodic.*

5.3.2 The Metropolis–Hastings Algorithm

The Metropolis–Hastings algorithm (Metropolis et al., 1953) has been listed as one of the top 10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century (Cipra, 2000; Andrieu et al., 2003). The algorithm belongs to the class of Markov Chain Monte Carlo approaches for the simulation of a probability

Algorithm 5.2 Metropolis–Hastings

```
Input: target density function $\pi$, transition kernel $T$, initial state $x_0$, number of steps $n$
Output: sample $x$ from $\pi$
$x \leftarrow x_0$
for $t = 1, 2, \ldots, n$ do
    $x_t \sim T\left(x_{t-1} \rightarrow \cdot\right)$ and $u \sim \mathcal{U}[0,1]$
    if $u < \min\left\{\frac{\pi(x_t)}{\pi(x_{t-1})} \cdot \frac{T(x_t \rightarrow x_{t-1})}{T(x_{t-1} \rightarrow x_t)}, 1\right\}$ then $x \leftarrow x_t$ end if
end for
```

distribution. The approach developed in this chapter relies on the Metropolis–Hastings algorithm to perform the sampling from a distribution of structures conditioned on their label being equal to that of the target property.

Algorithm 5.2 is a pseudo-code description of the approach. The algorithm takes as input a target density function π specified up to a normalization constant, the proposal generator given by a transition kernel T, an instance x_0 from the state-space $\mathcal{X}$ as the initial state, and the number of Markov chain steps n. In the first step of each iteration, the transition kernel is used to sample a candidate state from the corresponding conditional density function, conditioned on the current state of the chain (i.e., either the initial state which is provided as input to the algorithm or the state visited in the previous iteration). Following this, the chain makes a transition to the sampled candidate state with the acceptance probability $\min\left\{\frac{\pi(x)}{\pi(x_{t-1})} \cdot \frac{T(x \rightarrow x_{t-1})}{T(x_{t-1} \rightarrow x)}, 1\right\}$. The chain iterates for n steps and the last accepted state is returned as an approximate sample from the target density function π. To ensure that the sample indeed follows the target distribution, the number of steps needs to be sufficiently large so that the chain *forgets* the initial state and moves away from the stationary distribution of the proposal generator to the target density function π.

Having described the Metropolis–Hastings algorithm, we proceed to review the theoretical properties of the corresponding chain such as ergodicity and convergence. These properties are mainly determined by the choice of the transition kernel defining a proposal generator. For instance, if there exists a subset $A \subset \mathcal{X}$ such that $\pi(A) > 0$ together with $T(x \rightarrow x') = 0$ for all $x \in \mathcal{X}$ and any $x' \in A$, then the target density π is not the stationary distribution of the Markov chain generated using the Metropolis–Hastings algorithm. The latter can be seen by observing that the chain never visits the set A. Thus, a minimal necessary condition for convergence is that

$$\operatorname{supp}(\pi) \subseteq \cup_{x \in \mathcal{X}} T(x \rightarrow \cdot) \ .$$

Assuming that this condition is satisfied, it can be shown that the transition kernel of the Metropolis–Hastings chain satisfies the detailed balance condition with the density function π. The following proposition is a formal statement of the result.

Proposition 5.9. *(Robert and Casella, 2005, Theorem 7.2) Suppose that T is a transition kernel whose support contains that of a target density function π. Let $\{x_t\}_{t \in \mathbb{N}}$ be a Markov chain generated using the Metropolis–Hastings algorithm with π as the target density function and T as the transition kernel of the proposal generator. The transition kernel of the Metropolis–Hastings chain satisfies the detailed balance condition with the target density function π.*

Proof. Let M be the transition kernel of the Metropolis–Hastings chain. Then, it holds that

$$M(x \rightarrow x') = a(x, x')\, T(x \rightarrow x') + \delta_x(x')\, r(x) \ ,$$

where $a(x, x')$ is the acceptance probability of a transition from state x to x', δ_x is the Dirac mass in x, and $r(x) = \sum_{z \in \mathcal{X}} T(x \to z)(1 - a(x, z))$. The first term in this transition kernel can be transformed as

$$a(x, x') T(x \to x') = \min\left\{T(x \to x'), \frac{\pi(x') T(x' \to x)}{\pi(x)}\right\} = \frac{\pi(x')}{\pi(x)} T(x' \to x) a(x', x) \ .$$

Thus, we have that the detailed balance condition holds for all $x, x' \in \mathcal{X}$, i.e.,

$$\pi(x) M(x \to x') = \pi(x') T(x' \to x) a(x', x) + \pi(x') \delta_{x'}(x) r(x') = \pi(x') M(x' \to x) \ .$$

□

Now, from Theorem 5.8 it follows that the Metropolis–Hastings chain is uniformly ergodic if the transition kernel of the chain is aperiodic and π-irreducible. A sufficient condition for the chain to be aperiodic is that the transition kernel allows events $\{x_{t+1} = x_t\}$ with positive probability. More specifically, the Metropolis–Hastings chain is aperiodic if the acceptance probability, $a(x, x')$, satisfies

$$P\Big(a(x, x') \geq 1\Big) < 1 \ .$$

This condition implies that the transition kernel T corresponding to the proposal generator is not the transition kernel of a Markov chain with the stationary density function π. The latter is reasonable in the sense that if we have a transition kernel that corresponds to a stationary distribution then there is no point in perturbing it with the Metropolis–Hastings algorithm. A sufficient condition for the π-irreducibility of the Metropolis–Hastings chain is that the transition kernel of the proposal generator is positive on the support of π, i.e.,

$$T(x \to x') > 0 \text{ for all } x, x' \in \operatorname{supp}(\pi) \ .$$

Proposition 5.10. *(Robert and Casella, 2005, Theorem 7.4) Suppose that a transition kernel T defined on a discrete state-space is positive on the support of a target density function π. Assume also that π is not the stationary distribution of the Markov chain corresponding to T. Then, the Markov chain generated using the Metropolis–Hastings algorithm with π as the target density function and T as the transition kernel of the proposal generator is uniformly ergodic.*

Of particular interest to the algorithm proposed in this chapter is an instance of the Metropolis–Hastings algorithm where the transition kernel of a proposal generator is independent of the previous states, i.e., $T(x \to x') = T(x')$. This instance of the algorithm is called the *independent Metropolis–Hastings algorithm* and the following theorem provides a sufficient condition for the algorithm to produce a uniformly ergodic Markov chain.

Theorem 5.11. *(Mengersen and Tweedie, 1996; Robert and Casella, 2005) The independent Metropolis–Hastings algorithm produces a uniformly ergodic Markov chain if there exists a constant $c > 1$ such that $\pi(x) < cT(x)$ for all $x \in \operatorname{supp}(\pi)$. In this case, for all $x \in \mathcal{X}$*

$$\|P(x_n \mid x_0 = x) - \pi\|_{TV} \leq 2\left(1 - \frac{1}{c}\right)^n \ ,$$

where $\|\cdot\|_{TV}$ denotes the total variation norm.

Having reviewed the Metropolis–Hastings algorithm, we proceed to investigate the properties of two random processes characteristic to Algorithm 5.1, the consistency of the approach and the Metropolis–Hastings algorithm for drawing samples from $p(x \mid y^*, \theta)$.

5.4 Theoretical Analysis

In this section, we first show that Algorithm 5.1 is consistent and then analyze the mixing time of an independent Metropolis–Hastings chain for sampling from the posterior $p(x \mid y^*, \theta)$. The section concludes with a method for handling large importance weights that can occur in Algorithm 5.1 while performing the weighted maximum a posteriori estimation.

5.4.1 Consistency

Let Θ be a compact subset of a Euclidean space and suppose there exist constants $R, r > 0$ such that $\|\theta\| \leq R$ for all $\theta \in \Theta$ and

$$\left\|\phi(x,y)\right\|_{\mathcal{H}} = \sqrt{k\Big((x,y),(x,y)\Big)} \leq r$$

for all $(x,y) \in \mathcal{X} \times \mathcal{Y}$. In finite dimensional Euclidean spaces closed spheres are compact sets and, in line with our previous assumption, we can take Θ to be the sphere of radius R centered at the origin. In infinite dimensional spaces closed spheres are not compact sets and in this case it is possible to find an approximate finite dimensional basis of the kernel feature space using the Cholesky decomposition of the kernel matrix (Fine and Scheinberg, 2002) and define Θ as in the finite dimensional case. We note that this is a standard step for many kernel based approaches in machine learning (Bach, 2007).

Given the stationary distribution $\rho(x)$ of the proposal generator and the conditional label distribution of the evaluation oracle $\rho(y \mid x)$, the *latent data-generating distribution* is $\rho(x,y) = \rho(y \mid x)\rho(x)$. We use the Kullback–Leibler divergence (Akaike, 1973; White, 1982) to measure the difference between this data-generating distribution and the one given by a conditional exponential family model in place of $\rho(y \mid x)$. Eliminating the parameter-free terms from this divergence measure, we obtain the loss function of θ,

$$L(\theta) = -\int_{\mathcal{X}\times\mathcal{Y}} \ln p(y \mid x, \theta)\, d\rho\ .$$

We assume that there exists a unique minimizer of the loss function $L(\theta)$ in the interior of the parameter set Θ and denote this minimizer with θ^*. If, for all $x \in \mathcal{X}$, the optimal parameter vector $\theta^* \in \Theta$ satisfies

$$\mathbb{E}_{\rho(y|x)}\left[\phi(x,y)\right] = \mathbb{E}_{p(y|x,\theta^*)}\left[\phi(x,y)\right]\ ,$$

it is said that the model is *well-specified.*

In our case, sample points are obtained from a query distribution that *depends* on previous samples, i.e., $x_i \sim q(x \mid x_1, \ldots, x_{i-1})$, but labels are still obtained from the conditional label distribution $y_i \sim \rho(y \mid x_i)$ independent of x_j $(j < i)$. The main difficulty in proving the consistency of the approach in the general case where the queried structures are neither independent nor identically distributed comes from the fact that standard concentration bounds do not hold for this setting. A workaround frequently encountered in the literature is to assume that the model is well-specified as in this case the sampling process is consistent irrespective of the query distribution. Before proving convergence in the general case, we first briefly consider the cases of independent samples and well-specified models.

For the common case in which the training sample is drawn *independently* from a distribution $q(x)$, let

$$\hat{\theta}_n = \underset{\theta\in\Theta}{\operatorname{argmax}} \frac{1}{n}\sum_{i=1}^{n} \frac{\rho(x_i)}{q(x_i)} \ln p(y_i \mid x_i, \theta)\ . \tag{5.15}$$

The sequence of optimizers $\{\hat{\theta}_n\}_{n \in \mathbb{N}}$ converges to the optimal parameter vector θ^* (White, 1982; Shimodaira, 2000). For $q(x) = \rho(x)$, $\hat{\theta}_n$ is the *maximum likelihood* estimate of θ^* over an i.i.d. sample $\{(x_i, y_i)\}_{i=1}^n$. Moreover, for $\Theta = \{\theta \mid \|\theta\| \leq R\}$ the latter optimization problem is equivalent to finding the *maximum a posteriori* estimator with a zero-mean Gaussian prior on the parameter vector θ (e.g., see Section 5.2.2 or Altun et al., 2004).

In the case of a well-specified model, for all $x \in \mathcal{X}$, it holds

$$\mathbb{E}_{\rho(y|x)}[\phi(x,y)] = \mathbb{E}_{p(y|x,\theta^*)}[\phi(x,y)] \ .$$

Thus, for all query distributions $q(x)$, the gradient of the loss is zero at θ^*, i.e.,

$$\nabla L(\theta^*) = \int_{\mathcal{X}} q(x) \int_{\mathcal{Y}} \phi(x,y)(p(y \mid x, \theta^*) - \rho(y \mid x))\, d\mu = 0 \ .$$

In other words, if the model is well-specified, the maximum likelihood estimator is consistent for all query distributions (not necessarily independent from previous samples) and, in particular, for the marginal probability measure $\rho(x)$.

We now proceed to the general case for which we do not make the assumption that the model is well-specified and again show that the optimizer θ_t converges to the optimal parameter vector θ^*. At iteration t of Algorithm 5.1 an instance is selected by sampling from the query distribution $q(x \mid \mathcal{D}_{t-1}) = p(x \mid y^*, \theta_t)$, where θ_t denotes a parameter vector from Θ which is completely determined by the previously seen data $\mathcal{D}_{t-1}$. Thus, a candidate sampled at iteration t depends on previous samples through the parameter vector and the independence between input–output pairs within the sample is lost. As a result of this, the convergence of the sequence $\{\theta_t\}_{t \in \mathbb{N}}$ to θ^* for the general case of misspecified model cannot be guaranteed by the previous results relying on the independence assumption (Shimodaira, 2000). To show the consistency in this general case, we first rewrite the objective which is optimized at iteration t of Algorithm 5.1. For a fixed target property y^*, the parameter vector θ_{t+1} is obtained by solving the following problem

$$\theta_{t+1} = \operatorname*{argmin}_{\theta} \ \frac{1}{t} \sum_{i=1}^{t} \frac{A(\theta \mid x_i) - \langle \phi(x_i, y_i), \theta \rangle}{p(y^* \mid x_i, \theta_i)} + \lambda \|\theta\|^2 \ . \tag{5.16}$$

Assuming the parameter set is well behaved (Theorem 5.12), the objective in Eq. (5.16) is convex and can be optimized using standard optimization techniques. Before we show that the sequence of optimizers θ_t converges to the optimal parameter vector θ^*, let us formally define the empirical loss of a parameter vector θ given the data $\mathcal{D}_t$ available at iteration t,

$$L(\theta \mid \mathcal{D}_t) = \frac{1}{t} \sum_{i=1}^{t} \frac{p(y^*)\Big(A(\theta \mid x_i) - \langle \phi(x_i, y_i), \theta \rangle\Big)}{p(y^* \mid x_i, \theta_i)} \ . \tag{5.17}$$

The following theorem and corollary show that Algorithm 5.1 is consistent in the general case for misspecified models and a sample of structures which are neither independent nor identically distributed.

Theorem 5.12. *Let $p(y \mid x, \theta)$ denote the conditional exponential family distribution parameterized with a vector $\theta \in \Theta$, where Θ is a compact subset of a d dimensional Euclidean space $\mathbb{R}^d$. Let $\rho(x, y)$ denote a latent data generating distribution such that, for all $x \in \mathcal{X}$, the support of the likelihood function $\rho(y \mid x)$ is contained in the support of $p(y \mid x, \theta)$ for all $\theta \in \Theta$. Let $\Big|\ln p(y \mid x, \theta)\Big| \leq h(x,y)$ for all $\theta \in \Theta$ and some function $h(x,y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$*

which is Lebesque integrable in the measure $\rho(x,y)$*. Then for all* $0<\varepsilon,\delta<1$ *there exists* $N(\varepsilon,\delta)\in\Omega\Big(\frac{1}{\varepsilon^2}\left(d\ln\frac{1}{\varepsilon}+\ln\frac{1}{\delta}\right)\Big)$ *such that for all* $t\geq N(\varepsilon,\delta)$ *we have*

$$P\left(\sup_{\theta\in\Theta}|L(\theta)-L(\theta\mid\mathcal{D}_t)|\leq\varepsilon\right)\geq 1-\delta\,.$$

Before we proceed with the proof of Theorem 5.12, let us introduce three auxiliary claims required for our proof and discuss some implications of the assumptions. More specifically, the assumption that the parameter set Θ is a compact subset of a finite dimensional Euclidean space together with $p(y\mid x,\theta)$ being a conditional density function that is bounded away from zero (for all $x\in\mathcal{X}$, for all $y\in\mathcal{Y}$, and for all $\theta\in\Theta$) implies that there exists a constant $p_{\min}>0$ such that $p(y\mid x,\theta)\geq p_{\min}$. Let $\Lambda=\max_{\theta\in\Theta}\lambda_1(\theta)$, where λ_1 denotes the largest eigenvalue of the Hessian matrix of the importance-weighted negative log-likelihood objective function (Eq. 5.16). As Θ is a compact set and the likelihood function is continuous for all $x\in\mathcal{X}$, the eigenvalues of the Hessian matrix are bounded and, thus, there exists a finite maximizer Λ.

Lemma 5.13. *For all* $0<\varepsilon<1$ *and* $\theta_1,\theta_2\in\Theta$ *such that* $\|\theta_1-\theta_2\|<\frac{2p_{\min}\varepsilon}{r+\sqrt{r^2+2\Lambda p_{\min}\varepsilon}}$*, it holds* $|L(\theta_1)-L(\theta_2)|<\varepsilon$ *and* $|L(\theta_1\mid\mathcal{D}_t)-L(\theta_2\mid\mathcal{D}_t)|<\varepsilon$*.*

Proof. Performing the Taylor expansion of the log-likelihood around θ_1 we get

$$\ln p(y\mid x,\theta_2)\leq\ln p(y\mid x,\theta_1)+\mathbb{E}_{y\sim p(y\mid x,\theta_1)}\left[\phi(x,y)^\top(\theta_2-\theta_1)\right]+\frac{\Lambda}{2}\|\theta_2-\theta_1\|^2\;.$$

Now, applying the Cauchy-Schwartz inequality to the right hand-side and using the condition $\|\theta_1-\theta_2\|<\frac{2p_{\min}\varepsilon\Lambda}{r+\sqrt{r^2+2\Lambda p_{\min}\varepsilon}}$ the claim follows, i.e.,

$$\begin{aligned}|L(\theta_1)-L(\theta_2)|&\leq\|\theta_1-\theta_2\|\left(r+\frac{\Lambda}{2}\|\theta_1-\theta_2\|\right)<\varepsilon,\\ |L(\theta_1\mid\mathcal{D}_t)-L(\theta_2\mid\mathcal{D}_t)|&\leq\frac{\|\theta_1-\theta_2\|\left(r+\frac{\Lambda}{2}\|\theta_1-\theta_2\|\right)}{p_{\min}}<\varepsilon\,.\end{aligned}$$

□

Lemma 5.14. *Denote with* $\nu=\frac{p_{\min}\varepsilon}{\left(2r+\sqrt{4r^2+2\Lambda p_{\min}\varepsilon}\right)}$ *and let* $B_1,\ldots,B_{\mathcal{N}(\Theta,\nu)}$ *be a* ν*-cover of the parameter set* Θ*. Then,*

$$P\left(\sup_{\theta\in\Theta}|L(\theta)-L(\theta\mid\mathcal{D}_t)|\leq\varepsilon\right)>1-\mathcal{N}(\Theta,\nu)\sup_{s=1,\ldots,\mathcal{N}(\Theta,\nu)}P\left(|L(\theta_s)-L(\theta_s\mid\mathcal{D}_t)|>\frac{\varepsilon}{2}\right),$$

where θ_s *denotes the center of the ball* B_s*.*

Proof. From the assumptions of the lemma it follows that $\sup_{\theta\in\Theta}|L(\theta)-L(\theta\mid\mathcal{D}_t)|>\varepsilon$ if and only if there exists $1\leq s\leq\mathcal{N}(\Theta,\nu)$ such that $\sup_{\theta\in B_s}|L(\theta)-L(\theta\mid\mathcal{D}_t)|>\varepsilon$. Applying the union bound we get

$$P\left(\sup_{\theta\in\Theta}|L(\theta)-L(\theta\mid\mathcal{D}_t)|>\varepsilon\right)\leq\sum_{s=1}^{\mathcal{N}(\Theta,\nu)}P\left(\sup_{\theta\in B_s}|L(\theta)-L(\theta\mid\mathcal{D}_t)|>\varepsilon\right).\tag{5.18}$$

On the other hand, we have

$$|L(\theta_i) - L(\theta_i \mid \mathcal{D}_t) - L(\theta) + L(\theta \mid \mathcal{D}_t)| < |L(\theta_i) - L(\theta)| + |L(\theta_i \mid \mathcal{D}_t) - L(\theta \mid \mathcal{D}_t)| \ .$$

From the last equation and Lemma 5.13 for θ_i center of B_i and all $\theta \in B_i$ we get

$$|L(\theta) - L(\theta \mid \mathcal{D}_t)| - |L(\theta_i) - L(\theta_i \mid \mathcal{D}_t)| < \frac{\varepsilon}{2} \ .$$

As this holds for all $0 < \varepsilon < 1$ and $\theta \in B_i$ we get that $\sup_{\theta \in B_i} |L(\theta) - L(\theta \mid \mathcal{D}_t)| > \varepsilon$ implies $|L(\theta_i) - L(\theta_i \mid \mathcal{D}_t)| > \frac{\varepsilon}{2}$. From here it follows that

$$P\left(\sup_{\theta \in B_s} |L(\theta) - L(\theta \mid \mathcal{D}_t)| > \varepsilon\right) < P\left(|L(\theta_s) - L(\theta_s \mid \mathcal{D}_t)| > \frac{\varepsilon}{2}\right) \ . \tag{5.19}$$

Combining the results from Eq. (5.18) and (5.19) the claim follows. □

Proposition 5.15. *(Carl and Stephani, 1990) Let $\mathcal{B}$ be a finite dimensional Banach space and let B_R be the ball of radius R centered at the origin. Then, for $d = \dim(\mathcal{B})$, it holds*

$$\mathcal{N}(B_R, \varepsilon, \|\cdot\|) \leq \left(\frac{4R}{\varepsilon}\right)^d \ .$$

Having introduced all the relevant results, we are now ready to prove Theorem 5.12.

Proof of Theorem 5.12. We define all random variables with respect to a probability space $(\Omega, \mathcal{D}, \mathbb{P})$, where Ω is a state space, $\mathcal{D}$ is a σ-algebra of Ω, and $\mathbb{P}$ a probability measure of $\mathcal{D}$. The sampling process is performed using an external source of randomness which we model with an i.i.d. sequence of random variables $\{U_t\}_{t \in \mathbb{N}}$. We fix the filtration $\{\mathcal{D}_t\}_{t \in \mathbb{N}}$ where $\mathcal{D}_t \subset \mathcal{D}$ is the σ-algebra generated by $\{(U_1, \theta_1, x_1, y_1), \ldots, (U_t, \theta_t, x_t, y_t)\}$. The input-output pair (x_{t+1}, y_{t+1}) is measurable with respect to the σ-algebra generated by $(\mathcal{D}_t, U_{t+1})$. In other words, given the history of observations the pair is random only with respect to U_{t+1}.

Having defined our random variables, we proceed with the proof. In a part of the proof we use some of the standard techniques from the theory of martingales and follow the same principle as the proof of the importance weighted active learning (Beygelzimer et al., 2009). In the first step, we show that $\mathbb{E}_{\mathcal{D}_t}[L(\theta \mid \mathcal{D}_t)] = L(\theta)$. In particular, it holds

$$\mathbb{E}[L(\theta \mid \mathcal{D}_t)] = \frac{1}{t} \sum_{i=1}^{t} \int \frac{p(y^*)}{p(y^* \mid x_i, \theta_i)} l(x_i, y_i, \theta) \mathbb{P}(\mathcal{D}_t) =$$

$$\frac{1}{t} \sum_{i=1}^{t} \int \frac{p(y^*)}{p(y^* \mid x_i, \theta_i)} p(x_i \mid y^*, \theta_i) \rho(y_i \mid x_i) l(x_i, y_i, \theta) \underbrace{\int \mathbb{P}(\mathcal{D}_{t-1} \mid x_i, y_i, \theta_i)}_{=1} =$$

$$\frac{1}{t} \sum_{i=1}^{t} \int l(x_i, y_i, \theta) \rho(x_i, y_i) = L(\theta) \ ,$$

where $\ell(x, y, \theta) = A(\theta \mid x) - \langle \phi(x, y), \theta \rangle$.

In the second step of the proof, we bound the discrepancy between the empirical and the expected loss. As there is a dependence within the sample, we cannot rely on the concentration bounds requiring the independence assumption. Therefore, we introduce a sequence for which we prove it is a martingale and then proceed with bounding the discrepancy using a martingale concentration inequality.

Let W_j, $j = 1, \ldots, t$, be a sequence of random variables such that

$$W_j = -w_j \ln p\left(y_j \mid x_j, \theta\right) - L(\theta) \ , \tag{5.20}$$

where $w_j = \frac{p(y^*)}{p(y^* \mid x_j, \theta_j)}$. According to the assumptions, $p(y \mid x, \theta)$ is bounded away from zero for all $x \in \mathcal{X}$, for all $y \in \mathcal{Y}$, and for all $\theta \in \Theta$. Thus, it holds

$$\sup_{\theta \in \Theta, x \in \mathcal{X}, y \in \mathcal{Y}} \left| \ln p(y \mid x, \theta) \right| < -\ln p_{\min} \ .$$

From here it implies that $\left|W_j\right| \leq -\frac{\ln p_{\min}}{p_{\min}} < \infty$ and $\mathbb{E}\left[\left|W_j\right|\right] < \infty$.

We now show that the sequence $Z_t = \sum_{j=0}^{t} W_j$, with $W_0 = 0$, is a martingale. In particular,

$$\mathbb{E}\left[Z_t \mid Z_{t-1}, \ldots, Z_0\right] = Z_{t-1} + \mathbb{E}_{x_t, y_t \mid \mathcal{D}_{t-1}}\left[w_t l\left(x_t, y_t, \theta\right)\right] - L(\theta) = Z_{t-1} \ .$$

On the other hand, it holds $|Z_t - Z_{t-1}| = |W_t| \leq -\frac{\ln p_{\min}}{p_{\min}}$. From here using the inequality for martingales by Azuma (1967) we deduce

$$P\left(|L(\theta \mid \mathcal{D}_t) - L(\theta)| > \frac{\varepsilon}{2}\right) = P\left(|Z_t| > \frac{t\varepsilon}{2}\right) < 2\exp\left(-\frac{t\varepsilon^2 p_{\min}^2}{8\left(\ln p_{\min}\right)^2}\right) . \tag{5.21}$$

As this holds for all $\theta \in \Theta$, applying Lemma 5.14 for $\nu = \frac{p_{\min}\varepsilon}{2r + \sqrt{4r^2 + 2\Lambda p_{\min}\varepsilon}}$ we get

$$P\left(\sup_{\theta \in \Theta} |L(\theta) - L(\theta \mid \mathcal{D}_t)| > \varepsilon\right) < 2\mathcal{N}\left(\Theta, \frac{p_{\min}\varepsilon}{2r + \sqrt{4r^2 + 2\Lambda p_{\min}\varepsilon}}\right) \exp\left(-\frac{t\varepsilon^2 p_{\min}^2}{8\left(\ln p_{\min}\right)^2}\right) .$$

From the last equation and Proposition 5.15 we get

$$\ln\frac{\delta}{2} \geq d \ln \frac{4R\left(2r + \sqrt{4r^2 + 2\Lambda p_{\min}\varepsilon}\right)}{p_{\min}\varepsilon} - \frac{t\varepsilon^2 p_{\min}^2}{8\left(\ln p_{\min}\right)^2} \implies$$

$$t\left(\frac{p_{\min}}{\ln p_{\min}}\right)^2 \varepsilon^2 \in \Omega\left(d \ln\frac{1}{\varepsilon} + \ln\frac{1}{\delta}\right) \implies t \in \Omega\left(\left(\frac{\ln p_{\min}}{p_{\min}}\right)^2 \frac{1}{\varepsilon^2}\left(d \ln\frac{1}{\varepsilon} + \ln\frac{1}{\delta}\right)\right) .$$

Hence, we have shown that there exists a positive integer $N \in \Omega\left(\frac{1}{\varepsilon^2}\left(d \ln\frac{1}{\varepsilon} + \ln\frac{1}{\delta}\right)\right)$ such that for all $0 < \varepsilon, \delta < 1$, and all $t > N$ the claim holds. □

Corollary 5.16. *The sequence of estimators* $\{\theta_t\}_{t \geq 1}$ *converges in probability to* $\theta^* \in \Theta$.

Proof. First note that from the compactness of Θ, it follows that the Hessian of the negative log-likelihood is strictly positive definite and, therefore, there exist unique minimizers of the loss functions $L(\theta)$ and $L(\theta \mid \mathcal{D}_t)$. From Theorem 5.12, we have that for sufficiently large t with probability $1 - \delta$ it holds that $L(\theta^* \mid \mathcal{D}_t) \leq L(\theta^*) + \varepsilon$ and $L(\theta_t) \leq L(\theta_t \mid \mathcal{D}_t) + \varepsilon$. From the strict convexity of the optimization objective $L(\cdot \mid \mathcal{D}_t)$ it follows that $L(\theta_t \mid \mathcal{D}_t) \leq L(\theta^* \mid \mathcal{D}_t)$. Hence, with probability $1 - \delta$

$$L(\theta_t) - L(\theta^*) \leq |L(\theta_t) - L(\theta_t \mid \mathcal{D}_t)| + L(\theta_t \mid \mathcal{D}_t) - L(\theta^* \mid \mathcal{D}_t) + |L(\theta^* \mid \mathcal{D}_t) - L(\theta^*)| \leq 2\varepsilon.$$

From here it follows that the sequence of estimators $\{\theta_t\}_{t \geq 0}$ converges in probability to the optimal parameter θ^*. □

5.4.2 Mixing Time Analysis

Having shown the consistency of Algorithm 5.1, we proceed to bound the mixing time of the Metropolis–Hastings chain. For that, we consider an independent proposal generator $\mathcal{G}$ and provide a simple *coupling* analysis (Vembu et al., 2009) to bound the worst case mixing time of an independent Metropolis–Hastings chain for sampling from the posterior $p(x \mid y^*, \theta_t)$. We start by formally defining the coupling of two random processes and then provide a result by Aldous (1983) that relates the coupling and worst case mixing time of a Markov chain.

Definition 5.5. *Let $\mathcal{M}$ be a finite, ergodic Markov chain defined on a state space Ω with transition probabilities $p(x \longrightarrow x')$. A* coupling *is a joint process $(\mathcal{A}, \mathcal{B}) = (A_t, B_t)$ on $\Omega \times \Omega$ such that each of processes $\mathcal{A}$, $\mathcal{B}$, considered marginally, is a faithful copy of $\mathcal{M}$.*

The following result by Aldous (1983) allows us to utilize perfect sampling algorithms such as coupling from the past (Propp and Wilson, 1996) to draw samples from the posterior. In particular, suppose $|\mathcal{X}|$ parallel and identical chains are started from all possible states $x \in \mathcal{X}$ and an identical random bit sequence is used to simulate all the chains. Thus, whenever two chains move to a common state, all the future transitions of the two chains are the same. From that point on it is sufficient to track only one of the chains. This is called a *coalescence* (Huber, 1998). Propp and Wilson (1996) have shown that if all the chains were started at time $-\mathcal{T}$ and have coalesced to a single chain at step $-T$ with $\mathcal{T} > T > 0$, then samples drawn at time 0 are exact samples from the stationary distribution. The following lemma embodies this principle and it is crucial for our bound on the worst case mixing time for sampling from the posterior distribution of structures using an independent Metropolis–Hastings chain.

Lemma 5.17. *(Aldous, 1983) Let $\mathcal{M}$ be a finite, ergodic Markov chain, and let (A_t, B_t) be a coupling for $\mathcal{M}$. Suppose that $P(A_{t(\varepsilon)} \neq B_{t(\varepsilon)}) \leq \varepsilon$, uniformly over the choice of initial state (A_0, B_0). Then the mixing time $\tau(\varepsilon)$ of $\mathcal{M}$ (starting at any state) is bounded from above by $t(\varepsilon)$.*

The following proposition gives a worst case bound on the mixing time of an independent Metropolis–Hastings chain for sampling from the posterior distribution $p(x \mid y^*, \theta_t)$.

Proposition 5.18. *For all $0 < \varepsilon < 1$, the mixing time $\tau(\varepsilon)$ of an independent Metropolis–Hastings chain for sampling from the posterior distribution $p(x \mid y^*, \theta_t)$ is bounded from above by $\left\lceil \ln \varepsilon / \ln\left(1 - \exp(-4r \|\theta_t\|)\right)\right\rceil$.*

Proof. As $\min_{x \in \mathcal{X}} p\left(y^* \mid x, \theta_t\right) \leq \max_{x \in \mathcal{X}} p\left(y^* \mid x, \theta_t\right)$, the lower bound on the Metropolis–Hastings acceptance criterion is never greater than 1. Then, from Eq. (5.2) and (5.1) it follows that, for a finite space $\mathcal{Y}$, the transition probability from a state x to a state x' satisfies

$$p(x \longrightarrow x') \geq \frac{\exp\left(\left\langle \phi(x', y^*), \theta_t \right\rangle - A(\theta_t \mid x')\right)}{\exp\left(\left\langle \phi(x, y^*), \theta_t \right\rangle - A(\theta_t \mid x)\right)} = \frac{\sum_{\overline{y} \in \mathcal{Y}} \exp\left(\langle \phi\left(x', y^*\right) + \phi\left(x, \overline{y}\right), \theta_t \rangle\right)}{\sum_{\overline{y} \in \mathcal{Y}} \exp\left(\langle \phi\left(x, y^*\right) + \phi\left(x', \overline{y}\right), \theta_t \rangle\right)}.$$

Now, we can lower bound the transition probability by

$$p(x \longrightarrow x') \geq \frac{|\mathcal{Y}| \exp\left(2 \cdot \langle \phi\left(x_{\downarrow}, y_{\downarrow}\right), \theta_t \rangle\right)}{|\mathcal{Y}| \exp\left(2 \cdot \langle \phi\left(x_{\uparrow}, y_{\uparrow}\right), \theta_t \rangle\right)} \geq \exp\left(-2 \cdot \left|\left\langle \phi(x_{\downarrow}, y_{\downarrow}) - \phi(x_{\uparrow}, y_{\uparrow}), \theta_t \right\rangle\right|\right), \tag{5.22}$$

where $\langle \phi\left(x_{\downarrow}, y_{\downarrow}\right), \theta_t \rangle$ and $\langle \phi\left(x_{\uparrow}, y_{\uparrow}\right), \theta_t \rangle$ are the minimum and maximum values of the dot products appearing in the numerator and denominator of $p(x \longrightarrow x')$, respectively.

Then, using the Cauchy–Schwarz inequality, we derive

$$p(x \to x') \geq \exp\Big(-2\Big\|\phi(x_{\downarrow}, y_{\downarrow}) - \phi(x_{\uparrow}, y_{\uparrow})\Big\| \|\theta_t\|\Big).$$

From our assumptions we have that $\|\theta\| \leq R$ and $\|\phi(x,y)\| \leq r$. Thus, it holds that

$$p(x \to x') \geq \exp(-4r\|\theta_t\|) \geq \exp(-4Rr). \tag{5.23}$$

From Eq. (5.23) it follows that the probability of not coalescing for T steps is upper bounded by $\big(1 - \exp(-4r\|\theta_t\|)\big)^T$. Then for $t(\varepsilon) = \Big\lceil \ln \varepsilon / \ln\big(1 - \exp(-4r\|\theta_t\|)\big)\Big\rceil$, we have

$$P(A_{t(\varepsilon)} \neq B_{t(\varepsilon)}) \leq \big(1 - \exp(-4r\|\theta_t\|)\big)^{t(\varepsilon)} \leq \varepsilon,$$

and the result follows from the coupling lemma (e.g., see Lemma 5.17 or Aldous, 1983). □

The bound from Proposition 5.18 does not exploit the fact that the posterior distribution can be related to the stationary distribution of the proposal generator used in the Metropolis–Hastings sampler. The following bound uses this information and gives a significantly better estimate of the worst case mixing of an independent Metropolis–Hastings chain for sampling from $p\,(x \mid y^*, \theta_t)$. In fact, the chain mixes in sublinear time expressed as a function of the approximation quality $\varepsilon > 0$.

Proposition 5.19. *The mixing time $\tau\,(\varepsilon)$ of an independent Metropolis–Hastings chain for sampling from the posterior distribution $p(x \mid y^*, \theta_t)$ is bounded from above by $\left\lceil \frac{\ln 2/\varepsilon}{\ln c/c-1} \right\rceil$, where $c = \max_{x \in \mathcal{X}} p(y^* \mid x, \theta_t)/p(y^*)$.*

Proof. First observe that for all $x \in \mathcal{X}$ it holds that

$$p\,(x \mid y^*, \theta_t) = \frac{p\,(y^* \mid x, \theta_t)\,\rho\,(x)}{p\,(y^*)} \leq c\rho\,(x)\ ,$$

with $c \geq 1$. The result then follows from Theorem 5.11. □

Any bound on the worst case mixing time of the Metropolis–Hastings chain with a proposal generator defined with a conditional transition kernel depends on the specifics of that kernel. Such studies of the mixing time are beyond the scope of this thesis and will be deferred to future work with specific instantiations of Algorithm 5.1. However, we note here that a simple condition can be imposed on the proposal generator such that the corresponding Metropolis–Hastings chain is uniformly ergodic. The following theorem gives a sufficient condition for the uniform ergodicity of the Metropolis–Hastings chain with a proposal generator defined with a conditional transition kernel.

Proposition 5.20. *The Metropolis–Hastings chain is uniformly ergodic if $\mathcal{G}(x \to x') > 0$ for all $x, x' \in \operatorname{supp}\big(p\,(x \mid y^*, \theta_t)\big)$.*

For conditional exponential family models $p\,(y \mid x, \theta) > 0$, the lower bound can be controlled with the regularization parameter. Thus, there will always be a path with non-zero probability between any two target structures. As it is the case with other Metropolis algorithms, for difficult problems where clusters of targets are far apart in the search space, the mixing will be slower as the model becomes more confident.

5.4.3 Handling Large Importance Weights

In this section, we address an issue that can occur when Algorithm 5.1 samples instances corresponding to large importance weights. The issue can slow down the convergence of Algorithm 5.1 and potentially reduce the number of generated targets. We start the section by describing the problem with large importance weights and then review a simple reweighting scheme that can address the issue (Cortes et al., 2010).

In Section 5.4.1, we have shown that under reasonable assumptions importance weights $w(x) = 1/p(y^*|x,\theta)$ are bounded for all $x \in \mathcal{X}$ and $\theta \in \Theta$. This property is crucial for the consistency of the approach, demonstrated in Theorem 5.12. However, while the importance weights are bounded the approach can still select an instance with a large importance weight. Moreover, the latter can happen at initial stages of the algorithm when the sample is small. Such choice can then affect the estimation of the conditional exponential family model in that and the following iterations. In fact, it might take a number of rounds for the algorithm to fix the bias caused by one such importance weight. Such instability can be introduced into the described random process for different reasons. For example, it is possible that the approach over-fits while estimating the model based on a small sample. Alternatively, the issue can be caused by a random bit sequence that just corresponds to taking a sample from a low-density region of the currently estimated model.

To address the problems with model estimation caused by large importance weights, we propose to combine the importance-weighted cross-validation (Sugiyama et al., 2007) with weight capping (Cortes et al., 2010). This type of cross-validation is required because samples selected by Algorithm 5.1 are coming from different distributions and standard cross-validation with such samples would be biased. As shown by Sugiyama et al. (2007), the importance weighting of the samples from validation fold results in almost unbiased estimate of the loss function (in our case, negative log-likelihood). While the importance-weighted cross-validation resolves the issue with a bias introduced into the random process by samples originating from different distributions, it does not eliminate the problem with large importance weights. To address the latter issue, Cortes et al. (2010) have analyzed several reweighting strategies. Among the described reweighting schemes for addressing this issue, empirically the most efficient strategy is based on quantile importance weighting. More specifically, after splitting the sample into groups using quantiles an identical importance weight is assigned to all the samples from the same quantile range. The importance weight corresponding to a quantile range is given by the mean of the importance weights corresponding to samples from that range. While empirically very efficient, the strategy requires cross-validation to determine the best number of quantile ranges. As our approach repeatedly fits models, such cross-validation would be time-consuming. For this reason, we have opted to address the issue with large importance weights with another, much simpler strategy, also described in Cortes et al. (2010). This strategy caps importance weights using a pre-specified threshold value. More formally, for a constant $\eta > 0$, the reweighted importance weights are, for an $x \in \mathcal{X}$ and a $\theta \in \Theta$, given by

$$\tilde{w}(x) = \min\{w(x), \eta\} = 1/\max\left\{p(y^*|x,\theta), \tfrac{1}{\eta}\right\} .$$

5.5 Adaptations to Exemplary Cyclic Discovery Processes

An adaptation of Algorithm 5.1 to a cyclic discovery process can be characterized by three components: *i*) a proposal generator that provides access to an intensionally specified design/search space of interest, *ii*) an evaluation oracle capable of determining or approximating the properties of candidate structures, and *iii*) a kernel function that defines a family of conditional density functions (see Section 5.2) from which the algorithm selects a model that acts as a probabilistic surrogate of a target property evaluated by the oracle. In the remainder of the section, we describe adaptations of the proposed algorithm for three different examples of cyclic discovery processes: *i*) discovery of flavorful cocktail recipes (Section 5.5.1), *ii*) a focused drug design problem in which we search for a molecule effective against a chronic lung disease of significant pharmaceutical interest (Section 5.5.2), and *iii*) synthetic testbeds with properties that share many characteristics with drug design (Section 5.5.3).

Before we proceed with details specific to different discovery processes, we review the standard factorization of a tuple kernel which factors it into the product of domain kernels, $k((x,y),(x',y')) = k_{\mathcal{X}}(x,x')k_{\mathcal{Y}}(y,y')$, where $k_{\mathcal{X}}$ and $k_{\mathcal{Y}}$ are kernel functions over spaces $\mathcal{X}$ and $\mathcal{Y}$. In all considered adaptations of Algorithm 5.1, the property space $\mathcal{Y}$ is binary and equipped with the identity kernel. Such property spaces require the simplest feedback and the least effort from an evaluation oracle. In more complex experiments such as drug design, the evaluation oracle could output a structured label such as binary vector reflecting different aspects of designed molecular structures—binding affinity, toxicity, absorption etc. In these cases, one could take the property space $\mathcal{Y}$ to be the power set of elementary properties and use the intersection kernel $k_{\mathcal{Y}}(y_i, y_j) = |y_i \cap y_j|$.

5.5.1 Discovery of Flavorful Cocktail Recipes

For a customer that repeatedly orders cocktails and asks for recommendations in a bar, a creative bartender is able to suggest novel cocktails based on the feedback from the customer. A good bartender can serve flavorful cocktails without ever relying on a particular database of cocktails. Instead, he can rely on his knowledge and his observations. This cyclic discovery process can be realized through an AI-agent that mimics a creative bartender and one such agent can be created by adapting Algorithm 5.1 to personalized cocktail design. In the remainder of the section, we describe an *in silico* evaluation oracle for this discovery process, together with a kernel function and a proposal generator.

A cocktail recipe can be represented as a sparse real-valued vector with non-negative entries that sum to one. Each component in such a vector corresponds to an ingredient and the non-zero values in one such vector express the proportions of the respective ingredients. For the space of cocktail recipes, we have devised evaluation oracles using a small dataset of cocktails collected from `www.webtender.com`. The dataset was labeled by a human expert and based on such a labeling we have trained decision trees to distinguish two flavor profiles (see Appendix 5.8): dry and creamy cocktails.

To apply Algorithm 5.1 to the space of sparse real-valued vectors, we use the Gaussian kernel with diagonal relevance scale matrix M, i.e., $k_{\mathcal{X}}(x,x') = \exp\left(-1/2(x-x')^{\top}M^2(x-x')\right)$. For each coordinate, we set the relevance scale as

$$m_{jj} = \frac{2\sqrt{n/\text{nnz}}}{\left(\max_{i=1}^{n} x_{ij} - \min_{i=1}^{n} x_{ij}\right)} ,$$

where n denotes the number of instances, nnz the total number of non-zero entries in the data set, d dimension of the instances, and $1 \leq j \leq d$.

Having described the evaluation oracle and kernel function, we describe a method for sampling sparse vectors based on a small set of such instances. This exemplary set of instances serves as side knowledge facilitating the design of a proposal generator. We use this method in Section 5.6 to propose cocktail recipes which are represented as sparse real-valued vectors.

Algorithm 5.3 Sparse Vector Sampler

```
Input: moment-matched parameters $\eta$, $\{\tau_i\}_{i=0}^{d}$, $\{u_i\}_{i=1}^{d}$, $\{v_i\}_{i=1}^{d}$, $\{\mu_i\}_{i=1}^{d}$
Output: sparse $d$-dimensional vector $x$
1: $x \leftarrow 0$
2: repeat $n \sim \text{Poisson}(\eta)$ until $n > 1$
3: $c_1 \sim \text{Discrete}(\tau_0)$ and $x_{c_1} \sim \text{Triangular}\left(\mu_{c_1} \mid u_{c_1}, v_{c_1}\right)$
4: for $k = 2, \ldots, n$ do
5:     $\pi \leftarrow \sum_{i=1}^{k-1} \tau_{c_i}$ and $\pi_{c_i} \leftarrow 0$ with $1 \leq i < k$
6:     $c_k \sim \text{Discrete}(\pi)$ and $x_{c_k} \sim \text{Triangular}\left(\mu_{c_k} \mid u_{c_k}, v_{c_k}\right)$
7: end for
```

Algorithm 5.3 is a pseudo-code description of the sparse vector sampler. The inputs to the algorithm are parameters defining the sampling process. In the first step of the sampling process, the algorithm sets all the components of a d-dimensional vector to zero (line 1). Following this, the approach selects the number of components in the initialized vector with non-zero values (line 2). This is achieved by sampling the *Poisson* distribution with mean parameter η, provided as input to the algorithm, until the sampled number of non-zero components is greater than one. Having selected the number of non-zero components, the algorithm proceeds to sample their positions within the vector and corresponding values. The first non-zero component is obtained by sampling from a discrete distribution given by the parameter vector τ_0 such that each component in that parameter vector quantifies its frequency of appearance in the exemplary set of instances (line 3). In this way, the algorithm ensures that components with high appearance frequency in the exemplary set of instances are more likely to be sampled than other components. Having sampled the first non-zero component c_1, the algorithm sets its value by sampling the *Triangular* distribution given by its mode parameter μ_{c_1} and having support on the interval $[u_{c_1}, v_{c_1}]$. Following this, the algorithm proceeds to sample the remaining non-zero components and their values (lines $4 - 7$). While sampling the remaining components, the algorithm needs to ensure that combinations of frequently co-occurring components from the exemplary set of instances are sampled more often. This is achieved with the help of parameters specifying a component graph such that each component is assigned to a vertex in the graph and the edges between components co-occurring in the exemplary set of instances are weighted according to their co-occurrence frequency. Thus, having sampled the first non-zero component in vector x, the algorithm samples the next one from a discrete distribution given by the row-vector from the adjacency matrix of the component graph that corresponds to the first sampled component. The procedure is repeated for the following component with the difference being that the sum of the row-vectors of the already sampled components defines the discrete distribution of the available components (i.e., not sampled in the previous steps).

The parameter values, given as input to Algorithm 5.3, are estimated from exemplary instances by moment-matching them from data. A pseudo-code description of the parameter estimation is given in Algorithm 5.4. The input to the algorithm is a sample of d-dimensional sparse vectors. In the first step of parameter initialization, the algorithm sets the number of non-zero components in each of the exemplary instances (line 1). Following this, the parameter specifying the Poisson distribution of the number of components in Algorithm 5.3 is initialized with the average number of non-zero components in the exemplary set of instances (line 2). Then, the procedure initializes the parameter vector quantifying the occurrence frequency of each component in the set of exemplary instances (line 5) and the

Algorithm 5.4 Initialization of Sparse Vector Sampler

```
Input: sample of sparse $d$-dimensional vectors $\{x_i\}_{i=1}^n$
Output: moment-matched parameters $\eta$, $\{\tau_i\}_{i=0}^d$, $\{u_i\}_{i=1}^d$, $\{v_i\}_{i=1}^d$, $\{\mu_i\}_{i=1}^d$
1: $n_i \leftarrow |\{k \mid 1 \leq k \leq d \wedge x_{ik} \neq 0\}|$, where $1 \leq i \leq n$
2: $N \leftarrow \sum_{i=1}^n n_i$ and $\eta \leftarrow N/n$
3: $\tau_{kk} \leftarrow 0$, where $1 \leq k \leq d$
4: for $j = 1, 2, \ldots, d$ do
5:     $\tau_{0j} \leftarrow \frac{1}{N} \sum_{i=1}^n \mathbb{I}_{x_{ij} \neq 0}$
6:     $\tau_{kj} \leftarrow \frac{\sum_{i=1}^n \mathbb{I}_{x_{ij} \neq 0 \wedge x_{ik} \neq 0}}{\sum_{i:\, x_{ik} \neq 0} (n_k - 1)} + \frac{1}{d}$, where $k \neq j$ and $1 \leq k \leq d$
7:     $\mu_j \leftarrow \frac{\sum_{i:\, x_{ij} \neq 0} x_{ij}}{\sum_{i=1}^n \mathbb{I}_{x_{ij} \neq 0}}$ and $\sigma_j \leftarrow \sqrt{\frac{\sum_{i:\, x_{ij} \neq 0} (x_{ij} - \mu_j)^2}{\sum_{i=1}^n \mathbb{I}_{x_{ij} \neq 0}}}$
8:     $u_j \leftarrow \max\left\{0, \mu_j - 2\sigma_j\right\}$ and $v_j \leftarrow \min\left\{1, \mu_j + 2\sigma_j\right\}$
9: end for
```

adjacency matrix of the component graph weighted with co-occurrence frequencies of pairs of components (line 6). The remaining parameters specify the triangular distributions of values for each of the components. The triangular distribution of a component is given by the mode which is set to the mean value computed over the instances with non-zero values at that component and interval endpoints specifying the support of the distribution (lines 7-8). To allow sampling of sparse vectors with combinations of non-zero components which are not appearing together in the exemplary set of instances, we perform the Laplace smoothing of parameter vectors $\{\tau_i\}_{i=1}^d$ by adding $1/d$ to each of its components.

As the described proposal generator almost always samples recipes with 2-10 ingredients, for n possible ingredients the number of different ingredient combinations is $\sum_{k=2}^{10} \binom{n}{k}$ (approximately n^{10}). As the sampler is developed based on a set of cocktails with 335 ingredients there are approximately 10^{24} different combinations of ingredients in this search space. Thus, this is a huge search space that can provide an insight into the properties of the discovery process on large scale problems.

5.5.2 Focused Drug Design Problem

In this section, we propose an adaptation of Algorithm 5.1 to a focused drug design problem aimed at designing pharmaceuticals that are effective against idiopathic pulmonary fibrosis. This is a chronic lung disease with an urgent need for new medicines. The disease is characterized by scar tissue which forms in the lungs with increasing severity and it is often caused by micro-injuries from tobacco smoking, inhalation of micro particles, such as wood and metal dust, or by viral infection (Liu et al., 2017). In the United States, about 100 000 people have idiopathic pulmonary fibrosis, and a similar number in Europe. Each year approximately 35 000 new patients are diagnosed in Europe. The best current treatment, lung transplantation, is available to only 5% of patients. The recently approved drugs, Pirfenidone and Nintedanib, slow the disease but have side effects and do not reverse it (Liu et al., 2017).

Integrins are relatively large proteins that act as transmembrane receptors. They link the extracellular matrix with the cytoskeleton of cells. The general structure of an integrin is a heterodimer, consisting of an α and a β subunit. The group of RGD integrins recognizes an arginine-glycine-aspartate sequence in the endogenous ligands that bind at the interface of the two subunits. The RGD integrin receptors are thought to play a key role in fibrosis (and many other diseases, including cancer) and are likely to be druggable targets (Ley et al., 2016; Hatley et al., 2017; Reed et al., 2015). Antagonism of $\alpha_v\beta_6$ is one promising avenue of inquiry and some success has been reported (Adams et al., 2014) in discovering compounds with significant activity against $\alpha_v\beta_6$ that have physico-chemical properties commensurate with oral bioavailability. Moreover, molecular dynamics simulations of peptides binding to $\alpha_v\beta_6$ have been conducted (Maltsev et al., 2016), but there are only few published studies on docking of small molecules to this integrin. Motivated by these considerations, we search for an antagonist of an $\alpha_v\beta_6$ protein structure. As an *in silico* proxy of the binding affinity, we use a molecular docking score to an experimentally determined $\alpha_v\beta_6$ protein structure.

The docking program takes as input a molecular structure and outputs a real-valued score that quantifies the quality of the docking of that molecule to the $\alpha_v\beta_6$ receptor site. To obtain a discrete/binary label for the activity of a designed structure, we threshold the docking score. To determine a suitable threshold value, we have performed a series of preliminary experiments and defined a binary labeling oracle that assigns label 1 to molecular structures with a docking score below -11.75. Whilst it is recognized that there are many conformational changes of α_v integrins during their activation and signaling, we have elected to base our modeling on a published crystal structure. The structure (Dong et al., 2014) was taken from the Protein Data Bank (PDB code: 4um9). The zwitterionic forms of the ligands were considered, with the negatively charged carboxylate moiety at one end (coordinating to a metal in the MIDAS site) and the naphthyridine protonated (having a pKa of ≈ 7.8), making the aromatic nitrogen atom positively charged (Cacciari et al., 2009). This is important for a bidentate hydrogen bond interaction with Asp218. Molecular docking was performed using OpenEye FRED (McGann, 2011), which uses a rigid ligand approach, where a large number of conformations are generated and each of those are docked successively. The chemgauss3 scoring function was used in an initial docking, and the highest scoring positions were evaluated using the more sophisticated (and more computationally expensive) chemgauss4 scoring function, which includes improved terms for ligand-receptor hydrogen bonds and metal-chelator interactions. The latter is particularly pertinent, considering the importance of binding with the divalent metal cations within the active site (Millard et al., 2011). Both enantiomers were sampled separately, while individual conformers were deemed identical (and removed) if the root mean squared difference in the atomic positions was less than 0.5 Å. A maximum of $10\,000$ conformers was allowed per enantiomer and typically there were between $2\,500$ to $5\,000$ conformers per enantiomer. To allow more extended sampling of the conformational space, a truncated form of the MMFF94s forcefield (Halgren, 1999) was used to calculate individual conformer energy and the maximum range between the global minimum and any conformer was limited to 20 kcal/mol. This truncated form of the forcefield excludes both Coulomb and the attractive part of van der Waals interactions. The binding box used for the docking was centered on the Thr221 residue, in the middle of the active site. The edges were extended past important features, namely Asp218 and the Mg_2^+ ion, such that the final size was 27.0 by 29.7 by 21.3 Å, giving a total search volume of $17{,}010$ Å^3. The binding site was further restricted by enforcing an interaction with the divalent metal cation as well as a hydrogen bond with the Asp218 residue; a single hydrogen bond was required in order not to restrict the search space unduly. The grid point spacing was 1 Å with a second pass grid point spacing of 0.5 Å.

We represent molecular structures as vertex labeled graphs and use the Weisfeiler–Lehman graph kernel (Shervashidze et al., 2011) to embed these structures to a reproducing kernel Hilbert space. This graph kernel has been shown to be highly expressive on prediction tasks involving molecules (Shervashidze et al., 2011) and it is related to the molecular

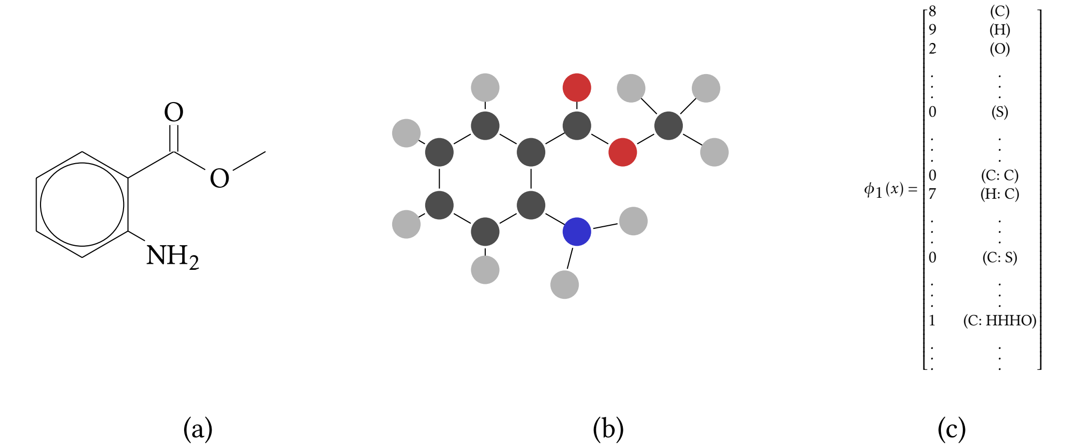


Figure 5.2: An illustration of the Weisfeiler–Lehman transformation for generating feature vectors from molecules. (a) An example molecular structure (Methyl anthranilate). (b) An undirected vertex labeled graph is formed from the molecule (notice that bond type is ignored). (c) Features are generated according to the appearance frequency of subtree patterns rooted at any of the vertices from the labeled graph. The height of any such subtree pattern needs to be less than a pre-specified parameter value (e.g., all subtree patterns of height less than 10). In this feature vector, additional components would be present for other subtree patterns such as P (here 0), N: CHH (here 1), C: COO (here 1), H: CHN (here 2) etc.

fingerprint called ECFP (Rogers and Hahn, 2010). In contrast to the Weisfeiler–Lehman graph kernel, that fingerprint: *i*) ignores hydrogen atoms, *ii*) uses binary features instead of counts to express occurrences of vertex centered subtree patterns in molecular graphs, and *iii*) incorporates more information than atomic number into the initial labels. Figure 5.2 provides an illustration of the Weisfeiler–Lehman embedding of a molecular structure.

A typical approach to candidate generation in drug design is to make alterations to a parent compound having a moderate binding affinity to a target protein site. Changes to the parent compound can include modifications to its functional groups and attachment of different fragments in the place of hydrogen atoms or small fragments contained in the parent molecule. Motivated by these approaches, we develop a proposal generator for finding candidate molecules of which some are likely to dock well to the receptor site. In particular, we start with an integrin antagonist compound as the parent and consider substitutions at five possible points on the aryl ring (Figure 5.3), which based on structure activity relationships, are known to profoundly influence potency and selectivity (Adams et al., 2014). Based on the integrin medicinal chemistry literature, we consider a variety of possible substituents: H, F, Cl, Br, methyl, ethyl, propyl, iso-propyl, cyclopropyl, methoxy, hydroxyl, CF_3, OCF_3, SO_2Me, nitrile and several heterocycles, imidazole, pyrazole and triazole (with possible substituents of H, methyl or ethyl). After a preliminary calculation, we elected to impose a couple of restraints on the molecules that could be generated, so that they would be more drug-like and more amenable to synthesis. Thus, catechols (where there are neighboring hydroxyls on the aryl ring) were precluded, as they are prone to autoxidation and, therefore, difficult to work with experimentally. The total number of hydrogen bond donors that could be present in a molecule was capped at five. A maximum of 500 was set for the molecular weight. Clearly, more of the Lipinski's rules (Lipinski et al., 2001) or other in silico restrictions (e.g., polar surface area) could be readily implemented, but the above restraints proved to be sufficient.

Algorithm 5.5 is a pseudo-code description of the proposal generator. The algorithm takes as input a parent compound, together with a set of fragments and a set of attachment points onto which the fragments can be substituted instead of hydrogen atoms. As described above, the Lipinski constraints are enforced with a maximum allowed molecular weight

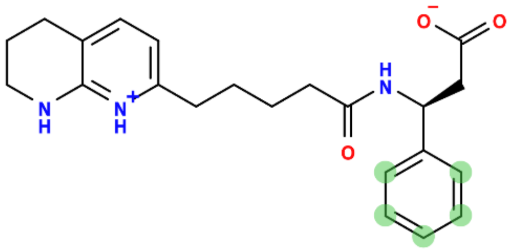


Figure 5.3: The parent compound considered in this focused drug design problem; green circles denote points where substituents could be attached.

and a maximum allowed number of hydrogen bond donors in the resulting compound. The sampling process is initialized by setting the total molecular mass to that of the parent compound and the total attached fragment mass to zero. Also, the set of attachment points of fragments with hydrogen bond donors is initialized with the empty set. The algorithm then starts to iterate until: *i*) there are no available attachment points, *ii*) there are no feasible fragments to be substituted at the available attachment points, *iii*) a random interruption event occurs, which is defined to happen with probability given by the ratio of the attached molecular mass and the total available attachment mass. This probabilistic constraint is introduced so that it is more likely to sample lighter compounds. The alterations to the parent compound are achieved through two steps: *i*) sampling uniformly at random an attachment point from the available ones, and *ii*) sampling a fragment uniformly at random from the set of feasible fragments for the sampled attachment point. For the constraints that were applied, there are approximately 185 000 different compounds that define our search space.

Algorithm 5.5 Molecular Structure Sampler

```
Input: parent compound $\mathcal{B}$, set of fragments $\mathcal{F}$, set of attachment points $\mathcal{A}$, maximum molecular weight $m$, maximum number of hydrogen bond donors $h$
Output: molecular structure
$\mathcal{H} \leftarrow \emptyset$, $m_{total} \leftarrow m(\mathcal{B})$, $m_{attached} \leftarrow 0$, $u \sim \mathcal{U}[0,1]$
repeat
    $a \sim \mathcal{U}(\mathcal{A})$
    if hydrogen bond donor neighbor of $a$ then $\mathcal{F}' \leftarrow \{f \in \mathcal{F} \mid h(f) = 0\}$ else $\mathcal{F}' \leftarrow \mathcal{F}$ end if
    $f \sim \mathcal{U}(\mathcal{F}')$
    $\mathcal{B}[a] \leftarrow f$
    if hydrogen bond donor in $f$ then $\mathcal{H} \leftarrow \mathcal{H} \cup \{a\}$ end if
    $m_{total} \leftarrow m_{total} + m(f)$ and $m_{attached} \leftarrow m_{attached} + m(f)$
    $\mathcal{F} \leftarrow \{f \in \mathcal{F} \mid m(f) + m_{total} \leq m \wedge h(f) + h(\mathcal{B}) + |\mathcal{H}| \leq h\}$
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{a\}$
    $u \sim \mathcal{U}[0,1]$
until $u > \frac{m_{attached}}{m - m(\mathcal{B})}$ and $\mathcal{F} \neq \emptyset$ and $\mathcal{A} \neq \emptyset$
```

5.5.3 Synthetic Testbeds

The main objective of our synthetic testbeds is to demonstrate that our approach can discover a diverse set of target-structures in non-smooth problems which act as *in silico proxies* for drug discovery. In particular, in the construction of Hamiltonian graphs and complements of these, there are numerous Hamiltonian graphs which become non-Hamiltonian with a removal of a single edge. Such graphs are structurally very similar and close in the design space. Thus, these testbeds can mimic well the activity cliffs specific to drug design where very similar structures have different binding affinities. To apply our algorithm to the space of graphs we use the random walk kernel (Gärtner et al., 2003). The kernel performs random walks on both graphs and counts the number of matching walks. It can be computed as

$$k_{\mathcal{X}}(G_1, G_2) = \sum_{i,j=1}^{|\mathcal{V}_\times|} \sum_{n=0}^{\infty} [\lambda_n E_\times^n]_{ij} \ , \tag{5.24}$$

where $E_\times$ denotes the adjacency matrix of the product graph $G_1 \times G_2$ and $\{\lambda_n\}$ is a sequence of hyperparameters that needs to be set such that the sum in (5.24) converges for any pair of graphs G_1 and G_2. We apply the kernel with $\lambda_n = \lambda^n$ to unlabelled graphs, and for this particular case $E_\times = E_1 \otimes E_2$. The kernel can be computed efficiently using the fixed-point method described by Borgwardt (2007).

As the set of graphs is a complicated, combinatorial object, it can be difficult to design an efficient proposal generator. In general, to sample a random unlabelled graph it is common to use the Erdős–Rényi model with $p = 1/2$. This approach, however, samples some graphs too often and does not provide sufficient diversity to the constructive process (e.g., the probability of sampling an unlabelled path with n vertices is $\frac{n!}{2}$ times higher than the probability of sampling the complete graph with the same number of vertices). Instead, one could try to first sample the parameter p uniformly at random and then to sample a graph with edge probability p. The last method does not generate unlabelled graphs u.a.r., but it can be used to efficiently sample some graph concepts (e.g., acyclic graphs). In our simulations (Section 5.6), we take the safest route and choose to propose graphs with n vertices using the uniform sampler. In the remainder of the section, we provide a brief review of this sampler.

Let $\mathcal{G}_n$ denote the set of all canonically labelled graphs with n vertices. A *left action* of a group S on a set X is a function $\mu\colon S \times X \rightarrow X$ with the following two properties: (i) $(\forall x \in X)(\forall s, t \in S) : \mu(t, \mu(s, x)) = \mu(ts, x)$; (ii) $(\forall x \in X) : \mu(e, x) = x$ (where e is the identity element of the group S). If no confusion arises we write $\mu(s, x) = sx$. A group action defines the equivalence relation $\sim$ on a set X, i.e., $a \sim b \Leftrightarrow sa = b$ for some $s \in S$ and $a, b \in X$. The equivalence classes determined by this relation are called *orbits* of S in X. The number of orbits of a group S in a set X can be computed using the Frobenius–Burnside theorem.

Theorem 5.21. *(Frobenius–Burnside Theorem, Cameron, 1998) Let X be a finite non-empty set and let S be a finite group. If X is an S-set, then the number of orbits of S in X is equal to*

$$\frac{1}{|S|} \sum_{s \in S} \Big| \{x \in X \mid sx = x\} \Big| \ .$$

To sample unlabelled graphs uniformly at random, Wormald (1987) proposed a rejection sampling method based on Theorem 5.21. The idea is to consider the action of a symmetric group S_n over the set $\mathcal{G}_n$. Then, the orbits of S_n in the set $\mathcal{G}_n$ are non-isomorphic unlabelled graphs and to sample unlabelled graphs uniformly it suffices to uniformly sample the orbits (Dixon and Wilf, 1983). Moreover, it is possible to show (Dixon and Wilf, 1983; Wormald, 1987) that uniform orbit sampling is equivalent to uniform sampling from the set

$$\Gamma = \Big\{ (\pi, g) \mid \pi \in S_n \wedge g \in \mathrm{Fix}(\pi) \Big\} \ ,$$

where $\mathrm{Fix}(\pi) = \{g \in \mathcal{G}_n \mid \pi g = g\}$. According to Theorem 5.21, an element $(\pi, g) \in \Gamma$ can be sampled u.a.r. by choosing a permutation π with probability proportional to $|\mathrm{Fix}(\pi)|$ and then choosing $g \in \mathrm{Fix}(\pi)$ uniformly at random. Dixon and Wilf (1983) proposed a more efficient sampling algorithm by partitioning the symmetric group into conjugacy classes $[\pi_i]$ $(1 \leq i \leq l)$ and sampling: (*i*) $[\pi_i] \sim {}^{|[\pi_i]||\mathrm{Fix}(\pi_i)|}/_{o_n|S_n|}$, (*ii*) $g \in \mathrm{Fix}(\pi_i)$ u.a.r.; where o_n denotes the number of non-isomorphic unlabelled graphs and π_i is a class representative for the class $[\pi_i]$. As it holds $|\mathrm{Fix}(\pi)| = |\mathrm{Fix}(\pi')|$ and $|\mathrm{Fix}(\pi) \cap [g]| = |\mathrm{Fix}(\pi') \cap [g]|$ for $\pi, \pi' \in [\pi_i]$ then (e.g., see Wormald, 1987, for a more detailed proof)

$$P\big([g]\big) = \sum_{i=1}^{l} P\big([g], [\pi_i]\big) = \sum_{i=1}^{l} P\big([\pi_i]\big) P\big([g] \mid [\pi_i]\big) = \sum_{i=1}^{l} \frac{\big|[\pi_i]\big|\big|\mathrm{Fix}(\pi_i)\big|}{o_n |S_n|} \frac{\big|\mathrm{Fix}(\pi_i) \cap [g]\big|}{\big|\mathrm{Fix}(\pi_i)\big|} = \frac{1}{o_n}.$$

The problem with the approach is the fact that we need to know the exact number of non-isomorphic graphs with n vertices o_n to apply the algorithm and this number is not computable in polynomial time. To overcome this, Wormald (1987) partitions the elements of the group S_n into classes $[c_k] = \{\pi \in S_n \mid \mathrm{support}(\pi) = k\}$, $0 \leq k \leq n$, and upper bounds $\big|[c_i]\big|\big|\mathrm{Fix}(\pi_i)\big| \leq B_i$. The algorithm then samples an unlabelled graph u.a.r. as follows: (*i*) $[c_i] \sim {}^{B_i}/_{\sum_j B_j}$, (*ii*) $\pi_i \in [c_i]$ u.a.r., (*iii*) $g \in \mathrm{Fix}(\pi_i)$ u.a.r., and (*iv*) accept the sampled graph g with probability $B_i^{-1}\big|[c_i]\big|\big|\mathrm{Fix}(\pi_i)\big|$; otherwise, restart. On average, the method generates an unlabelled graph in time polynomial in the number of vertices.

5.6 Experiments

Having provided theoretical justification for our approach and means to adapt it to different cyclic discovery processes, here we evaluate its effectiveness with a series of experiments that are designed to mimic the discovery of cocktail recipes, pharmaceuticals, and graphs with desired properties. We first describe the baselines (Section 5.6.1) used to assess the effectiveness of the approach and then present the results of our empirical evaluation. In the first part of the empirical evaluation (Section 5.6.2), we focus on evaluating the effectiveness of the approach in relation to the baselines on synthetic testbeds and discovery of flavorful cocktails. For these cyclic discovery processes, the properties evaluated by the oracle are not computationally intensive and allow for a comparison from different perspectives. Having established that the proposed approach can discover a diverse set of structures with desired properties, we apply it to a focused drug design problem described in Section 5.5.2. The preliminary results (presented in Section 5.6.3) augur well for the future, more extensive work, which will include more extended search and exploration of significantly larger spaces.

5.6.1 Baselines

Our first baseline is k-NN active search proposed by Garnett et al. (2012). While the approach represents a good baseline from the perspective of our success measure (i.e., generate as many targets as possible), it is not designed for search in intensionally specified structured spaces. In particular, the algorithm requires a fixed set of instances to be provided as input. To account for this limitation of k-NN active search we have simulated the approach with a pool of 50 000 instances sampled from a proposal generator. For the number of nearest neighbors in the k-NN probabilistic model acting as a surrogate of the target property, we have selected $k = 50$ (ties are not possible due to the choice of the hyperparameters). In order to apply this approach to large sets of structures with a different probabilistic model an efficient pruning strategy needs to be devised. In the original paper, the authors gave pruning rules only for the k-NN probabilistic model. As it is non-trivial to come up with pruning rules for conditional exponential family models, a search with these models would be inefficient. For example, for the investigated case of Hamiltonian graphs with 10 vertices and an extensional description in the form of a sample with less than 1% of all structures from this space k-NN active search with 2-step look-ahead (Garnett et al., 2012) and a budget of 500 oracle evaluations requires more than 50 million parameter fittings for the search modeled with a conditional exponential family. The latter is clearly inefficient and the reason for simulating this approach only with the k-NN probabilistic model.

Our second baseline is deterministic greedy argmax search that first takes a sample of instances from a proposal generator and then picks an instance from this sample with the highest conditional probability of having a target property. The selected instance is then evaluated by the oracle and the conditional model is updated to account for this new observation. This method is designed to compensate for the fact that k-NN active search with 1-step look-ahead (Garnett et al., 2012) requires a finite sample of instances. For graphs, the approach is combined with the uniform proposal generator, the most exploratory proposal generator for this type of greedy search. The approach is simulated with conditional exponential family model and kernels described in Section 5.5.

All the reported results were obtained by averaging over 5 runs of the respective algorithms. In Algorithm 5.1, the Metropolis–Hastings sampling was performed with a burn-in sample of 50 000 proposals and sampling was done for 50 rounds/batches. In each round we take 10 i.i.d. samples by running 10 Metropolis–Hastings chains in parallel (note that samples from different rounds are dependent). To allow for models of varying complexity, we have estimated the conditional exponential family regularization parameter in each round using 5-fold stratified cross-validation. As the competing approaches are not designed to search for targets without an a priori provided labeled structures, we have made a minor modification to our problem setting and warm-started each method with a random sample of 5 target and the same number of non-target structures. For graphs these were chosen uniformly from the search space and for cocktails uniformly from the available sample of cocktails. Note that without this warm-start the argmax search estimates the distribution of target structures with a single peak around the first discovered target. Moreover, k-NN probabilistic model cannot learn a property until it sees more than k labeled structures and it is unlikely to observe a target in k successive samples from a proposal generator.

5.6.2 Comparison against Baselines

In the first set of experiments, we design cocktails of different flavors (see Section 5.5.1), Hamiltonian and connected planar graphs (see Section 5.5.3), as well as the respective complements of these classes. As we can not expect to be able to perfectly distinguish each of the graph concepts from its complement due to the hardness of complete graph kernels Gärtner et al. (2003), we can not expect to learn to perfectly generate these concepts. The main objective of these experiments is to demonstrate that our approach can discover a diverse set of target-structures in non-smooth problems which act as *in silico proxies* for drug design.

As described in Section 5.6.1, we compare Algorithm 5.1 to k-NN active search with 1- and 2-step look-ahead (Garnett et al., 2012) and a deterministic greedy approach which discovers structures by repeatedly performing *argmax* search over samples from a proposal generator using the learned conditional label distribution (selected structures are labeled by an oracle and the model is updated in each iteration). In the first step of this evaluation, we measure the improvement of each of the considered approaches over plain Monte Carlo search performed with a proposal generator. We assess the performance of the approaches with correct-construction curves which show the cumulative number of distinct target structures discovered as a function of the budget expended (see Section 5.1). To quantify the

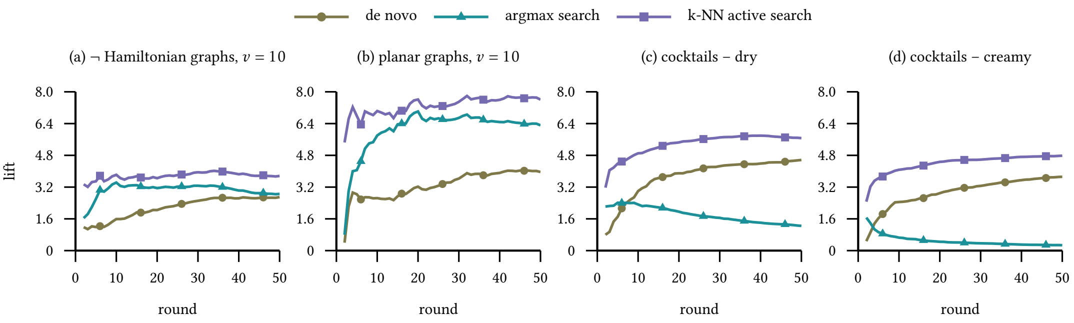


Figure 5.4: The figure shows the lift of correct-construction curves for graph and cocktail concepts, which indicates how much more likely it is to see a target compared to Monte Carlo search.

improvement of the approaches over plain Monte Carlo search, we measure the lift of the correct-construction curves. In particular, for sampling from the minority class of a proposal generator the lift is computed as the ratio between the number of distinct structures from this class generated by an algorithm and the number of such structures observed in a sample (of the same size) from the distribution of the proposal generator. In the second step of our empirical evaluation, we assess the structural diversity between the targets discovered by an algorithm. We do this by incorporating diversity into the correct-construction curves (see Section 5.1). In particular, we take a sample of 50 000 structures from the proposal generator and filter out targets. We consider these as undiscovered targets and compute the average distance between an undiscovered structure and a subsample of budget size from this set of structures. With this average distance as radius we circumscribe a sphere around each of the undiscovered targets. Then, instead of construction-curves defined with the number of discovered targets, we use the construction-curves defined with the number of the spheres having a target structure within them. To quantify the effectiveness of the considered algorithms in discovering structurally diverse targets, we normalize these sphere based construction-curves with one such curve corresponding to an ideal algorithm that only generates targets – the output of this algorithm can be represented with a subsample of budget size from the undiscovered target structures.

In Figure 5.4, we show the lift of the correct-construction curves for all the considered approaches. We have defined these correct-construction curves by considering isomorphic graphs and cocktails with equal sets of ingredients (ignoring portions of each ingredient) as identical structures. The plots indicate that our approach and k-NN active search are able to emphasize the target class in all the domains for all the considered properties. Moreover, for our approach the magnitude of this emphasis is increasing over time and it is more likely to generate a target as the process evolves. In all domains and for all properties, k-NN active search discovers more target structures than our approach. For graph properties, we see that argmax search also discovers more targets than our approach. For cocktails, argmax search discovers many cocktails with identical sets of ingredients and different portions of these (such cocktails are considered identical in the correct-construction curves). Thus, if we are only interested in discovering target structures without considering structural diversity between them, our empirical evaluation indicates that it is better to use k-NN active search than Algorithm 5.1.

In Figure 5.5, we show the dispersion of target structures discovered by each of the considered approaches. The plots indicate that our approach achieves a large structural variety of discovered targets. In all domains and for all properties, our approach outperforms

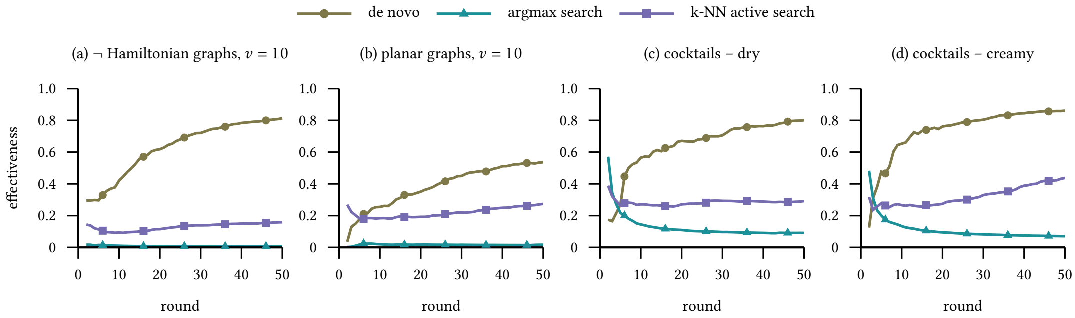


Figure 5.5: The figure shows the dispersion of discovered targets relative to an algorithm with the identical proposal generator that outputs only targets. The reported curves can be seen as the percentage of discovered target class partitions given a budget.

both k-NN active and greedy argmax search. These experiments also indicate that k-NN active search explores more than argmax search. In some of the plots, a dip can be observed in the curves for k-NN active and argmax search. This can be explained by the exploitative nature of these algorithms and the fact that the search is focused to a small region of the space until all the targets from it are discovered. In contrast to this, our approach discovers targets from the whole search space and can cover a large number of spheres centered at undiscovered samples with a relatively small number of targets. Thus, if we are interested in discovering diverse target structures, our results indicate that it is better to use Algorithm 5.1 than k-NN active or argmax search.

Graphs, $v = 7$		Graphs, $v = 10$		Cocktails	
Hamiltonian	Connected Planar	Hamiltonian	Connected Planar	dry	creamy
36.68% (±0.24)	65.01% (±0.20)	77.45% (±0.28)	8.68% (±0.15)	11.27% (±0.14)	16.83% (±0.14)

Table 5.1: The table shows the fraction of target structures observed within 50 000 samples from proposal generators. The sampling was performed 5 times and the reported values are mean and standard deviation of the fractions computed for these runs.

5.6.3 Drug Discovery

Having established the effectiveness of our approach on synthetic testbeds and *in silico* discovery of flavorful cocktails, we proceed to evaluate its effectiveness on a focused drug design problem where we search for pharmaceuticals effective against idiopathic pulmonary fibrosis (see Section 5.5.2). We assess the performance of the algorithm from several different perspectives. First, we confirm that the approach represents a significant improvement over plain Monte Carlo search performed with the proposal generator. Following this, we quantify the learning rate of our approach by measuring how much more likely the approach is to generate desired molecular structures compared to the proposal generator as a function of the budget expended. Having assessed the algorithmic performance of the approach, we proceed to analyze the designed molecules from the perspective of medicinal chemistry. In particular, we discuss some of the designed molecular structures in the context of compounds (Adams et al., 2014) already reported in the literature.

Similar to Section 5.6.2, we evaluate the effectiveness of the approach using the correct-construction curve that shows the cumulative number of discovered molecules exhibiting the target property, i.e., a docking score lower than −11.75, as a function of the budget

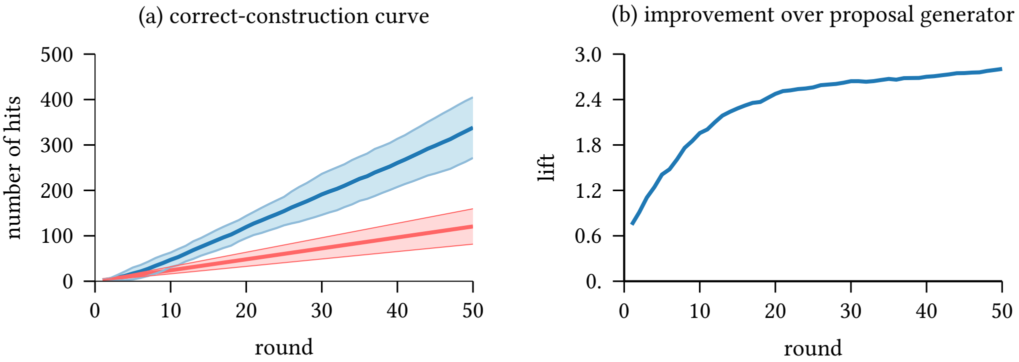


Figure 5.6: Panel (a) shows the cumulative numbers of *hits* as a function of the budget expended. The blue curve is the correct-construction curve of our approach (with corresponding confidence interval colored in light blue) and the red curve is the correct-construction curve of the proposal generator. Panel (b) indicates how much more likely it is to see a hit compared to a standard Monte Carlo search performed with the proposal generator.

expended. Figure 5.6, which shows correct-construction curves for our approach (blue curve) and the described proposal generator (red curve), confirms that our approach generates more hits than Monte Carlo search with the proposal generator. Moreover, the correct-construction curve of the proposal generator is, apart from a few initial rounds, always below the lower endpoint of the confidence interval for the curve of our approach. The lift of the correct-construction curve for our approach (showed in Figure 5.6, Panel b) indicates that the approach is approximately 2.8 times more likely to output a hit than the proposal generator after 50 rounds of model calibration. The results presented in this study have been generated using simulations consuming approximately 28 hours of CPU time and running on 10 processors in parallel. This offers significant speed up over an exhaustive exploration of the search space specified by the proposal generator that would take more than 8 months of CPU time using the same number of processors. Moreover, while the simulations were relatively short (with a budget of 500 evaluations), the approach managed to discover a number of interesting compounds from the perspective of medicinal chemists.

The experimental knowledge provided to the algorithm was limited to the X-ray crystal structure of the receptor and some basic constraints on the mode of binding applied to the molecular docking. The parent compound and the possible fragments were informed, in a broad sense, by expert knowledge from medicinal chemistry, but no explicit data on the experimental activities of any compounds were used. Previous work (Adams et al., 2014) presented the synthesis and experimental assay (reported as pIC_{50} values, i.e., the negative logarithm of the concentration required for 50% inhibition) of 30 derivatives of the parent compound shown in Figure 5.3. A $\text{pIC}_{50} = 6.0$ corresponds to a $1\mu M$ potency, and the compound might be considered active or worth further investigation. From Table 5.2, we can see that the parent compound has a $\text{pIC}_5 0$ of 5.7 and would, therefore, be considered inactive. Encouragingly, 19 out of the 26 reported active compounds were found by the algorithm. Several of the compounds reported as active were not found, but two of these were not discoverable by the algorithm, because the substitution pattern (forming new ring structures) was not part of the proposal generator. There were two compounds for which the docking score was not sufficiently low, which indicates that there is an opportunity to improve the docking protocol. In total, 20 of the 30 compounds known from previous work (Adams et al., 2014) were discovered.

The described cyclic discovery process (see Section 5.5.2) is a proof of concept and still requires considerable refinement but nonetheless, from a medicinal chemistry and drug discovery perspective, the molecules suggested for synthesis are promising for several reasons. First, many of the molecules suggested align with the structure activity relationships (Anderson et al., 2016a,b) which were not part of the input to the algorithm, either in terms of parameters or design. An example is the algorithm predominantly suggests substitutions at the meta position, which indeed appears to be crucial for $\alpha_v\beta_6$ activity. Suggested substitutions also often feature heterocycles which are known to deliver $\alpha_v\beta_6$ activity (Anderson et al., 2016a,b). Secondly, most of the molecules are drug-like: that is, they resemble both the structures and physico-chemical properties of oral drugs. Thirdly and particularly promising is the speed at which new molecules can be evaluated computationally allowing several iterations to be easily carried out to improve the design quality of the molecules (as detailed earlier). Moving forward, it will be straightforward to incorporate additional constraints, such as scoring molecules against $\alpha_v\beta_3$, which should improve the selectivity window for $\alpha_v\beta_6$ over $\alpha_v\beta_3$, including polar surface area cut-offs (which correlate with several important drug-like properties) and simple synthetic chemistry considerations.

No.	Fragments	Docking Score	pIC_{50}	No. Hits
		Compounds independently identified as hits		
[4]	3-F	−12.16	6.1	25
[22]	3-MeO	−11.79	6.5	8
[25]	4-Me	−11.96	6.1	7
[32]	3-CN	−11.94	6.6	23
		Compounds not discovered by the algorithm		
[31]	4-Ph	−11.92	6.4	-
[38]	3,4-Me_2	−12.18	6.7	0
[39]	3,4,-$CH_2CH_2CH_2$	−11.93	6.8	-
		Compounds with $pIC_{50} \geq 7.0$**, but with a docking score above the threshold** (−11.75)		
[33]	3-CF_3	−11.54	7.0	10
[43]	3-CF_3-4-Cl	−11.39	7.0	1
		Parent compound		
[15]	H	−10.26	5.7	0

Table 5.2: Comparison of hits identified by the approach with compounds that have been experimentally assayed (Adams et al., 2014).

5.7 Discussion

In this section, we place our work in the context of machine learning approaches closely related to ours (Section 5.7.1) and discuss some directions for future development of the cyclic discovery process characteristic to drug design (Section 5.7.2).

5.7.1 Machine Learning Perspective

Active search with k-NN probabilistic model (Garnett et al., 2012) is a related approach with the problem setting similar to that of de novo design. A key distinction between the investigated problem setting and k-NN active search is in the requirement to discover structures from the whole domain. Garnett et al. (2012) assume that an extensional description in the form of a finite subset of the domain is explicitly given as input to the algorithm. In this work we require only an intensional description of the domain. For instance, for the domain of graphs with $n \in \mathbb{N}$ vertices, the intensional description is just that of the number of vertices, while the extensional one consists of a list of all graphs with n vertices. In many cases, considering intensional descriptions is much more promising because an algorithm with an extensional description of an exponentially large or uncountable search space can only consider small and often arbitrary subsets of this space. The second key distinction between k-NN active search and de novo design is in the assessment of their outcomes. In particular, both approaches try to find, as soon as possible, as many as possible target structures. However, k-NN active search is designed to only discover members of a target class and Algorithm 5.1 is designed to find members of distinct structural partitions of a target class. This is very useful in domains where there are numerous isofunctional structures and in which k-NN active search outputs structures from small number of structural partitions of a target class.

Recently, active search has been applied to a problem related to our cocktail construction task—interactive exploration of patterns in a cocktail dataset (Paurat et al., 2014). The difference between our setting and that of Paurat et al. (2014) is in the requirement to generate novel and previously unseen cocktails exhibiting a target property rather than searching for patterns in an existing dataset. In addition to this, active search has been applied to real-world problems where the search space is given by a single combinatorial graph, and a subset of its nodes is interesting (Wang et al., 2013). This is different from applications considered here for which the search space consists of all graphs of a given size.

As the investigated problem setting can be seen as a search in structured spaces, our approach is, with certain distinctions, closely related to structured output prediction (Tsochantaridis et al., 2004; Daumé III et al., 2009). In structured output prediction the goal is to find a mapping from an instance space to a 'structured' output space. A common approach is to find a joint scoring function, from the space of input–output pairs to the set of reals, and to predict the output structure which maximizes the scoring function for each test input. Finding a good scoring function can often be cast as a convex optimization problem with exponentially many constraints. It can be solved efficiently if the so-called *separation* and/or *decoding* sub-problems can be solved efficiently. One difference between the investigated setting and structured output prediction is in the assumption how input–output pairs are created. In particular, structured output prediction assumes that the provided outputs are optimal for the given inputs. In many de novo design problems, it is infeasible to find the best possible output for a given input. For de novo drug design this assumption implies that we would need to know the best molecule—from the space of all synthesizable molecules—with respect to different properties, such as binding affinity to specific protein sites. Moreover, as the decoding problem is designed assuming that the input–output pairs are optimal the greedy *argmax* approach to solving this problem does not incorporate exploration. As a result of this, similar to argmax search these methods generate structures from a very small number of structural partitions of the target class. Other differences are in the iterative nature of de novo design and in the hardness of the separation or decoding sub-problems that most structured output prediction approaches need to solve. Another related sub-problem is that of finding *preimages* (Weston et al., 2004) which is typically hard in the context of structured domains except for some special cases such as strings (Cortes et al., 2005; Giguère et al., 2015).

Related to the proposed approach are also methods for *interactive learning and optimization* as well as *Bayesian optimization*. Interactive learning and optimization methods implement a two-step iterative process in which an agent interacts with a user until a satisfactory solution is obtained. Some well-known interactive learning and optimization methods tackle problems in information retrieval (Yue and Joachims, 2009; Shivaswamy and Joachims,

2012) and reinforcement learning (Wilson et al., 2012; Jain et al., 2013). However, these methods are only designed to construct a single output from the domain of real-valued vectors and can not be directly applied to structured domains. Bayesian optimization (Brochu et al., 2010; Shahriari et al., 2015), on the other hand, is an approach to sequential optimization of an expensive, black-box, real-valued objective. Rather than seeking a set of high-quality items, Bayesian optimization focuses on finding the single highest-scoring point in the domain. We, in contrast, consider discrete labels and wish to maximize the number of diverse targets found in an intensionally specified structured space. In drug design, this emphasis on exploring all parts of the search space is known as *scaffold-hopping* (Schneider and Fechner, 2005) and it is related to the problem of attrition (Schneider and Schneider, 2016). Namely, in order to address this problem it is not sufficient to search for a molecule with the highest activity level as it can be toxic or bind to an undesired protein in addition to the target protein. If attrition is to be reduced an algorithm needs to find a number of structurally different molecules binding to a target protein. As our approach achieves a large structural variety of discovered targets, it has a potential to tackle this difficult problem.

5.7.2 De Novo Drug Design Perspective

In drug discovery, de novo design refers to a family of approaches for finding novel molecules with desired properties from an intensionally specified chemical space of interest (Schneider and Fechner, 2005; Schneider, 2013). An algorithm from this family can be characterized by three core components: *i*) (adaptive) proposal generator, *ii*) scoring function, and *iii*) adaptation scheme that adapts/shrinks the search space by modifying the proposal generator using scores assigned to the previously generated compounds. A de novo design approach is assessed by the quality of the designed compounds, which depends on the ability of the algorithm to cope with the combinatorial complexity of the search space (Schneider and Fechner, 2005). This ability and, thus, the outcome of any de novo design approach, crucially depends on the adaptation scheme. As described earlier, our approach copes with the combinatorial complexity of the search space by focusing the search with a probabilistic surrogate of the binding affinity. Moreover, the focused search is iteratively refined by updating the surrogate model as we observe the binding affinity of the previously selected designs. The whole process is consistent and guaranteed not to perform arbitrarily bad. In particular, after at most polynomially many oracle queries our approach is guaranteed to sample from the posterior distribution over molecular structures that is defined by the best conditional model of the binding affinity from a family of such models provided as input to the algorithm. In contrast to the presented approach, adaptation schemes in de novo design are typically not driven by adaptive models/hypotheses and the achieved reductions in the search space do not come with any type of guarantee. More specifically, de novo design methods are stochastic processes that usually discover good candidates but there is no guarantee that any of these random processes will not become arbitrarily bad (i.e., fail to discover satisfactory lead candidates after at most polynomially many queries). Moreover, our empirical results (e.g., see Figure 5.6) indicate that our approach exhibits a fast learning rate and after several hundred oracle queries samples a model that approximates fairly well the in silico proxy of the binding affinity to the selected protein binding site.

Most in silico scoring functions used in de novo design are developed to approximate primary target constraints, that is, the binding affinity of a ligand to a target protein site (Schneider and Fechner, 2005; Schneider, 2013). In silico evaluation of any compound can be a challenging and computationally intensive task. For example, the docking oracle employed in this chapter takes approximately 20-25 CPU minutes to dock an individual molecule and typically involves the explicit docking of 3 000 to 7 000 conformations per molecule. This number of conformations is somewhat larger than that usually considered due to the presence of stereo-centers as well as exploring extended, slightly higher energy conformations. Receptor-based scoring functions typically employed in de novo design can be divided into three groups (Perola et al., 2004; Schneider and Fechner, 2005; Schneider, 2013): *i*) explicit force-field methods, *ii*) empirical scoring functions, and *iii*) knowledge-based scoring functions. The approaches with force-field scoring functions can be computationally expensive and discovered compounds are evaluated by approximating the binding energy (Nishibata and Itai, 1993; Rotstein and Murcko, 1993; Liu et al., 1999; Zhu et al., 2001; Pearlman and Murcko, 1993). The empirical scoring functions rely on a small set of known actives to train a regression model that weights individual ligand-receptor interactions types (Böhm, 1992; Clark et al., 1995; Murray et al., 1997; Wang et al., 2000). However, as only a small set of known actives is available beforehand such oracles tend to bias the discovery process toward structural components present in the set of known actives (Schneider and Fechner, 2005; Schneider, 2013). Knowledge-based evaluation oracles are based on statistical properties of ligand-receptor structures, that is, frequencies of interactions between all possible pairs of atoms (DeWitte and Shakhnovich, 1996; Ishchenko and Shakhnovich, 2002). Such oracles require only structural information to derive the interaction frequencies for all pairs of atoms and are known to be less biased than the empirical ones.

Access to compounds from an intensionally specified chemical space of interest is typically provided through proposal generators (also known as structure samplers). The existing compound generation procedures can be classified into two groups (Schneider and Fechner, 2005; Schneider, 2013): receptor and ligand based structure samplers. Proposal generators based on a particular receptor structure can provide additional information characteristic to a protein binding site. Several such approaches have been developed over the years with the prominent ones being: *i*) linking of fragments placed at key interaction sites of the receptor structure (Böhm, 1992; Clark et al., 1995; Wang et al., 2000), *ii*) growing of a fragment randomly selected from a set of possible initial fragments which have all been placed at interaction sites of the receptor using expert knowledge (Nishibata and Itai, 1993; DeWitte and Shakhnovich, 1996; Ishchenko and Shakhnovich, 2002), and *iii*) structure sampling where randomly selected fragments are assembled at the receptor with the help of molecular dynamics simulations (Liu et al., 1999; Zhu et al., 2001; Pearlman and Murcko, 1993; Goodford, 1985). Ligand based proposal generators are independent of the receptor structure and work by sampling atoms/fragments and connecting them using valence rules (Schneider and Fechner, 2005; Schneider, 2013). Atom-based samplers (Nishibata and Itai, 1991; Pearlman and Murcko, 1993; Todorov and Dean, 1997) are known to generate diverse compounds and span a large chemical space. This then increases the combinatorial complexity of the search space and makes the search for active compounds more difficult. Contrary to this, fragment based approaches (Pellegrini and Field, 2003; Böhm, 1992; Clark et al., 1995; Brown et al., 2004; Pierce et al., 2004; Gillet et al., 1993) can significantly reduce the size of the search space. The reduction is deemed meaningful (Schneider and Fechner, 2005; Schneider, 2013) when the used fragments are common structures found in a variety of known drug-like compounds. In our simulations, we take the latter approach and investigate a local neighborhood of a parent compound with relatively moderate activity level (Adams et al., 2014).

While the binding affinity is of primary concern for de novo design, equally important are secondary target constraints (Schneider and Fechner, 2005; Schneider, 2013; Vangrevelinghe and Ruedisser, 2007): absorption, distribution, metabolism, excretion, and toxicity (ADMET properties). Similar to binding affinity, ADMET properties can also be approximated in silico (Vangrevelinghe and Ruedisser, 2007; van de Waterbeemd and Gifford, 2003; van de Waterbeemd and Rose, 2008). Previous attempts to approximate secondary target constraints in silico include approaches based on QSAR analysis (Vangrevelinghe and Ruedisser, 2007; van de Waterbeemd and Rose, 2008) and/or protein modelling (Vangrevelinghe and Ruedisser, 2007). Thus, any effective drug design algorithm needs to be successful across multiple different criteria. While the proof of concept presented in this chapter focuses on binding affinity only, our approach can be easily adapted to multiple objectives. More specifically, instead of oracles with binary feedback we could employ an oracle providing binary vectors as feedback. Such vectors could, for example, have a component for each of the ADMET properties and the kernel function on such space of properties can be given by a simple dot product between the binary vectors. The desired property class from which the algorithm would aim at sampling from is given by the vector of all ones. We consider this to be a promising avenue for future work.

Wider application of de novo design algorithms has been hindered by two main shortcomings (Vangrevelinghe and Ruedisser, 2007): synthetic accessibility of the designed compounds and the insufficient reliability of the affinity approximations. In particular, proposal generators that only consider valence rules while proposing candidate compounds are not sufficient to ensure generation of stable and synthetically accessible molecules (Schneider and Fechner, 2005; Schneider, 2013). Several different approaches have been developed for tackling the problem of synthetic accessibility of compounds (Schneider, 2013; Vangrevelinghe and Ruedisser, 2007). In Section 5.5.2, we have taken one such approach by substituting fragments consisting of functional groups in place of hydrogen atoms at specific attachment points of the parent compound. As the parent compound is synthetically accessible, it is expected that the substitutions mimicking chemical reactions would yield synthetically accessible designs, as well. In addition to this, we have incorporated filters into our proposal generator to increase the drug-likeness of the proposals and their synthetic accessibility. The filters consist of some of Lipinski's rules (Lipinski et al., 2001) and a constraint preventing undesirable (from the perspective of synthetic accessibility) placements of hydroxyl groups. Moreover, as lead compounds with large molecular mass are likely to reduce the chance of drug reaching the market (Vangrevelinghe and Ruedisser, 2007), we enhance the Lipinski's constraint on molecular mass by incorporating a stochastic stopping criterion into our proposal generator that favors lighter compounds (e.g., see the mass-dependent stopping criterion in line 12 of Algorithm 5.5). Further improvements to structure sampling are possible with the addition of information from available sets of actual chemical reactions (Murray et al., 1997; Schneider et al., 2000; Lewell et al., 1998; Vinkers et al., 2003). This type of additional information has the potential to generate a viable synthesis path together with a novel compound (i.e., recipe for derivation of any designed compound) and will be considered in future work.

5.8 Appendix

Dry

[1 | jagermeister ≥ 0.225 ? Dry : **go to** 2]
[2 | gin ≥ 0.465639 ? Dry : **go to** 3]
[3 | jackdaniels ≥ 0.138889 ? Dry : **go to** 4]
[4 | 151 proof rum ≥ 0.291666 ? Dry : **go to** 5]
[5 | vodka ≥ 0.437037 ? Dry : Not Dry]

Creamy

[1 | bailey's irish cream ≥ 0.03324 ? Creamy : **go to** 2]
[2 | creme de cacao ≥ 0.0059365 ? Creamy : **go to** 3]
[3 | milk ≥ 0.21495 ? Creamy : **go to** 4]
[4 | irish cream ≥ 0.006375 ? Creamy : **go to** 5]
[5 | cream ≥ 0.014754 ? Creamy : Not Creamy]

List of Figures

List of Tables

Notation

Chapter 2

$\mathcal{X}$	instance space
$\mathcal{Y}$	space of labels
ρ	Borel probability measure defined on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
$\rho_{\mathcal{X}}$	marginal probability measure defined on $\mathcal{X}$
X	data matrix with instances as columns
n	number of instances
d	dimension of the instance space
$\mathcal{H}$	reproducing kernel Hilbert space with kernel $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
$\mathcal{H}_X$	span of evaluation functions from $\mathcal{H}$, i.e., $\mathcal{H}_X = \operatorname{span}\left(\{k(x,\cdot) \mid x \in X\}\right)$
K	kernel matrix
$\mathcal{A}_s$	set of control points' placements along the s-th knowledge-based kernel principal component (p. 21)
$\mathcal{B}_s$	set of must-link and cannot-link constraints imposed along the s-th knowledge-based kernel principal component (p. 21)
$\mathcal{C}_s$	set of classification constraints imposed along the s-th knowledge-based kernel principal component (p. 21)
Υ	linear operator expressing hard knowledge-based constraints (p. 23)
A	diagonal matrix with $A_{ii} = 1$ when a coordinate placement is provided for instance x_i and $A_{ii} = 0$ otherwise (p. 23)
B	Laplacian matrix given by must-link and cannot-link constraints (p. 24)
C	diagonal matrix with $C_{ii} = 1$ when a label is provided for instance x_i and $C_{ii} = 0$ otherwise (p. 24)
H	matrix expressing the soft orthogonality constraint (p. 24)

Chapter 3

X	compact subset of a finite dimensional Euclidean space

Y	subset of the set of real numbers with labels in its interior
n	number of instances
m	number of greedy features
d	dimension of the instance space
ρ	Borel probability measure defined on a Euclidean space $Z = X \times Y$
ρ_X	marginal probability measure defined on X
$\mathbf{z}$	sample of n examples drawn independently from ρ (p. 52)
f_ρ	regression function of a measure ρ defined on Z (p. 53)
$\mathcal{E}_\rho$	expected squared error in the measure ρ defined on Z (p. 53)
$\mathcal{E}_\mathbf{z}$	mean squared error given by a sample $\mathbf{z}$ (p. 53)
$\mathcal{L}^2_\rho(X)$	Hilbert space of square integrable functions in a measure ρ
$\mathcal{C}(X)$	Banach space of continuous functions on X with the uniform norm
ϕ	ridge-wave basis function (p. 54)
$\mathrm{co}(S)$	convex hull of elements from a set S
$\overline{S}$	closure of a set S
$\mathcal{F}_\Theta$	set of ridge-wave functions defined on X (p. 54)
$\mathcal{F}$	hypothesis space defined as $\mathcal{F} = \overline{\mathrm{co}(\mathcal{F}_\Theta)}$ (p. 56)
$\mathcal{N}_\varepsilon(A;\ \lVert\cdot\rVert)$	ε-covering number of a set A in the metric space given by $\lVert\cdot\rVert$ (p. 57)

Chapter 4

$\mathcal{X}$	instance space
ρ	Borel probability measure defined on $\mathcal{X}$
X	data matrix with independent samples from ρ as columns
n	number of instances
d	dimension of the instance space
l, m	number of landmarks
$\mathcal{H}$	reproducing kernel Hilbert space with a Mercer kernel $h\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
H	kernel matrix
Z	set of landmarks defining the Nyström approximation of H
$H_{X \times Z}$	block in the kernel matrix H corresponding to kernel values defined by instances in X and Z
$\lVert\cdot\rVert_p$	Schatten p-norm of a symmetric and positive definite matrix (p. 91)
K	number of clusters in an instance of K-means clustering
$\phi(\cdot)$	K-means clustering potential (p. 93)
$\mathcal{P}$	centroid assignment function in K-means clustering (p. 93)
P	cluster indicator matrix (p. 93)
$U^\perp$	the dual matrix of a matrix U
C^*	optimal set of K-means centroids
H_K	optimal rank K approximation of a kernel matrix H (p. 92, 102)
$\phi(C^* \mid U)$	clustering potential defined by the projections of X and C^* onto

the subspace spanned by the columns of U (p. 102)

Chapter 5

$\mathcal{X}$	instance space
$\mathcal{Y}$	space of properties/labels
$\mathcal{O}$	evaluation oracle
$y^* \in \mathcal{Y}$	target property
B	number of oracle evaluations
$\mathcal{G}$	proposal generator
$\phi(x, y)$	sufficient statistics defined on $\mathcal{X} \times \mathcal{Y}$
$A(\theta \mid x)$	log-partition function of a conditional exponential family model (p. 129, 130)
$\mathcal{H}$	reproducing kernel Hilbert space with a tuple kernel $k\colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
$\Theta \subseteq \mathcal{H}$	parameter set specifying a family of conditional exponential family models (p. 130, 143)
ρ	Borel probability measure defined on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ (p. 129, 130, 143)
$\rho_{\mathcal{X}}$	marginal probability measure defined on $\mathcal{X}$
g	transition kernel of a Markov chain with stationary distribution $\rho_{\mathcal{X}}$ (p. 129)
$\mathrm{supp}(\cdot)$	support of a probability distribution
$\mathcal{L}^2_{\mu}(\mathcal{X} \times \mathcal{Y})$	Hilbert space of square integrable functions in a measure μ
$H(p \mid q)$	conditional entropy of a conditional probability density function p with respect to a marginal probability density function q (p. 131)
$L(\theta)$	log-loss function of a parameter vector $\theta \in \Theta$ (p. 143)
$\mathcal{D}_t$	data available at iteration t (p. 144)
$L(\theta \mid \mathcal{D}_t)$	empirical loss of a parameter vector given observed data $\mathcal{D}_t$ (p. 144)
k	kernel function defined on $\mathcal{X} \times \mathcal{Y}$ (p. 151)
$k_{\mathcal{X}}$	kernel function defined on $\mathcal{X}$ (p. 151)

Bibliography

Adams, J., Anderson, E. C., Blackham, E. E., Chiu, Y. W. R., Clarke, T., Eccles, N., Gill, L. A., Haye, J. J., Haywood, H. T., Hoenig, C. R., Kausas, M., Le, J., Russell, H. L., Smedley, C., Tipping, W. J., Tongue, T., Wood, C. C., Yeung, J., Rowedder, J. E., Fray, M. J., McInally, T., and Macdonald, S. J. F. (2014). Structure activity relationships of α_v integrin antagonists for pulmonary fibrosis by variation in aryl substituents. *ACS Medicinal Chemistry Letters*, 5(11):1207–1212.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In Petrov, B. N. and Csaki, F., editors, *Second International Symposium on Information Theory*, pages 267–281. Akadémiai Kiado.

Akrour, R., Schoenauer, M., and Sebag, M. (2012). APRIL: Active preference learning-based reinforcement learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 116–131. Springer-Verlag.

Akrour, R., Schoenauer, M., Sebag, M., and Souplet, J.-C. (2014). Programming by feedback. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31th International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1503–1511. PMLR.

Alaoui, A. E. and Mahoney, M. W. (2015). Fast randomized kernel ridge regression with statistical guarantees. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 775–783. Curran Associates Inc.

Aldous, D. (1983). Random walks on finite groups and rapidly mixing Markov chains. *Séminaire de Probabilités de Strasbourg*, 17:243–297.

Aloise, D., Deshpande, A., Hansen, P., and Popat, P. (2009). NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248.

Altun, Y. and Smola, A. (2006). Unifying divergence minimization and statistical inference via convex duality. In Lugosi, G. and Simon, H. U., editors, *Proceedings of the 19th Annual Conference on Computational Learning Theory*, pages 139–153. Springer Berlin Heidelberg.

Altun, Y. and Smola, A. (2007). Estimating conditional densities of structured outputs in RKHS. In Bakir, G., Hofmann, T., Schölkopf, B., Smola, A. J., Taskar, B., and Vishwanathan, S., editors, *Machine Learning with Structured Outputs*. MIT Press.

Altun, Y., Smola, A. J., and Hofmann, T. (2004). Exponential families for conditional random fields. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 2–9. AUAI Press.

Anderson, N. A., Campbell, I. B., Campbell-Crawford, M. H. J., P.Hancock, A., Lemma, S., Macdonald, S. J. F., Pritchard, J. M., and A.Procopiou, P. (2016a). Naphthyridine derivatives as $\alpha_v\beta_6$ integrin antagonists for the treatment of fibrotic diseases.

Anderson, N. A., Campbell, I. B., Campbell-Crawford, M. H. J., P.Hancock, A., Lemma, S., Macdonald, S. J. F., Pritchard, J. M., A.Procopiou, P., and S.Swanson (2016b). Novel compounds.

Andersson, S., Armstrong, A., Björe, A., Bowker, S., Chapman, S., Davies, R., Donald, C., Egner, B., Elebring, T., Holmqvist, S., Inghardt, T., Johannesson, P., Johansson, M., Johnstone, C., Kemmitt, P., Kihlberg, J., Korsgren, P., Lemurell, M., Moore, J., Pettersson, J. A., Pointon, H., Pontén, F., Schofield, P., Selmi, N., and Whittamore, P. (2009). Making medicinal chemistry more effective—application of lean sigma to improve processes, speed and quality. *Drug Discovery Today*, 14(11/12):598–604.

Andrews, C., Endert, A., and North, C. (2010). Space to think: Large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 55–64. ACM.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43.

Anthony, M. and Bartlett, P. L. (2009). *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1st edition.

Arbenz, P. (2012). *Lecture Notes on Solving Large Scale Eigenvalue Problems*, pages 77–93. ETH Zürich.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404.

Arthur, D. and Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.

Azuma, K. (1967). Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3):357–367.

Bach, F. (2007). Active learning for misspecified generalized linear models. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 65–72. MIT Press.

Bach, F. R. (2013). Sharp analysis of low-rank kernel matrix approximations. In *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 185–209. PMLR.

Bach, F. R. and Jordan, M. I. (2005). Predictive low-rank decomposition for kernel methods. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 33–40. ACM.

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3).

Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12(1):149–198.

Belabbas, M. A. and Wolfe, P. J. (2009). Spectral methods in machine learning: New strategies for very large datasets. *Proceedings of the National Academy of Sciences of the USA*, 106(2):369–374.

Belkin, M. and Niyogi, P. (2002). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Bertinet, A. and Agnan, T. C. (2004). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers.

Beygelzimer, A., Dasgupta, S., and Langford, J. (2009). Importance weighted active learning. In *Proceedings of the 26th International Conference on Machine Learning*, pages 49–56. ACM.

Bochner, S. (1932). Vorlesungen über Fouriersche Integrale. In *Akademische Verlagsgesellschaft*, volume 12 of *Mathematik und ihre Anwendungen*.

Böhm, H. J. (1992). LUDI: Rule-based automatic design of new substituents for enzyme inhibitor leads. *Journal of Computer-Aided Molecular Design*, 6(6):593–606.

Bondy, A. J. and Murty, U. S. R. (2008). *Graph Theory*. Springer London.

Boothby, W. M. (1986). *An introduction to differentiable manifolds and Riemannian geometry*. Academic Press.

Borgwardt, K. M. (2007). *Graph kernels*. PhD thesis, Ludwig Maximilians University Munich.

Boutsidis, C., Drineas, P., and Mahoney, M. W. (2009). Unsupervised feature selection for the K-means clustering problem. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 153–161. Curran Associates Inc.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Brochu, E., Cora, V. M., and de Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Broekens, J., Cocx, T., and Kosters, W. A. (2006). Object-centered interactive multi-dimensional scaling: Ask the expert. In *In Proceedings of the Eighteenth Belgium-Netherlands Conference on Artificial Intelligence*.

Brown, N., McKay, B., Gilardoni, F., and Gasteiger, J. (2004). A graph-based genetic algorithm and its application to the multiobjective evolution of median molecules. *Journal of Chemical Information and Computer Sciences*, 44(3):1079–1087.

Buja, A., Swayne, D. F., Littman, M. L., Dean, N., Hofmann, H., and Chen, L. (2008). Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*.

Bunch, J. R., Nielsen, C. P., and Sorensen, D. (1978). Rank-one modification of the symmetric eigenproblem. *Numerische Mathematik*, 31(1):31–48.

Burges, C. J. C. (1999). Geometry and invariance in kernel based methods. In Schölkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in Kernel Methods*, pages 89–116. MIT Press.

Cacciari, B., Crepaldi, P., Federico, S., and Spalluto, G. (2009). Recent developments in the field of non peptidic $\alpha_v\beta_3$ antagonists. *Frontiers in Medicinal Chemistry*.

Callahan, E. and Koenemann, J. (2000). A comparative usability evaluation of user interfaces for online product catalog. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pages 197–206. ACM.

Cameron, P. J. (1998). *Introduction to Algebra*. Oxford University Press.

Carl, B. and Stephani, I. (1990). *Entropy, Compactness, and the Approximation of Operators*. Cambridge Tracts in Mathematics. Cambridge University Press.

Chapelle, O., Schölkopf, B., and Zien, A., editors (2006). *Semi-Supervised Learning*. MIT Press, 1st edition.

Cipra, B. A. (2000). The best of the 20th century: Editors name top 10 algorithms. *Society for Industrial and Applied Mathematics News*.

Clark, D. E., Frenkel, D., Levy, S. A., Li, J., Murray, C. W., Robson, B., Waszkowycz, B., and Westhead, D. R. (1995). PRO-LIGAND: An approach to de novo molecular design. 1. Application to the design of organic molecules. *Journal of Computer-Aided Molecular Design*, 9(1):13–32.

Cohen, M. B., Elder, S., Musco, C., Musco, C., and Persu, M. (2015). Dimensionality reduction for K-means clustering and low rank approximation. In *Proceedings of the Forty-seventh Annual ACM Symposium on Theory of Computing*, pages 163–172. ACM.

Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. MIT Press, 3rd edition.

Cortes, C., Mansour, Y., and Mohri, M. (2010). Learning bounds for importance weighting. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 442–450. Curran Associates Inc.

Cortes, C., Mohri, M., and Weston, J. (2005). A general regression technique for learning transductions. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 153–160. ACM.

Cox, T. F. and Cox, M. A. A. (2000). *Multidimensional Scaling*. Chapman and Hall/CRC, 2nd edition.

Cucker, F. and Smale, S. (2002). On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39:1–49.

Dai, B., Xie, B., He, N., Liang, Y., Raj, A., Balcan, M.-F., and Song, L. (2014). Scalable kernel methods via doubly stochastic gradients. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 3041–3049. Curran Associates, Inc.

Daumé III, H., Langford, J., and Marcu, D. (2009). Search-based structured prediction. *Machine Learning*, 75(3):297–325.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

DeWitte, R. S. and Shakhnovich, E. I. (1996). SmoG: De novo design method based on simple, fast, and accurate free energy estimates. 1. Methodology and supporting evidence. *Journal of the American Chemical Society*, 118(47):11733–11744.

Diestel, R. (1997). *Graph Theory*. Number 173 in Graduate Texts in Mathematics. Springer.

Dietterich, T. G. (2003). Machine learning. *Nature Encyclopedia of Cognitive Science*.

Dietterich, T. G. and Horvitz, E. J. (2015). Rise of concerns about AI: Reflections and directions. *Communications of the ACM*, 58(10):38–40.

Ding, C. and He, X. (2004). K-means clustering via principal component analysis. In *Proceedings of the 21st International Conference on Machine Learning*, pages 29–37. ACM.

Dinuzzo, F. and Schölkopf, B. (2012). The representer theorem for Hilbert spaces: A necessary and sufficient condition. In P. Bartlett, F. C. N. Pereira, C. J. C. B. L. B. and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 189–196. Curran Associates Inc.

Dixon, J. D. and Wilf, H. S. (1983). The random selection of unlabeled graphs. *Journal of Algorithms*, 4(3):205–213.

Donahue, M. J., Darken, C., Gurvits, L., and Sontag, E. (1997). Rates of convex approximation in non-Hilbert spaces. *Constructive Approximation*, 13(2):187–220.

Dong, X., Hudson, N. E., Lu, C., and Springer, T. A. (2014). Structural determinants of integrin β-subunit specificity for latent tgf-β. *Nature Structural & Molecular Biology*, 21(12):1091–1096.

Drineas, P., Kannan, R., and Mahoney, M. W. (2006). Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):132–157.

Drineas, P. and Mahoney, M. W. (2005). On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(12):2153–2175.

Dudík, M. and Schapire, R. E. (2006). Maximum entropy distribution estimation with generalized regularization. In Lugosi, G. and Simon, H. U., editors, *Proceedings of the 19th Annual Conference on Computational Learning Theory*, pages 123–138. Springer Berlin Heidelberg.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Endert, A., Han, C., Maiti, D., House, L., Leman, S., and North, C. (2011). Observation-level interaction with statistical models for visual analytics. In *IEEE Conference on Visual Analytics Science and Technology*, pages 121–130.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Fine, S. and Scheinberg, K. (2002). Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264.

Forsythe, G. E. and Golub, G. H. (1965). On the stationary values of a second-degree polynomial on the unit sphere. *Journal of the Society for Industrial and Applied Mathematics*, 13(4):1050–1068.

Friedman, J. H. (2000). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232.

Frobenius, G. F. (1912). Über Matrizen aus nicht negativen Elementen. *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften.*

Gander, W. (1980). Least squares with a quadratic constraint. *Numerische Mathematik*, 36(3):291–307.

Gander, W., Golub, G., and von Matt, U. (1989). A constrained eigenvalue problem. *Linear Algebra and its Applications*, 114-115:815–839.

Garnett, R., Krishnamurthy, Y., Xiong, X., Schneider, J., and Mann, R. P. (2012). Bayesian optimal active search and surveying. In *Proceedings of the 29th International Conference on Machine Learning*. Omnipress.

Gärtner, T. (2003). A survey of kernels for structured data. *SIGKDD Explorations*, 5(1):49–58.

Gärtner, T. (2005). *Kernels for Structured Data.* PhD thesis, Universität Bonn.

Gärtner, T., Flach, P. A., and Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory*, pages 129–143.

Gelfand, I. M. and Fomin, S. V. (1963). *Calculus of variations.* Prentice-Hall Inc.

Genton, M. G. (2002). Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312.

Giguère, S., Rolland, A., Laviolette, F., and Marchand, M. (2015). Algorithms for the hard pre-image problem of string kernels and the general problem of string prediction. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2021–2029. PMLR.

Gillet, V., Johnson, A. P., Mata, P., Sike, S., and Williams, P. (1993). SPROUT: A program for structure generation. *Journal of Computer-Aided Molecular Design*, 7(2):127–153.

Gittens, A. and Mahoney, M. W. (2016). Revisiting the Nyström method for improved large-scale machine learning. *Journal Machine Learning Research*, 17(117):1–65.

Golub, G. H. and van Loan, C. F. (1996). *Matrix Computations.* Johns Hopkins University Press.

Goodford, P. J. (1985). A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *Journal of Medicinal Chemistry*, 28(7):849–857.

Gu, M. and Eisenstat, S. C. (1994). A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem. *SIAM Journal on Matrix Analysis and Applications*, 15(4):1266–1276.

Guruswami, V. (2000). Rapidly mixing Markov chains: A comparison of techniques (survey). Technical report, MIT.

Halgren, T. A. (1999). MMFF VI. MMFF94s option for energy minimization studies. *Journal of Computational Chemistry*, 20(7):720–729.

Ham, J., Lee, D. D., Mika, S., and Schölkopf, B. (2004). A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the 21st International Conference on Machine Learning*, pages 47–56. ACM.

Hartenfeller, M., Eberle, M., Meier, P., Nieto-Oberhuber, C., Altmann, K.-H., Schneider, G., Jacoby, E., and Renner, S. (2011). A collection of robust organic synthesis reactions for in silico molecule design. *Journal of Chemical Information and Modeling*, 51(12):3093–3098.

Hartenfeller, M., Zettl, H., Walter, M., Rupp, M., Reisen, F., Proschak, E., Weggen, S., Stark, H., and Schneider, G. (2012). Dogs: Reaction-driven de novo design of bioactive compounds. *PLOS Computational Biology*, 8(2):1–12.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning.* Springer Series in Statistics. Springer New York Inc.

Hatley, R. J. D., Macdonald, S. J. F., Slack, R. J., Le, J., Ludbrook, S. B., and Lukey, P. T. (2017). An α_v-RGD integrin inhibitor toolbox: Drug discovery insight, challenges and opportunities. *Angewandte Chemie International Edition*, 57(13):3289–3321.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441.

Huber, M. (1998). Exact sampling and approximate counting techniques. In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*, pages 31–40. ACM.

Ishchenko, A. V. and Shakhnovich, E. I. (2002). SMall Molecule Growth 2001 (SMoG2001): An improved knowledge-based scoring function for protein-ligand interactions. *Journal of Medicinal Chemistry*, 45(13):2770–2780.

Izenman, A. J. (2008). Linear discriminant analysis. In *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*, pages 237–280. Springer New York.

Jain, A., Wojcik, B., Joachims, T., and Saxena, A. (2013). Learning trajectory preferences for manipulators via iterative improvement. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 575–583. Curran Associates Inc.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical Review*, 106:620–630.

Jeong, D. H., Ziemkiewicz, C., Fisher, B. D., Ribarsky, W., and Chang, R. (2009). iPCA: An interactive system for PCA-based visual analytics. *Proceedings of the 11th Eurographics/IEEE-VGTC Conference on Visualization*, 28(3):767–774.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. In *Proceedings of the 20th International Conference on Machine Learning*, pages 290–297. AAAI Press.

Jolliffe, I. T. (1986). *Principal Component Analysis.* Springer-Verlag.

Jones, L. K. (1992). A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *Annals of Statistics*, 20(1):608–613.

Kakade, S. M., Kanade, V., Shamir, O., and Kalai, A. T. (2011). Efficient learning of generalized linear and single index models with isotonic regression. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 927–935. Curran Associates, Inc.

Kalai, A. T. and Sastry, R. (2009). The isotron algorithm: High-dimensional isotonic regression. In *Proceedings of the 22nd Annual Conference on Computational Learning Theory.*

Kanamori, T. and Shimodaira, H. (2003). Active learning algorithm using the maximum weighted log-likelihood estimator. *Journal of Statistical Planning and Inference*, 116(1):149–162.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. (2002). A local search approximation algorithm for K-means clustering. In *Proceedings of the Eighteenth Annual Symposium on Computational Geometry*, pages 10–18. ACM.

Keerthi, S., Sindhwani, V., and Chapelle, O. (2007). An efficient method for gradient-based adaptation of hyperparameters in SVM models. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 673–680. MIT Press.

Kolmogorov, A. N. and Tikhomirov, V. M. (1959). ε-entropy and ε-capacity of sets in function spaces. *Uspehi Matematicheskikh Nauk*, 14(2).

Kulis, B., Sustik, M., and Dhillon, I. (2006). Learning low-rank kernel matrices. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 505–512. ACM.

Kumar, S., Mohri, M., and Talwalkar, A. (2012). Sampling methods for the Nyström method. *Journal of Machine Learning Research*, 13(1):981–1006.

Le, Q., Sarlos, T., and Smola, A. (2013). Fastfood – approximating kernel expansions in loglinear time. In Dasgupta, S. and McAllester, D., editors, *30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 244–252. PMLR.

Le, Q. V., Smola, A. J., and Canu, S. (2005). Heteroscedastic gaussian process regression. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 489–496. ACM.

Leman, S., House, L., Maiti, D., Endert, A., and North, C. (2013). Visual to parametric interaction (V2PI). *PLOS ONE*, 8(3):1–12.

Lewell, X. Q., Judd, D. B., Watson, S. P., and Hann, M. M. (1998). RECAPRetrosynthetic combinatorial analysis procedure: A powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *Journal of Chemical Information and Computer Sciences*, 38(3):511–522.

Ley, K., Rivera-Nieves, J., Sandborn, W. J., and Shattil, S. (2016). Integrin-based therapeutics: Biological basis, clinical use and new drugs. *Nature Reviews Drug Discovery*, 15(3):173–183.

Li, C., Jegelka, S., and Sra, S. (2016). Fast DPP sampling for Nyström with application to kernel methods. In *Proceedings of the 33nd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2061–2070. PMLR.

Li, R.-C. (1993). Solving secular equations stably and efficiently. *EECS Department, University of California, Berkeley*, (UCB/CSD-94-851).

Lipinski, C. A., Lombardo, F., Dominy, B. W., and Feeney, P. J. (2001). Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, 46(1):3–26.

Liu, H., Duan, Z., Luo, Q., and Shi, Y. (1999). Structure-based ligand design by dynamically assembling molecular building blocks at binding site. *Proteins*, 36:462––470.

Liu, Y.-M., Nepali, K., and Liou, J.-P. (2017). Idiopathic pulmonary fibrosis: Current status, recent progress, and emerging targets. *Journal of Medicinal Chemistry*, 60(2):527–553.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.

Luenberger, D. G. (1973). *Introduction to Linear and Nonlinear Programming*. Addison-Wesley.

Lütkepohl, H. (1997). *Handbook of Matrices*. Wiley, 1st edition.

Lázaro-Gredilla, M., Candela, J. Q., Rasmussen, C. E., and Figueiras-Vidal, A. R. (2010). Sparse spectrum gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881.

Mackay, D. J. C. (1997). Gaussian processes. *NIPS Tutorial – A Replacement for Supervised Neural Networks?*

Malik, D. S., Mordeson, J. M., and Sen, M. K. (1997). *Fundamentals of Abstract Algebra*. McGraw–Hill.

Maltsev, O. V., Marelli, U. K., Kapp, T. G., di Leva, F. S., di Maro, S., Nieberler, M., Reuning, U., Scwaiger, M., Novellino, E., Marinelli, L., and Kessler, H. (2016). Stable peptides instead of stapled peptides: Highly potent $\alpha_v\beta_6$ selective integrin ligands. *Angewandte Chemie International Edition*, 55(4):1535–1539.

Mason, L., Baxter, J., Bartlett, P. L., and Frean, M. (2000). Functional gradient techniques for combining hypotheses. In Smola, A., Bartlett, P., Schölkopf, B., and Schuurmans, D., editors, *Advances in large margin classifiers*, pages 221–246. MIT Press.

Matérn, B. (1986). *Spatial variation*, volume 36 of *Lecture notes in statistics*. Springer-Verlag.

Mayer, S., Ullrich, T., and Vybiral, J. (2015). Entropy and sampling numbers of classes of ridge functions. *Constructive Approximation*, 42(2):231–264.

McGann, M. (2011). FRED pose prediction and virtual screening accuracy. *Journal of Chemical Information and Modeling*, 51(3):578–596.

Mengersen, K. L. and Tweedie, R. L. (1996). Rates of convergence of the Hastings and Metropolis algorithms. *Annals of Statistics*, 24(1):101–121.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 83(559):69–70.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.

Meyn, S. P. and Tweedie, R. L. (2009). *Markov Chains and Stochastic Stability*. Cambridge University Press, 2nd edition.

Micchelli, C. A., Xu, Y., and Zhang, H. (2006). Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667.

Millard, M., Odde, S., and Neamati, N. (2011). Integrin targeted therapeutics. *Theranostics*, 17:154–188.

Mirsky, L. (1960). Symmetric gauge functions and unitarily invariant norms. *Quaterly Journal of Mathematics, Oxford II. Series*, 11:50–59.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Inc., 1st edition.

Murray, C. W., Clark, D. E., Auton, T. R., Firth, M. A., Li, J., Sykes, R. A., Waszkowycz, B., Westhead, D. R., and Young, S. C. (1997). PRO_SELECT: Combining structure-based drug design and combinatorial chemistry for rapid lead discovery. 1. Technology. *Journal of Computer-Aided Molecular Design*, 11:193–207.

Nishibata, Y. and Itai, A. (1991). Automatic creation of drug candidate structures based on receptor structure. Starting point for artificial lead generation. *Tetrahedron*, 47(43):8985–8990.

Nishibata, Y. and Itai, A. (1993). Confirmation of usefulness of a structure construction program based on three-dimensional receptor structure for rational lead generation. *Journal of Medicinal Chemistry*, 36(20):2921–2928.

Nyström, E. J. (1930). Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*, 54(1):185–204.

Oglic, D., Garnett, R., and Gärtner, T. (2014a). Learning to construct novel structures. In *NIPS Workshop on Discrete and Combinatorial Problems in Machine Learning*.

Oglic, D., Garnett, R., and Gärtner, T. (2017). Active search in intensionally specified structured spaces. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2443–2449.

Oglic, D. and Gärtner, T. (2016). Greedy feature construction. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 3945–3953. Curran Associates, Inc.

Oglic, D. and Gärtner, T. (2017). Nyström method with kernel K-means++ samples as landmarks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2652–2660. PMLR.

Oglic, D., Oatley, S. A., Macdonald, S. J. F., Mcinally, T., Garnett, R., Hirst, J. D., and Gärtner, T. (2018). Active search for computer-aided drug design. *Molecular Informatics*, 37(1-2):1700130.

Oglic, D., Paurat, D., and Gärtner, T. (2014b). Interactive knowledge-based kernel PCA. In Calders, T., Esposito, F., Hüllermeier, E., and Meo, R., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 501–516. Springer Berlin Heidelberg.

Paurat, D., Garnett, R., and Gärtner, T. (2013a). Constructing cocktails from a cocktail map. In *NIPS Workshop on Constructive Machine Learning*.

Paurat, D., Garnett, R., and Gärtner, T. (2014). Interactive exploration of larger pattern collections: A case study on a cocktail dataset. In *Proceedings of KDD IDEA*.

Paurat, D. and Gärtner, T. (2013). InVis: A tool for interactive visual data analysis. In *European Conference on Machine Learning and Knowledge Discovery in Databases*.

Paurat, D., Oglic, D., and Gärtner, T. (2013b). Supervised PCA for interactive data analysis. In *Proceedings of the 2nd NIPS Workshop on Spectral Learning*.

Pearlman, D. A. and Murcko, M. A. (1993). CONCEPTS: New dynamic algorithm for de novo drug suggestion. *Journal of Computational Chemistry*, 4:1184–1193.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine and Journal of Science*, 2:559–572.

Pellegrini, E. and Field, M. J. (2003). Development and testing of a de novo drug-design algorithm. *Journal of Computer-Aided Molecular Design*, 17(10):621–641.

Perola, E., Walters, W. P., and Charifson, P. S. (2004). A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance. *Proteins: Structure, Function, and Bioinformatics*, 56(2):235–249.

Pierce, A. C., Rao, G., and Bemis, G. W. (2004). BREED: Generating novel inhibitors through hybridization of known ligands. Application to CDK2, P38, and HIV protease. *Journal of Medicinal Chemistry*, 47(11):2768–2775.

Propp, J. G. and Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. In *Proceedings of the Seventh International Conference on Random Structures and Algorithms*, pages 223–252. John Wiley & Sons Inc.

Rahimi, A. and Recht, B. (2008a). Random features for large-scale kernel machines. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc.

Rahimi, A. and Recht, B. (2008b). Uniform approximation of functions with random bases. In *46th Annual Allerton Conference on Communication, Control, and Computing*, pages 555–561. IEEE.

Rahimi, A. and Recht, B. (2009). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 21*, pages 1313–1320. Curran Associates, Inc.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press.

Reed, N. I., Jo, H., Chen, C., Tsujino, K., Arnold, T. D., DeGrado, W. F., and Sheppard, D. (2015). The $\alpha_v\beta_1$ integrin plays a critical in vivo role in tissue fibrosis. *Science Translational Medicine*, 7(288):288ra79.

Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer-Verlag New York Inc.

Rockafellar, R. T. (1966). Extension of Fenchel' duality theorem for convex functions. *Duke Mathematical Journal*, 33(1):81–89.

Rogers, D. and Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754.

Rotstein, S. H. and Murcko, M. A. (1993). GroupBuild: A fragment-based method for de novo drug design. *Journal of Medicinal Chemistry*, 36(12):1700–1710.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Rowland, T. (2017). Manifold. From MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein. `http://mathworld.wolfram.com/Manifold.html`.

Rudin, W. (1991). *Functional Analysis*. International Series in Pure and Applied Mathematics. McGraw-Hill Inc., 2nd edition.

Scannell, J. W., Blanckley, A., Boldon, H., and Warrington, B. (2012). Diagnosing the decline in pharmaceutical R&D efficiency. *Nature Reviews Drug Discovery*, 11:191–200.

Schneider, G., editor (2013). *De Novo Molecular Design*. Wiley-VCH.

Schneider, G. and Fechner, U. (2005). Computer-based de novo design of drug-like molecules. *Nature Reviews Drug Discovery*, 4(8):649–663.

Schneider, G., Lee, M. L., Stahl, M., and Schneider, P. (2000). De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks. *Journal of Computer-Aided Molecular Design*, 14(5):487–494.

Schneider, P. and Schneider, G. (2016). De novo design at the edge of chaos. *Journal of Medicinal Chemistry*, 59(9):4077–4086.

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*, pages 416–426. Springer-Verlag.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* Adaptive Computation and Machine Learning. MIT Press.

Schölkopf, B., Smola, A. J., and Müller, K.-R. (1999). Advances in kernel methods. chapter Kernel Principal Component Analysis, pages 327–352. MIT Press.

Schönberg, I. J. (1938). Metric spaces and completely monotone functions. *Annals of Mathematics*, 39(4):811–841.

Settles, B. (2012). *Synthesis Lectures on Artificial Intelligence and Machine Learning.* Morgan & Claypool Publishers.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2015). Taking the human out of the loop: A review of Bayesian optimization. Technical report, Universities of Harvard, Oxford, Toronto, and Google DeepMind.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104:148–175.

Shearer, C. (2000). The CRISP-DM model: The new blueprint for data mining. *Journal of data warehousing*, 5(4):13–22.

Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. (2011). Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244.

Shivaswamy, P. and Joachims, T. (2012). Online structured prediction via coactive learning. In *Proceedings of the 29th International Conference on Machine Learning*, pages 59–66. Omnipress.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman & Hall.

Smola, A. J. and Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 911–918. Morgan Kaufmann Publishers Inc.

Smola, A. J. and Vishwanathan, S. V. N. (2010). *Introduction to machine learning.* Cambridge University Press.

Stein, M. (1999). *Interpolation of Spatial Data: Some Theory for Kriging.* Springer Series in Statistics. Springer-Verlag New York Inc., 1st edition.

Sugiyama, M., Krauledat, M., and Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005.

Tenenbaum, J. B., Silva, V. D., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Tesch, M., Schneider, J., and Choset, H. (2013). Expensive function optimization with stochastic binary outcomes. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1283–1291. PMLR.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society*.

Todorov, N. and Dean, P. (1997). Evaluation of a method for controlling molecular scaffold diversity in de novo ligand design. *Journal of Computer-Aided Molecular Design*, 11:175–192.

Torgo, L. (accessed September 22, 2016). Repository with regression data sets. `http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html`.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the 21st International Conference on Machine Learning*, pages 104–112. ACM.

Tukey, J. (1977). *Exploratory Data Analysis*. Addison-Wesley series in behavioral science. Addison-Wesley.

Tukey, J. W. (1974). Mathematics and the picturing of data. In *Proceedings of the international congress of mathematicians*, volume 2, pages 523–532.

van de Waterbeemd, H. and Gifford, E. (2003). ADMET in silico modelling: Towards prediction paradise? *Nature Reviews Drug Discovery*, 2(3):192–204.

van de Waterbeemd, H. and Rose, S. (2008). Quantitative approaches to structure–activity relationships. In *The Practice of Medicinal Chemistry*, pages 491–513. Academic Press, 3rd edition.

Vangrevelinghe, E. and Ruedisser, S. (2007). Computational approaches for fragment optimization. *Current Computer-Aided Drug Design*, 3.

Vapnik, V. (1982). *Estimation of Dependences Based on Empirical Data*. Springer-Verlag New York Inc.

Varshney, L. R., Pinel, F., Varshney, K. R., Bhattacharjya, D., Schörgendorfer, A., and Chee, Y. (2013). A big data approach to computational creativity. *arXiv preprint arXiv:1311.1213*.

Vembu, S., Gärtner, T., and Boley, M. (2009). Probabilistic structured predictors. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 557–564. AUAI Press.

Vinkers, H. M., de Jonge, M. R., Daeyaert, F. F. D., Heeres, J., Koymans, L. M. H., van Lenthe, J. H., Lewi, P. J., Timmerman, H., Van Aken, K., and Janssen, P. A. J. (2003). SYNOPSIS: SYNthesize and OPtimize system in silico. *Journal of Medicinal Chemistry*, 46(13):2765–2773.

Wahba, G. (1990). *Spline models for observational data*. Society for Industrial and Applied Mathematics.

Walder, C., Henao, R., Mørup, M., and Hansen, L. K. (2010). Semi-supervised kernel PCA. *arXiv preprint arXiv:1008.1398*.

Wang, R., Gao, Y., and Lai, L. (2000). LigBuilder: A multi-purpose program for structure-based drug design. *Molecular modeling annual*, 6(7):498–516.

Wang, X., Garnett, R., and Schneider, J. (2013). Active search on graphs. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 731–738. ACM.

Weidmann, J. (1980). *Linear operators in Hilbert spaces*. Springer-Verlag.

Weinberger, K. Q. and Saul, L. K. (2004). Unsupervised learning of image manifolds by semidefinite programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 988–995. IEEE Computer Society.

Welch, B. L. (1947). The generalization of student's problem when several different population variances are involved. *Biometrika*, 34(1/2):28–35.

Weston, J., Schölkopf, B., and Bakir, G. H. (2004). Learning to find pre-images. In Thrun, S., Saul, L. K., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*, pages 449–456. MIT Press.

Weyl, H. (1912). Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Mathematische Annalen*, 71:441–479.

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, 50:1–25.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Williams, C. K. I. and Seeger, M. (2001). Using the Nyström method to speed up kernel machines. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press.

Wilson, A., Fern, A., and Tadepalli, P. (2012). A Bayesian approach for policy learning from trajectory preference queries. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1133–1141. Curran Associates Inc.

Woltosz, W. S. (2012). If we designed airplanes like we design drugs... *Journal of Computer-Aided Molecular Design*, 26(1):159–163.

Wormald, N. C. (1987). Generating random unlabelled graphs. *SIAM Journal on Computing*, 16(4):717–727.

Xu, Q., Ding, C., Liu, J., and Luo, B. (2015). PCA-guided search for K-means. *Pattern Recognition Letters*, 54:50–55.

Yaglom, A. (1957). *Correlation Theory of Stationary and Related Random Functions*. Springer Series in Statistics. Springer-Verlag New York Inc., 1st edition.

Yang, T., Li, Y.-f., Mahdavi, M., Jin, R., and Zhou, Z.-H. (2012). Nyström method vs random Fourier features: A theoretical and empirical comparison. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 476–484. Curran Associates Inc.

Yang, Z., Smola, A. J., Song, L., and Wilson, A. G. (2015). A la Carte – Learning Fast Kernels. In Lebanon, G. and Vishwanathan, S. V. N., editors, *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 1098–1106. PMLR.

Yu, S., Yu, K., Tresp, V., Kriegel, H. P., and Wu, M. (2006). Supervised probabilistic principal component analysis. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 464–473. ACM.

Yue, Y. and Joachims, T. (2009). Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1201–1208. ACM.

Zhang, K., Tsang, I. W., and Kwok, J. T. (2008). Improved Nyström low-rank approximation and error analysis. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1232–1239. ACM.

Zhu, J., Yu, H., Fan, H., Liu, H., and Shi, Y. (2001). Design of new selective inhibitors of cyclooxygenase-2 by dynamic assembly of molecular building blocks. *Journal of Computer-Aided Molecular Design*, 15(5):447–463.

Zinkevich, M. A., Smola, A. J., Weimer, M., and Li, L. (2010). Parallelized stochastic gradient descent. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 2595–2603. Curran Associates, Inc.