Multi-View Kernel Methods for Binding Affinity Prediction

Dissertation

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Katrin Ullrich

aus

Suhl

Bonn 2021

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

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

Tag der Promotion: 10.09.2021
Erscheinungsjahr: 2021

Abstract

In the present thesis, we focus on the potential and limits of multi-view regression techniques in the field of ligand affinity prediction.

Multi-view learning (MVL) denotes machine learning approaches that utilise different representations (views) on data. MVL can be grouped into three classes of algorithms: multiple kernel learning (MKL), co-training style algorithms, and subspace learning-based approaches [Xu et al., 2013]. The first group considers prediction models that use a linear combination of view-related predictors. Co-training style algorithms include the pairwise comparison of predictions from multiple views into the training process. The class of subspace learning-based approaches incorporates a common subspace of different feature spaces for their predictions. It is known that MVL improves the performance in many important real-world applications, but there is hardly any thorough evaluation of MVL in the life science domain. We are the first to apply MVL to affinity prediction under particular consideration of the availability of molecular compounds with affinity annotation.

The binding of small compounds to large protein molecules is central to the activity of the cell as such processes are involved in the majority of biochemical pathways. A real-valued binding affinity characterises the binding strength of the protein-ligand complex. The identification of these affinities serves as an initial point for the discovery of drugs correlated with the respective pathways and is therefore an important real-world problem to master. Binding affinities can be determined on a large scale via high-throughput screening (HTS) [Mahé and Vert, 2009]. Databases with millions of annotated compounds are the result of these efforts. As HTS is very time- and cost-consuming, and the number of proteins and potential compounds is huge, machine learning methods for the prediction of binding affinities were established as support. For binding affinity prediction in ligand-based virtual screening, single-view support vector regression (SVR) utilising molecular fingerprints [Balfer and Bajorath, 2015] is the state-of-the-art approach.

The special situation with respect to the representation and availability of data suggests the application of multi-view regression for affinity prediction. Views are data representations canonically related to so-called kernel functions which provide a generalised similarity measure for data instances. On the one hand, different representations of data instances are available for affinity prediction naturally as a large variety of molecular descriptors designed for different purposes exist [Bender et al., 2009]. On the other hand, labelled data is typically not abound because of the huge number of existing proteins. We address these challenges and present multi-view kernel approaches to overcome the mentioned difficulties. The general question of the thesis is: *Can affinity prediction benefit from the diversity of useful representations for molecular compounds via multi-view learning?* We will answer this question in three different multi-view prediction settings of high practical relevance in concordance with the classes of MVL. We show that the affinity prediction performance can be improved by the application of MVL techniques. We present a systematical procedure to deal with a multitude of graph representations as well as novel kernel algorithms for semi-supervised and unsupervised learning.

Initially, we consider the case of supervised learning where labelled training compounds for a precise protein are available. Moreover, we assume different molecular fingerprints based on the graph structure or other molecular properties exist. We enhance the set of existing fingerprints with feature vectors of systematically enumerated cyclic, tree, and shortest path patterns, as well as Weisfeiler-Lehman label patterns of different calculation depths [Ullrich et al., 2016b]. We are the first who apply multiple kernel learning (MKL) that identifies a linear combination of the utilised set of views in the context of affinity prediction. In addition to the rich set of data representations, we investigate both a loss function known from regularised least squares regression (RLSR) [Cortes et al., 2009] and one from SVR [Vishwanathan et al., 2010]. In our practical experiments, we analyse the influence of different patterns on the affinity prediction performance and address the question: *Can we find better molecular fingerprint representations for affinity prediction by a systematic combination of graph patterns and omit the expensive choice of the optimal representation in the training phase?* We suggest a scheme to perform a systematical preselection of graph patterns for molecular compounds. In our empirical analysis we show that MKL with a preselection of graph patterns or standard molecular fingerprints outperforms state-of-the-art algorithms for ligand affinity prediction.

In the second group of approaches we take into account the small number of compounds with known affinity and exploit the availability of unlabelled data. In addition to empirical risk minimisation in the supervised case, the technique of co-regularisation permits a semi-supervision via an adjustment of predictions for unlabelled instances. This adjustment occurs for pairwise predictions from different views. We define co-regularised support vector regression (CoSVR) [Ullrich et al., 2016a, 2017] analogously to the approach of co-regularised least squares regression (CoRLSR) [Brefeld et al., 2006]. We present the CoSVR algorithm and theoretical properties of it. CoSVR is the first support vector regression approach with a co-regularisation term for the comparison of view-related predictions for unlabelled data. We contrast a co-regularisation term with squared and ε-insensitive loss function. Both theoretically and empirically we answer the question: *Can we compensate for few labelled examples by an abundance of unlabelled instances and multiple views on data?* We define a novel kernel algorithm for semi-supervised learning in different variants with decreasing number of optimisation variables. Furthermore, we derive a multi-view CoSVR variant with single-view complexity and a Rademacher bound for the corresponding function class. We prove empirically that ligand affinity prediction profits from the application of CoSVR in comparison to the baselines.

Finally, we consider the unsupervised task of orphan screening where no labelled training data is available for the considered protein. We focus on the following question: *How can we tackle orphan screening using binding information for other proteins and similarity values for proteins?* We propose two algorithms for the solution of this problem. Firstly, we define corresponding projections (CP) [Ullrich and Gärtner, 2014, Giesselbach et al., 2018] as a novel kernel method for unsupervised or transfer learning. Secondly, we show how orphan screening can be solved via knowledge-based principal component analysis (IPCA) [Oglic et al., 2014] in form of orphan principal component analysis (OPCA) [Giesselbach et al., 2018]. CP and OPCA can be applied as single- and multi-view algorithm and both are also applicable to learning tasks like classification. Our empirical results show that CP outperforms the orphan screening baseline of the target-ligand kernel approach and approximates the performance of supervised algorithms that utilise very few labelled training examples.

Acknowledgements

First of all, I would like to thank my supervisors Prof. Dr. Thomas Gärtner and Prof. Dr. Stefan Wrobel for teaching me and inviting me to the exciting and groundbreaking research field of machine learning and for their long-term support. I am also grateful for the support of the b-it research school of Bonn and Aachen universities. I am deeply grateful for all the kind and helpful friends and colleagues at Fraunhofer IAIS as well as from the University of Bonn. I thank my coauthors Pascal, Micha, Gecko, Sven, Jenny, Martin, Christoph, as well as my colleagues Tamás, Olana, Mario, Dino, Daniel, Olga, Roman, Myriam, Marie-Luise, Silvia, all members of the IAIS group of Prof. Dr. Kristian Kersting, and all others I do not list here in person. In particular, I appreciated the valuable technical discussions with Prof. Dr. Thomas Gärtner, Prof. Dr. Stefan Wrobel, Prof. Dr. Kristian Kersting, Micha, Pascal, Gecko, Dino, Roman, and Olana. I thank the members of Prof. Dr. Jürgen Bajorath's research group at the LIMES institute in Bonn, in particular Hanna and Martin, for giving me valuable insights from life science informatics and providing me with real-world datasets to do affinity prediction. I am very much obliged for the cordial welcome and support of my new colleagues and friends at Fraunhofer IWU and in our new home Chemnitz. While writing the thesis, very kind people took care for our children in a heartly manner. I give my thanks to our *Leihoma* Elisabeth, Kita Newmanhaus, Nil, Sarah, and Nadja from Ratz und Rübe, Kita Campulino, and of course to my parents. I am very grateful for all my friends from Suhl, Ilmenau, Jena, Madrid, Bonn, Chemnitz, and everywhere else to be there for me. Most importantly, I am grateful and thankful for my parents Britta and Rainer, Bine, Boro, Paul, Felix, Lotta, Maja, Otto, and my husband Tino to have you and for everything you give to me.

Contents

List of Figures

List of Tables

Abbreviations

ADME	**A**bsorption, **D**istribution, **M**etabolism, and **E**xcretion
CCA	**C**anonical **C**orrelation **A**nalysis
(Co)**CP**	(Co-regularised) **C**orresponding **P**rojections
(K)**CP**	(Kernel) **C**orresponding **P**rojections
(L)**CP**	(Linear) **C**orresponding **P**rojections
(MV)**CP**	(Multi-view) **C**orresponding **P**rojections
(S)**CP**	(Simplified) **C**orresponding **P**rojections
CPK	**C**yclic **P**attern **K**ernel
CV	**C**ross-**V**alidation
CVXOPT	**C**on**V**e**X** **OPT**imisation software package
DNA	**D**eoxyribo**N**ucleic **A**cid
ECFP	**E**xtended **C**onnectivity **F**inger**P**rint
ERM	**E**mpirical **R**isk **M**inimisation
GPCR	**G**-**P**rotein **C**oupled **R**eceptor
GpiDAPH3	**3**-point **G**raph-based π-**D**onor-**A**cceptor **PH**armacophore fingerprint
HTS	**H**igh-**T**hroughput **S**creening
i.i.d.	**i**ndependently **i**dentically **d**istributed
JL	**J**ohnson-**L**indenstrauss
KKT	**K**arush-**K**uhn-**T**ucker
LKRR	**L**earning **K**ernel **R**idge **R**egression
Maccs	**M**olecular **ACC**ess **S**ystem
MKL	**M**ultiple **K**ernel **L**earning
MPK	**M**ulti-**P**attern **K**ernel
MVL	**M**ulti-**V**iew **L**earning
NP	**N**on-Deterministic **P**olynomial Time Complexity Class

(I)**PCA**	(Interactive knowledge-based) **P**rincipal **C**omponent **A**nalysis
((MV)O)**PCA**	((Multi-View) Orphan) **P**rincipal **C**omponent **A**nalysis
PDB	**P**rotein **D**ata **B**ank
(Co)**RLS**	(Co-)**R**egularised **L**east **S**quares
(Co)**RLSR**	(Co-)**R**egularised **L**east **S**quares **R**egression
RKHS	**R**eproducing **K**ernel **H**ilbert **S**pace
RNA	**R**ibo**N**ucleic **A**cid
RMSE	**R**oot **M**ean **S**quared **E**rror
(Co)**RRM**	(Co-)**R**egularised **R**isk **M**inimisation
SDF	**S**tructure **D**ata **F**ormat
SMARTS	**SM**ILES **AR**bitrary **T**arget **S**pecification
SMILES	**S**implified **M**olecular **I**nput **L**ine **E**ntry **S**pecification
SMO	**S**equential **M**inimal **O**ptimisation
SP(K)	**S**hortest **P**ath (Kernel)
SPP	**S**imilarity **P**roperty **P**rinciple
s.t.	**s**ubject **t**o
SVC	**S**upport **V**ector **C**lassification
SVM	**S**upport **V**ector **M**achine
(Co)**SVR**	(Co-regularised) **S**upport **V**ector **R**egression
TLK	**T**arget-**L**igand **K**ernel
(QC)**QP**	(Quadratically Constrained) **Q**uadratic **P**rogram
QSAR	**Q**uantitative **S**tructure-**A**ctivity **R**elationship
WL(K)	**W**eisfeiler-**L**ehman (Kernel)

Symbols

$\mathcal{C}$, $\mathcal{L}$, $\mathcal{P}$, $\mathcal{T}$	...	Classes of cyclic, label, shortest path, and tree patterns
$\mathcal{D}$	...	Probability distribution
$\mathcal{H}$	...	Reproducing kernel Hilbert space
$\mathcal{O}(f)$	...	Complexity class, $g \in \mathcal{O}(f) \Leftrightarrow \exists\ c > 0\ \forall\ x : g(x) \leq c \cdot f(x)$
$\mathcal{Q}(x)$	...	Objective function of variable x
$\mathrm{Corr}(X, Y)$	...	Correlation (coefficient) of random variables X and Y
$\mathrm{Cov}(X, Y)$	...	Covariance of random variables X and Y
$\mathrm{Var}(X)$	...	Variance of random variable X
$\mathbb{E}, \hat{\mathbb{E}}$	...	expectation, empirical expectation (mean value)
$\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}^+$	...	Natural numbers, real numbers, non-negative real numbers
$\mathbb{1}_A$	...	Indicator function of set A
$\mathbf{0}_n, \mathbf{1}_n$	...	All-zero vector $(0, \ldots, 0)^T$, all-one vector $(1, \ldots, 1)^T \in \mathbb{R}^n$
$\mathbf{0}_{n \times n}$, $\mathbf{1}_{n \times n}$	...	Matrix of $n \times n$ zeros, matrix of $n \times n$ ones
$\mathbf{I}_n$	...	Identity matrix of dimension $n \times n$
L	...	Lagrangian function
ℓ	...	Loss function
e_i	...	Unit vector $(0, \ldots, 0, 1, 0, \ldots, 0)^T$ of i-th dimension
x^*	...	Optimal value for function variable x
$\|x - y\|$	...	Euclidean distance between x and y in $\mathbb{R}^d$, equals $\|x - y\|_2$
$\|x\|_p$	...	ℓ_p-Norm of $x \in \mathbb{R}^d$: $\|x\|_p = (\sum_{i=1}^d x^p)^{1/p}$
$\langle x, y \rangle$	...	Inner product of $x, y \in \mathbb{R}^d$
$M^\dagger$	...	Pseudoinverse of matrix M
Φ	...	Feature vector
Σ, Σ^*	...	Alphabet, all words over alphabet Σ
		(Σ also used as summation symbol in Chapter 4)

Für Tino, Maja und Otto.

Chapter 1

Introduction

Artificial intelligence and machine learning in particular are highly topical research fields with a rapid development. At present, their achievements have an almost immediate and wide influence on everyone's lives. Just to mention a few, we point to navigation systems, placing of advertisement, or search engines. Thus, the great potential of machine learning research comes along with a responsible handling of its outcomes. In the present thesis we focus on *ligand affinity prediction* as a promising and important application in the medical domain. Prospectively, it inherits the capability to substantially support and guide the discovery of novel drugs via computational methods.

According to Kaplan and Haenlein [2019], artificial intelligence is a system's ability to correctly interpret external data, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation. It is the entirety of intelligent data reception, processing, and reaction of machine hardware and software. Machine learning denotes autonomous learning and adaption processes of computers or programs by turning experience into expertise [Shalev-Shwartz and Ben-David, 2014]. In this spirit, machine learning can be considered the heart of artificial intelligence. Strongly related and overlapping with machine learning are the fields of data mining and knowledge discovery. They refer to the (typically unsupervised) extraction process of knowledge and patterns from huge amounts of data [Berthold et al., 2010]. In contrast to data, which correspond to single instances or events, knowledge expresses general rules and principles for a group of objects [Berthold et al., 2010]. As a subfield of practical computer science [Herold et al., 2007], machine learning uses insights of many other scientific subjects. For example, techniques appear from convex optimisation [Boyd and Vandenberghe, 2004], functional analysis (theory on kernels) [Werner, 1995], complexity and information theory [MacKay, 2003], probability theory and statistics [Mitchell, 1997, Cherkassky and Mulier, 1998, Hastie et al., 2001]. Moreover, accomplishments like autonomous driving, military drones, or automatic speech and face recognition have to be discussed under the viewpoint of law and ethics. Groundbreaking technological achievements in the last centuries like powerful computers, storage media and data connections [Berthold et al., 2010] enabled machines to already achieve astonishing results. The additional gathering of huge amounts of different types of data in a very short time period (*big data*) facilitates artificial intelligence in various applications on a human level and beyond [Kaplan and Haenlein, 2019]. Because of the enormous calculation power of computers, the increasing potential of algorithms, and the huge amounts of data

daily arising in the bio-medical area, computer-aided medicine and pharmacy became an important application of machine learning in recent years.

The idea to apply machine learning in the biomedical research field yet appeared decades ago [Mitchell, 1997] and already became reality, if one considers, for example, the computational analysis of the genome. The intention is to turn medical data into medical knowledge in order to finally improve the treatment of diseases [Shalev-Shwartz and Ben-David, 2014]. The present thesis is dedicated to an application from chemoinformatics named *ligand affinity prediction* where one intends to predict the chemical binding affinity of small compounds (ligands) to protein molecules. Protein-ligand complexes play an important role in the effectiveness of drug substances. Therefore, the correct prediction of ligands with computational methods would support the drug discovery process and make laboratory experiments more time- and cost-efficient [Michielan and Moro, 2010, Sheridan et al., 2015]. Numerous virtual screening approaches model the ligand prediction task as a classification problem which is a simplification of the reality. For this reason we, solve it in a regression scenario as the prediction of precise affinities, i.e., the actual strength of the protein-ligand binding, is more meaningful for the assessment of the ligand's activity towards a certain protein.

Small molecular compounds as potential ligands can be represented with molecular fingerprints that gather various structural or physico-chemical properties of the respective molecules. A variety of such vectorial representations for molecules exist a priori from other similar applications in chemoinformatics and can be used to train a prediction model for ligand affinity using information of known affinity values. The multi-view machine learning paradigm seems convenient in this setting of multiple data representations with no particular preferences to one representation or view. With respect to the practical application, we want to take advantage from the multitude of data representations for the affinity prediction performance. Regarding machine learning we exploit that views on data and kernel functions are canonically related. Kernel functions imply a generalised similarity measure for the data instances of interest. We will see that the comparison between instances, such as small molecular compounds or proteins, in form of kernel values plays a central role in the machine learning models below. Thus, we contribute to the field of (multi-view) kernel methods as well. Interestingly, the categorisation of multi-view learning algorithms can be aligned well with different learning scenarios of affinity prediction regarding the availability and format of labelled training examples, numbered with (i)–(iii).

(i) Firstly, we consider the supervised learning scenario where affinity-labelled ligands are sufficiently available. Our aim is to particularly benefit from the graph structure of molecules and combine the various graph patterns that can be used for the representation of compounds via multiple-kernel learning techniques.

(ii) Secondly, the group of co-regularised algorithms can be assigned to the field of semi-supervised learning. Typically only few annotated compounds are available for a given protein but countless unlabelled molecules in chemical databases.

(iii) Thirdly, the class of projection-based algorithms is applied to tackle the orphan screening learning problem. Orphan screening denotes the search for novel ligands if no training affinities for the considered protein are known yet. We show how this unsupervised learning problem can be solved via dimensionality reduction techniques which are based on projections of high-dimensional molecular fingerprints.

Due to the availability of different effective representations for small molecules, affinity prediction is qualified well to be approached with multi-view learning. Due to its high practical relevance in the process of drug discovery and design it is worth a thorough investigation. However, the proposed algorithms can be applied in various other learning scenarios as well which were not in the focus of the present work. Other applications which come along with multiple views are cross-language and web text classification, natural language processing problems or issues of computer vision like object or face recognition [Xu et al., 2013].

In the following section, we introduce the subject of the present thesis avoiding formal definitions. We explain why it is interesting and non-trivial, and highlight problems with existing approaches. Subsequently, we summarise the results and achievements of the present thesis. A thorough related work section on *multi-view learning* and *ligand affinity prediction* follows in Sections 1.2 and 1.3, respectively. Section 1.4 gives an outlook on the thesis content.

1.1 Objectives and Contribution

Binding affinity prediction is an application for regression which describes the determination of real-valued chemical affinities of small molecules (ligands) to proteins with machine learning techniques. The learning scenario of affinity prediction comes along with a particular situation concerning data availability and data representation. Firstly, molecular learning objects can canonically be identified with graphs. Secondly, in addition to ligands with known affinities, a big number of synthesizable small molecules without affinity label are gathered in databases for learning. In contrast to unlabelled compounds, annotated ligands are rare and expensive to obtain. A third property of the affinity prediction setting is the representation of data instances typically in form of molecular fingerprints. Many different fingerprints are available and each of these usually high-dimensional vectorial representations comprises a characteristic set of molecular features. The mentioned prerequisites lend themselves for an application of multi-view algorithms for binding affinity prediction. These algorithms are machine learning methods which utilise different views or representations on data instances in order to train a model. The involved kernel function can be imagined a generalised similarity measure for data instances, e.g., for graphs or vectors. The kernel function is canonically related to the data representation and endows the corresponding kernel methods with beneficial properties. The objective of the present thesis is to answer the question *whether binding affinity prediction can be improved under particular consideration of molecular representations and availability of molecular data using multi-view kernel algorithms for regression?* We will specify the objectives for the three main chapters below.

Ligands are small molecular compounds with a low molecular weight, whereas proteins are large molecules composed of amino acids. Protein-ligand complexes are highly relevant in the majority of biochemical processes of organisms. Numerous drugs act as protein ligands and by this means trigger or regulate cellular pathways connected with the development of diseases. In view of this background, the identification or prediction of binding strengths for protein-ligand complexes is of prime importance for the discovery and development of novel drugs. In this connection, a special position is taken on by so-called orphan proteins for which no ligand affinities are known so far. Although

laboratory experiments for the determination of affinities can already be performed efficiently, the process is still very time-consuming and cost-intensive in practice. Machine learning should be used to assist and support this relevant research field. The automatic suggestion of ligand candidates and their protein affinities would speed up the drug discovery process and at the same time make it more resource efficient. Although we focus on the prediction of affinities throughout this work, the proposed approaches can as well be applied to other applications from a regression domain with analogous preconditions on data and learning scenario.

Existent affinity prediction approaches do not or only rarely exploit the precise learning scenario accompanied with the particular regression task.

(i) Ligands of proteins and their corresponding affinity can be found in molecular databases and used as training examples for supervised learning algorithms. For the representation of molecules one can choose between a variety of molecular fingerprints. Their respective feature sets comprise physico-chemical properties, structural patterns of the molecular graph, or 3D information, depending on the original purpose they were designed for. It is not a trivial decision which molecular fingerprint to utilise for the affinity prediction task. In previous work this problem was addressed via fingerprint fusion or recombination techniques as well as the plain comparison of results for various fingerprints.

(ii) As the overall number of proteins is large, for one particular protein there are often only very few affinity-annotated ligands. In contrast, there are many compounds that carry affinity values with respect to other proteins and millions of potentially synthesizable database molecules without binding information. This unlabelled data was to the best of our knowledge not yet utilised in the context of affinity prediction.

(iii) For the special case of orphan screening, i.e., affinity prediction for proteins without known ligands, only very few machine learning approaches exist at all.

The aim of this work is to propose solutions for these non-trivial issues in the context of affinity prediction. However, the insights and results will be applicable for general learning problems with appropriate preconditions on the learning scenario as well.

(i) If there are multiple options for the representation of molecular data, the optimal representation for the respective regression problem must be found in a preprocessing step. Apart from fingerprint recombination attempts there are hardly any systematic approaches to tackle the variety of fingerprints. Instead of making a choice, we suggest to utilise multiple fingerprints simultaneously via multi-view learning. *We investigate whether we can find better molecular fingerprint representations for affinity prediction by a systematic combination of graph patterns and omit the expensive choice of the optimal representation in the training phase?*

(ii) A small number of labelled training molecules most probably leads to weak prediction models for the considered protein. We intend to compensate for the lack of sufficient labelled data with the inclusion of unlabelled data. To this aim, we make use of multiple fingerprints and multi-view learning techniques. *We investigate whether it is possible to compensate for few labelled examples for affinity prediction by an abundance of unlabelled instances and multiple views on data?*

(iii) In the case of orphan screening the absence of affinity information for the orphan protein is problematic per se, as the binding preferences cannot be concluded from labelled training compounds or transferred from one protein to another without further information sources. We present a solution for this unsupervised learning task by a novel transfer learning algorithm that uses a kernel function for proteins and one for ligands. *We investigate both in a single- and a multi-view scenario, how we can tackle orphan screening using binding information for other proteins?*

The present thesis contributes both to the field of chemoinformatics and to machine learning research. With respect to machine learning, we investigate a wide group of approaches as we consider and explore algorithms in the field of supervised, semi-supervised, and unsupervised learning. We advance multi-view regression in a semi-supervised scenario via the introduction of the novel algorithm *co-regularised support vector regression* and variants of it. Furthermore, in the scenario of unsupervised learning we introduce *corresponding projections* that can be used for single- and multi-view learning and is an all-purpose concept in the sense that it is not restricted to regression. Additionally, we contribute to kernel methods as we present the approaches in a general kernelised formulation. With regard to chemoinformatics, we suggest procedures and algorithms to handle the variety of existing molecular fingerprints for small compounds and investigated the prerequisites of affinity prediction in terms of data availability. Moreover, we present a solution for orphan screening, for which only few regression approaches exist at all. Affinity prediction research itself will be a central point on the way to automatic drug discovery and can therefore be regarded important.

(i) In the supervised setting, we answer the questions of the present thesis affirmatively. More precisely, we tackle the inherent challenge of the optimal fingerprint choice for the representation of compound instances by using multiple representations simultaneously. The application of a linear combination of multiple predictor functions which relate to the views on data enables the concurrent inclusion of different compound representations in a single optimisation. In particular, we take the graph structure of the learning objects into consideration and perform a systematic selection of cyclic, tree, and shortest path patterns as well as Weisfeiler-Lehman labels for graphs [Shervashidze et al., 2010]. The preselection of patterns can then be utilised in the actual multiple kernel learning model. We call this novel preselection scheme to handle the multitude of data representations multi-pattern kernel multiple kernel learning [Ullrich et al., 2016b]. In our work *Ligand Affinity Prediction with Multi-Pattern Kernels* [Ullrich et al., 2016b], we show that the performance of ligand affinity prediction can be improved by the simultaneous inclusion of different data representations via multiple kernel learning [Cortes et al., 2009, Vishwanathan et al., 2010].

(ii) Analogous to the supervised case, we achieve the objectives of the present thesis stated above for the semi-supervised setting as well. The novel co-regularised support vector regression algorithm presented in *Ligand-Based Virtual Screening with Co-Regularised Support Vector Regression* [Ullrich et al., 2016a] and *Co-Regularised Support Vector Regression* [Ullrich et al., 2017] includes unlabelled data in addition to annotated molecules as well as multiple representations on data into the learning process. We define co-regularised support vector regression as a novel kernelised multi-view algorithm and further variants with respect to the number of optimisation variables. We show that the variant Σ-co-regularised support vector

regression has complexity properties of a single-view algorithm, which reduces the running time drastically. Moreover, we prove a bound for the Rademacher complexity of the corresponding co-regularised candidate function class that can be applied to restrict the expected error. By means of the co-regularisation technique we are able to reduce the prediction error of ligand affinities despite of only few annotated training molecules and without an expensive choice for the best data representation. To be more precise, the multi-view approaches in the empirical analysis at least performed as good as the best single-view baseline by including all molecular representations in one optimisation problem. By investigating the more realistic scenario of few labelled ligands and sufficient unlabelled database compounds, we address one of the limitations of affinity prediction from the introduction.

(iii) For the unsupervised scenario, we achieved the objectives stated above by proposing the two novel projection-based methods corresponding projections and orphan principle component analysis for the solution of the ambitious orphan screening problem. Both corresponding projections [Ullrich and Gärtner, 2014] and orphan principal component analysis can be applied as single- and multi-view algorithm. In *Corresponding Projections for Orphan Screening* [Giesselbach et al., 2018] we show how the combination of labelled information from other proteins and inter-protein relations can be used to solve orphan screening. We prove empirically that corresponding projections outperforms the state-of-the-art approach of target-ligand kernels and approximates the results of supervised single-view support vector regression using only very few labelled training examples. We obtained similar results for corresponding projections and baselines when we applied canonical multi-view compound representations, in particular, by means of a dimensionality reduction step for the generation of the multi-view representation.

The content of this thesis is based on the following publications

- [Ullrich and Gärtner, 2014] Kernel Corresponding Projections for Orphan Targets. K. Ullrich and T. Gärtner. Extended abstract for the workshop on Multi-Target Prediction (KERMIT) at the *European Conference on Machine Learning*, 2014,
- [Ullrich et al., 2016b] Ligand Affinity Prediction with Multi-Pattern Kernels. K. Ullrich and J. Mack and P. Welke. Conference paper in *Proceedings of the International Conference on Discovery Science*, 2016,
- [Ullrich et al., 2016a] Ligand-Based Virtual Screening with Co-Regularised Support Vector Regression. K. Ullrich and M. Kamp and T. Gärtner and M. Vogt and S. Wrobel. Workshop paper in *Proceedings of the workshop on Data Mining in Biomedical Informatics and Healthcare (DMBIH) at the International Conference on Data Mining*, 2016,
- [Ullrich et al., 2017] Co-Regularised Support Vector Regression. K. Ullrich and M. Kamp and T. Gärtner and M. Vogt and S. Wrobel. Conference paper in *Proceedings of the European Conference on Machine Learning*, 2017,
- [Giesselbach et al., 2018] Corresponding Projections for Orphan Screening. S. Giesselbach and K. Ullrich and M. Kamp and D. Paurat and T. Gärtner. Workshop paper in *Proceedings of the workshop on Machine Learning for Health (ML4H) at the Neural Information Processing Systems conference*, 2018.

Preliminary work in the field of ligand prediction via structured outputs was done in [Ullrich et al., 2010].

1.2 Multi-View Learning

With regard to algorithms we focus on the field of *multi-view learning* (MVL), which means to solve machine learning tasks using different views on data. A *view* is a sight or representation of data instances of interest and can be imagined as a predefined collection of features. In contrast to the conventional single-view learning, MVL approaches utilise multiple data representations with distinct feature sets at the same time [Sun, 2013]. Multiple views arise from various sources of supervision or description. For example, video and audio recording are two different ways to monitor the same object or event. Apart from the improvement in learning performance that has been proven both in theory [Dasgupta et al., 2002, Rosenberg and Bartlett, 2007, Cortes et al., 2010] and many practical applications (see examples from above), MVL offers a way to manage the variety of data descriptors that frequently appear in real-world scenarios by simply using them all simultaneously. It supersedes an exhaustive choice procedure for the optimal view for a given learning task. In the last decade, multi-view learning became more and more prominent in machine learning. It turned out that many practical and theoretical aspects of learning can be studied within this setting, e.g., the availability of data, the handling of different learning tasks, or the generalisation performance comparison between different algorithms. Using the example of ligand affinity prediction we demonstrate the flexibility of multi-view learning algorithms and at the same time suggest novel techniques to the chemical community for the practical problem. MVL can be grouped differently. We introduce MVL according to the survey of Xu et al. [2013] and adhere to the classes *co-training style algorithms*, *multiple kernel learning*, and *subspace learning-based approaches* as it was motivated at the beginning of the present chapter. MVL techniques can potentially be applied in a wide range of learning tasks and applications. However, we solve the task to learn a predictor function for real-valued ligand affinities.

1.2.1 Multiple Views and Definition

For the present and the following sections on MVL we anticipate Chapter 2 with regard to machine learning and notation. We consider objects from a space $\mathcal{X}$ for which we intend to solve a learning task. A view v on data is a representation of the learning objects in an appropriate feature space $\mathcal{H}$. For the time being, we restrict to the case that the corresponding feature map Φ_v implies a d_v-dimensional vectorial representation for the data. As mentioned above, a thorough theoretical introduction can be found in Chapter 2 below. In particular, a formal definition of a view will be presented in Definition 2.17. Moreover, we will explain how feature maps and kernel functions are canonically related and go into detail with the feature space $\mathcal{H}$. Multiple views essentially appear in two situations which in a way are contrary. On the one hand, different feature representations of the same objects or events exist a priori in case different information sources are available. For example,

- color, texture, and attached text can describe one and the same image [Xu et al., 2013],

- video and audio signals which describe a movie or event [Sun, 2013],
- different camera angles are another way to describe a movie literally with multiple views [Sridharan and Kakade, 2008],
- a text can be translated into different languages [Sridharan and Kakade, 2008],
- and finally, structural graph patterns or physico-chemical properties can be used to describe molecular compounds as potential ligands of proteins (see Section 1.3 below).

On the other hand, a given set of features can be used to generate multiple views via feature selection or partitioning, for example

- via randomly splitting the features into two ore more subsets [Brefeld et al., 2006],
- via feature clustering approaches or other optimised division algorithms [Xu et al., 2013],
- by representing texts with terms of different lengths [Matsubara et al., 2005],
- or simply by the application of different kernel functions on one and the same feature set [Xu et al., 2013].

However, not every view combination is appropriate for the application in a MVL algorithm, independent of whether multiple views exist naturally or are the result of a view generation procedure. There are attempts to assess the sufficiency or quality of views to be profitable in an MVL scenario (for more details we refer to Xu et al. [2013]). To some extent, the multi-view approaches *multiple kernel learning* and *canonical correlation analysis* introduced below deliver some information on the appropriateness of the utilised views as a byproduct in form of a kernel linear combination and correlation coefficients (compare Sections 1.2.3 and 1.2.5).

Suppose we face a learning task to assign to an input from $\mathcal{X}$ a certain output from $\mathcal{Y}$ via a functional model f. Assume, the data objects $\mathcal{X}$ can be described in M different ways, i.e., there are M feature maps $\Phi_v : \mathcal{X} \to \mathbb{R}^{d_v}$, where $v = 1, \ldots, M$. Then

$$f_v(\Phi_v(\mathcal{X})) \to \mathcal{Y}$$

is the learning model based on the v-th feature representation. For the sake of simplicity, we often write $f_v(\mathcal{X})$ which implies that f_v actually operates on the v-th view on data. In the case of regression we consider a predictor function f_v with output space $\mathcal{Y} = \mathbb{R}$. Without further restrictions, a view model f_v can be found with a single-view method, i.e., a machine learning algorithm that only utilises one view on data, such as for example *least squares regression*, *support vector machines*, or *principal component analysis*. These approaches will be discussed later and appear as important baselines in comparative experiments between single-view and multi-view learning. We will denote a model f_v a *single-view model*. In contrast, if the predictor function is not independent of the respective other views but corresponds to a fixed view, we will call it a *view model*.

In the literature, MVL is introduced as learning in the presence of distinct feature sets or representations [Sun, 2013]. Although the intention behind this definition of MVL

is intuitively clear, it would include single-view models or an approach that uses the average of M independently learned single-view predictors as final predictor. Therefore, we claim that the idea of MVL is to find a model f that depends on M views

$$f(\Phi_1(\mathcal{X}), \ldots, \Phi_M(\mathcal{X})) \to \mathcal{Y},$$

such that the simultaneous awareness of all M views has an influence on the final MVL model f and, hopefully, leads to an improved performance in comparison with respective baselines. For example, a canonical MVL approach is to attach the M feature representations of data instances one after another and learn a model with a single-view approach with the concatenated feature vectors. According to Xu et al. [2013] the concatenation method suffers from overfitting. Another simple MVL approach for classification or regression is to use the average of M view predictors as prediction model. We point to the fact that this is different from taking the average of M independent view predictors. Both the concatenation approach and the average predictor approach will be considered in the empirical sections below.

1.2.2 Principles and Branches

In our definition of MVL we postulate the simultaneous knowledge of all views. Two principles underlie MVL approaches [Xu et al., 2013] which finally result in this demand. At first, one assumes that each view suffices to describe the data appropriately and solve the learning task alone. Hence, the *consensus principle* embraces the efforts in MVL to find consistent view predictors by minimising the differences between pairs of hypotheses [Dasgupta et al., 2002]. Though, if all predictors would be consistent from the beginning there was nothing to benefit from multiple views. Therefore, the *complementary principle* unites the beliefs in MVL that each view should contribute some information to the solution of the respective learning task which the other views do not deliver. However, just like the complete accordance of two models also grave differences between views could hinder a useful MVL result. This aspect of MVL has already been considered [Nigam and Ghani, 2000, Christoudias et al., 2008].

Similar to the definition of MVL, also the branches of MVL are not reported concordantly in the literature. For reasons explained at the beginning of the introductory chapter, in the present thesis we orient to the MVL classes suggested by Xu et al. [2013]

(i) Multiple kernel learning,

(ii) Co-training style algorithms, and

(iii) Subspace learning-based approaches,

and dedicate a main chapter to each class towards affinity prediction

(i) Multiple kernel learning for supervised affinity prediction (Chapter 3),

(ii) Co-regularisation for affinity prediction with few labelled data (Chapter 4), and

(iii) Projection-based learning for orphan screening (Chapter 5).

The numbering (i)–(iii) refers to the one already used above. In the Sections 1.2.3, 1.2.4, and 1.2.5 we briefly introduce the three classes of MVL and present related work in the respective fields which is relevant to the present work. We anticipate some concepts from machine learning which will be explained formally in Chapter 2 below. Sridharan and Kakade [2008] similarly divide MVL into *co-regularisation* and *canonical correlation analysis-based algorithms*, whereas Sun [2013] only distinguishes between co-training and co-regularisation style algorithms. In addition to co-training and co-regularisation, Zhao et al. [2017] mention a further class called *margin-consistency style algorithms*. The literature on MVL is predominated by the prediction task of classification, in particular, co-training has been studied extensively in theory and in practice.

1.2.3 Multiple Kernel Learning

We already know that views on data are canonically related to so-called kernel functions. If there is a feature representation or view on data this automatically implies a way to compare data instances. The precise relation between data representation and kernel functions will be explained in detail in Section 2.5 below. Every view or kernel delivers different aspects of similarity which all can be useful for the respective learning task [Xu et al., 2013]. In order to utilise multiple sources of information at the same time and to prevent an exhaustive search for the optimal data representation, *multiple kernel learning* (MKL) looks for a combination of multiple kernels to form a new kernel [Gönen and Alpaydin, 2011]. The combination parameters provide the opportunity to regulate the influence of each kernel function. That means, using MKL one is looking for a kernel k_b as a function of different kernels $k_1, \ldots, k_M$

$$k_b(x, x') = f_b(k_1, \ldots, k_M|b),$$

where x, x' are learning objects and b the parameterisation of the functional relationship f_b. The idea to connect multiple kernels that appear in a parameterised form of the target function for learning is very similar to the concept of *boosting* or *ensemble learning*, where a number of classifiers or even learning algorithms are merged such that the final model is better than the potentially weaker single classifiers or algorithms. *Multiple kernel learning* (MKL) might be confused with *multi-view learning* (MVL). However, MKL denotes the subgroup of MVL algorithms reviewed in this section. MKL is also often used synonymously with learning a linear combination of kernel function

$$k_b(x, x') = \sum_{v=1}^{M} b_v k_v(x, x')$$

in the notation from above. Actually, this is the predominant approach in the literature and will be referred to with MKL in the remainder of this work. However, MKL also comprises non-linear and data dependent kernel combinations and can be grouped according to various criteria [Gönen and Alpaydin, 2011]. In addition to the functional form of the kernel combination these criteria also include the target function for optimisation (e.g., structural risk minimisation), the training method (e.g., simultaneous or iterative approach), and the base learner (e.g, regularised least squares regression or support vector regression) [Gönen and Alpaydin, 2011]. As already mentioned above, we focus on structural risk minimisation approaches which learn a linear combination of kernel functions. For the sake of convenience, we will use the term *kernel linear combination* below.

Lanckriet et al. [2004b] firstly came up with the idea of combining kernels via a kernel combination to prevent the choice of a particular kernel function. Instead of considering the whole kernel function, they applied a transductive classification approach and learned the linear combination of the kernel's Gram matrices. The corresponding optimisation problem with ℓ_1-norm regularisation of the linear coefficients turned out to be a *quadratically constrained quadratic program* which becomes intractable if the number of instances or kernels become large [Rakotomamonjy et al., 2008] and can be solved with techniques of *semi-definite programming* (for more details also confer Section 2.5 below). Interestingly, early work on MKL was already applied in the biochemical domain, e.g., for protein classification tasks [Lanckriet et al., 2004c,a]. Bach et al. [2004] presented a reformulation of Lanckriet et al. [2004b]'s problem version using both the ℓ_1- and the ℓ_2-norm such that a *sequential minimal optimisation* approach could be employed for its solution. In order to apply MKL for large datasets and many kernels, Sonnenburg et al. [2006] found another reformulation as *semi-infinite linear program*. Rakotomamonjy et al. [2008] used an iterative approach and the *SimpleSVM* algorithm to solve MKL in an ℓ_2-norm regularised variant and call their efficient and well-performing approach *SimpleMKL*. SimpleMKL works by minimising the primal problem which, in contrast to the dual problem, is differentiable. In addition to classification, SimpleMKL can also be applied to regression, one-class, and multi-class classification. Another iterative ℓ_2-regularisation variant of MKL for regression was proposed and investigated by Cortes et al. [2009]. Finally, the most general case in this list of MKL variations was investigated by Kloft et al. [2009, 2011] in form of an ℓ_p-norm regularisation of the kernel linear coefficients in the minimisation objective for $p \geq 1$. Vishwanathan et al. [2010] showed how this general MKL formulation can be solved via sequential minimal optimisation. In addition to aspects of efficiency and performance also learning theoretical properties of MKL have been investigated, e.g., generalisation bounds in terms of the *Rademacher complexity* [Kloft et al., 2011, Cortes et al., 2010].

1.2.4 Co-Regularisation

Multi-view learning is the intention to profit from the simultaneous application of different data representations and involved information content without the need to decide in favour for one particular view or kernel function. We introduced MKL as the first important class of multi-view learning algorithms above. The class of multi-view algorithms we introduce in the present section includes unlabelled data in order to compensate for a small number of labelled examples. Co-regularisation aims at maximising the prediction agreement with respect to the labelled set of instances and minimising the disagreement for the unlabelled set. Whereas MKL comprises supervised approaches, *co-regularisation* is a technique of semi-supervised learning. More precisely, a predictor function for every single view is learned simultaneously such that both the empirical risk for each view predictor and the pairwise prediction differences with respect to different views for unlabelled instances are minimised. That means to solve

$$\min_{f_1,\ldots,f_M} \sum_{v=1}^{M} \mathcal{R}(f_v) + \sum_{u,v=1}^{M} \hat{\mathcal{R}}(f_u, f_v),$$

where $f_1, \ldots, f_M$ are predictor functions that correspond to M different views or kernels and $\mathcal{R}$ and $\hat{\mathcal{R}}$ are appropriately defined risk functionals. Because of the correspondence

of views and kernels we will discuss co-regularisation algorithms again in the context of kernel methods.

The idea of co-regularisation for regression originates from the concept of *co-training* that was introduced for classification by Blum and Mitchell [1998]. Co-training was originally developed for the scenario of two views and only few labelled training data. The two corresponding view predictors trained from the labelled examples in each view should successively be boosted via additional examples. These arise from unlabelled instances that obtained their labels from the respective other view predictor function. Thus, the compatibility and independence assumption for co-training are an implementation of the consensus and complementary principle of MVL [Nigam and Ghani, 2000, Dasgupta et al., 2002, Balcan and Blum, 2005, Leskes, 2005, Sridharan and Kakade, 2008]. Zhou and Li [2005] came up with a single-view variant of co-training that in a sense can be regarded as the bridge to co-regularisation for regression. More precisely, they trained two *k-nearest neighbour* predictors for regression that used different metrics instead of different views and improved their performance utilising unlabelled examples. Sindhwani et al. [2005] presented a multi-view co-regularisation approach for regression (or *co-regression*) which included the predictions for unlabelled instances directly in the global optimisation objective. Brefeld et al. [2006] found an analytic solution for their *co-regularised least squares* algorithm instead of an iterative description. A bound on the Rademacher complexity of the respective co-regularised function classes was proven by Rosenberg and Bartlett [2007]. Sindhwani and Rosenberg [2008] deduced an algorithm from the originally multi-view objective for co-regularisation with the properties of a single-view approach. As multiple languages constitute multiple views on data very naturally, Wan [2013] successfully applied co-regression for cross-language review rating.

As mentioned already, co-regularisation is one way to include unlabelled instances in addition to labelled examples into the training procedure for the prediction model which is commonly known as semi-supervised learning. *Co-regression* denotes co-regularisation approaches for regression tasks. As labelled data are available, semi-supervised learning can be considered a special case of supervised learning [Chapelle et al., 2006]. An overview of semi-supervised methods was presented by Zhu [2006]. Graph-based methods, the *expectation-maximisation* algorithm [Dempster et al., 1977], and the *transductive support vector machine* [Joachims, 1999] are examples for the variety of semi-supervised algorithms that do not base upon co-regularisation. *Support vector regression* (SVR) and *regularised least squares regression* (RLSR) play a central role in the present thesis. Semi-supervised variants of support vector machines and least squares regression can be found in the literature already for different learning scenarios. Semi-supervised variants of *support vector classification* were considered by Bennett and Demiriz [1998], Chapelle et al. [2008], Kondratovich et al. [2013], whereas Zhou and Li [2005], Wang et al. [2010a], Xu et al. [2011] introduced semi-supervised SVR in the one-view scenario. Sun [2011] and Farquhar et al. [2005] came up with support vector classification using multiple views. Also structured output support vector machines were investigated in the multi-view setting of Brefeld and Scheffer [2006]. A co-regularised variant of RLSR was introduced by Brefeld et al. [2006].

1.2.5 Projection-Based Learning

Projection-based learning (also referred to as *subspace learning*) refers to a large number of algorithms with numerous applications both in the single- and multi-view learning

scenario. For this reason, we give a short summary of the intentions of projection-based learning and the classes of comprised multi-view learning approaches. The idea behind approaches that utilise projections of the respective data instances is that a high-dimensional feature representation of data might be redundant such that the true underlying information is smeared over an unnecessary large number of variables which in turn complicates calculations and storage. Therefore, the aim of (multi-view) projection-based learning can either be the pure compression of the data representation (*dimensionality reduction*) or the enhancement of the learning result or both at once. The prime example of multi-view projection-based algorithms that can be used for both dimensionality reduction and prediction tasks is *canonical correlation analysis* (CCA) [Hardoon et al., 2004, Kakade and Foster, 2007, Foster et al., 2008]. Comparable to MKL and co-regularisation, the usage of multiple views in projection-based learning offers a broad spectrum of information without the need for an optimal view choice. Again the correspondence between kernel functions and views make projection-based learning an important branch of kernel methods.

According to Xu et al. [2013], subspace learning can be categorised into *CCA-based algorithms*, *multi-view Fisher discriminant analysis*, *multi-view embedding*, *multi-view metric learning*, and *latent space models*. CCA was introduced by Hotelling [1936] and intends to identify common latent relations between different data representations [Hardoon et al., 2004, Welling]. Therefore, CCA aims at projections P_1, P_2 of two views of data $\Phi_1(x)$ and $\Phi_2(x)$ such that the mapped vectors correlate maximally, i.e., for

$$\max_{P_1,P_2} \operatorname{Corr}(P_1^T \Phi_1(x), P_2^T \Phi_2(x)),$$

where $\operatorname{Corr}(a,b) = \operatorname{Cov}(a,b)/(\sqrt{\operatorname{Var}(a)}\sqrt{\operatorname{Var}(b)})$. In contrast to CCA which is an unsupervised method, kernel Fisher discriminant analysis [Mika et al., 1999] finds a projection of data such that the geometric class mean differences are maximised and the respective class variances are minimised. Diethe et al. [2008] generalise this approach to multiple views. Embedding and metric learning deliver further supervised and unsupervised projection approaches for multi-view data to lower dimensional feature spaces which are optimal to some objective criteria. In contrast, latent space models focus on the latent relationships between different views for learning [Xu et al., 2013].

In Chapter 5 of the present thesis, projection-based methods are considered to solve an unsupervised problem. More precisely, the aim is to find labels of instances with respect to a target for which no labelled training examples are available. However, for the same or a very similar learning task and related targets there are labelled examples available. Projection-based approaches turn out to be very useful in this scenario. In the first instance, the projections serve as a transfer tool for label information from one target to another. In order to solve this problem from transfer learning [Pan and Yang, 2010], scalar projections as well as a variant of *principal component analysis* (PCA) [Schölkopf et al., 1997] are applied. PCA finds a projection of the data feature representation in an unsupervised manner such that the mapped variables exhibit maximal variance from the original data points. The knowledge-based PCA variant of Oglic et al. [2014] includes further information of the learning domain in form of must-link and cannot-link constraints. The definitions of the two novel algorithms introduced in Chapter 5 are not based on multiple views in the first place. However, we utilise *Johnson-Lindenstrauss* (JL) projections [Dasgupta and Gupta, 2003] to include multiple views in the model training phase in the empirical analysis. A multi-view PCA approach for transfer learning was presented by Ji et al. [2011].

1.3 Affinity Prediction

As mentioned already in Section 1.1, in the present thesis we focus on a problem of *chemoinformatics* called *ligand affinity prediction*. We illustrate why affinity prediction is an important and challenging application from practice. In the subsequent main chapters we exploit the particular characteristics of the learning scenario typically accompanied with affinity prediction in order to improve existing machine learning approaches for its solution.

Ligands are small molecules that bind to proteins with a real-valued chemical affinity that we intend to predict. On the one hand, this is a fundamental learning task in practice as bindings of proteins and ligands are essential for the understanding of protein function in biological organisms [Nelson and Cox, 2001]. Proteins are crucially involved in the majority of biochemical cell processes which make them the central molecules for life besides the nucleic acids DNA and RNA (more details in Section 1.3.1). Hence, influencing proteins via ligands is one excellent starting point for *drug discovery* efforts. On the other hand, the learning scenario and the nature of typical datasets (see below) depicts an interesting setting for learning which is worth investigating, independent of affinity prediction. Indeed, other relevant applications share a similar scenario, e.g., *object detection* from different perspectives, translation based on *multilingual corpora*, and *disease diagnosis* from different physiological markers.

In concordance with the three presented classes of multi-view learning in Section 1.2, we consider three different affinity prediction variants in the main chapters below.

(i) Supervised affinity prediction: Small molecules and their affinity values with respect to a fixed protein are used to train a binding model for that protein. Different molecular fingerprint designs (see Section 1.3.4) for small molecules are available.

(ii) Semi-supervised affinity prediction: Small molecules and their affinity values with respect to a fixed protein are used to train its regression model using further small molecular compounds without known affinity as unlabelled data. Multiple molecular fingerprint formats exist. In particular, also unlabelled compounds can be represented and compared using these multiple fingerprint representations.

(iii) Unsupervised affinity prediction (orphan screening): We consider a protein for which no affinity values of small molecules are known at all. Therefore, this particular protein is called *orphan protein*. Nevertheless, labelled instances with respect to other proteins are available and used to learn an affinity model for the orphan protein. Again, for all included small molecules different fingerprint representations are available.

In the following section, we give a general introduction to the practical problem of *ligand affinity prediction* to which all methods of this thesis are oriented. We go into detail with the biochemical background, explain how molecular data instances can be displayed for learning, and place affinity prediction within the research field of chemoinformatics. Furthermore, we review existing machine learning approaches for its solution, in particular, the small number of already existing multi-view attempts in chemoinformatics. As affinity prediction seems predestined for multi-view learning because of the different representations for data instances of interest, we aim at complementing the mentioned attempts with our proposed approaches in Chapter 3, 4, and 5.

1.3.1 Biochemical Background

From a very general point of view, a ligand is a molecule that binds to another determined molecule named target which is typically much bigger than the ligand itself. Usually, the binding is non-covalent and reversible. There is a large variety of target-ligand relationships. For example, a ligand can be a single iron ion that builds a complex with the biomolecule haemoglobin, the red blood pigment. In contrast, large DNA-binding proteins are also referred to as (DNA) ligands. However, in the present thesis we consider ligands of proteins. The ligands are molecules with usually organic scaffolds and a low molecular weight. Proteins themselves are high-molecular chains of amino acids that form a 3-dimensional structure with *binding sites* or *binding pockets* for ligands that emerge from the spacial arrangement of amino acids and their respective functional groups.

Being the direct product of gene transcription and translation, proteins are crucial for the structure and functionality of life. Among others, they serve as transporters (thyroxin-binding globulin), ion channels (transmembrane sodium channel), hormones (oxytocin), receptors (G-protein coupled serotonin receptor), scaffold proteins (regulators of signal pathways), enzymes or catalysts (DNA ligase). Numerous biochemical reactions are triggered via protein-ligand bindings [Nelson and Cox, 2001]. The tendency or capacity of a ligand to stick to a certain protein is called *binding affinity* and is mainly due to intermolecular forces. The binding affinity can be expressed quantitatively via different characteristics. The most common is the half-maximal *inhibitory concentration* IC_{50} which is the concentration (unit molar $1M$) of ligand molecules necessary to achieve the half-maximal biological activity of the protein. The inhibitory concentration essentially quantifies the same physico-chemical property of molecules like the *inhibition constant* K_i. For reasons of statistics and scalability often

$$pK_i = -\log_{10} K_i$$

is considered. In contrast to the values of K_i, IC_{50} values depend on experimental conditions. Active compound concentrations lie in the range of nano- or micro-molars nM and μM, respectively. A small inhibition constant indicates a high binding affinity. In our experiments below we use the inhibition constant K_i as measure of ligand affinity. However, other measures such as *ligand efficiency* have been investigated in the context of affinity prediction as well [Sugaya, 2013]. The usage of the name *ligand* differs depending on the binding model into consideration. If one is interested in whether a small molecule binds or not (ligand prediction or classification model) we distinguish between *ligands* and *non-ligands*. If the focus is to determine the actual binding strength (affinity prediction or regression model) practically every small molecule is a ligand to some extent. However, if a ligand's binding strength is too small, the assigned affinity value might not be biologically meaningful anymore. In both model cases, the name *(molecular) compound* refers to a small molecule whose binding behaviour is or is not of interest or unknown.

Proteins and their derivatives are important points of contact for chemical substances in living organisms. For this reason, the active components of many drugs work as protein ligands, e.g., insulin against diabetes, beta blocker against hypertension, opioids as painkiller, thyroxine against hypothyreodism, and many more. As it is the case for insulin or serotonin, the ligands can be small proteins or amino acids themselves. Though,

the molecular fingerprints used in this thesis are only applicable to small molecular compounds and not to protein or polypeptide macromolecules. Most obviously, the search for novel and effective protein ligands for drug discovery depicts an important real-world problem.

In practice, the protein-ligand interaction can be measured in the laboratory via screening assays. These assays were developed by researchers in order to prove and quantify a particular biochemical process, e.g., by the concentration of reaction products or by reflection and absorption effects during the process. Nowadays, large physical libraries of small molecule compounds can be tested whether they bind to proteins on large scale via *high-throughput screening* (HTS). Via HTS millions of experiments can be conducted simultaneously by robots on well plates that contain the reacting agents. The determination of K_i or IC_{50} values usually occurs in two phases. In a first phase many different potential ligands are tested with a fixed ligand concentration. The promising candidates from the first phase are then investigated under varying concentration conditions in order to estimate K_i or IC_{50} as good as possible.

1.3.2 Molecular Fingerprints and Databases

In order to include complex objects such as molecules into a mathematical algorithm it is necessary to find an informative representation that can be worked on and stored easily. Actually, dozens of file formats exist that save chemical structures together with associated properties. We restrict our presentation to the formats relevant for this thesis.

Molecular fingerprints are vectorial representation formats that take structural or physicochemical information into account. Every element of the typically high-dimensional vector represents a molecular feature in a binary, integer-valued or real-valued manner. Binary features indicate the presence or absence of a structural property whereas integer values appear if structural features are counted [Heikamp, 2014]. Physico-chemical properties, such as molecular weight, solubility, lipophilic or hydrophilic character, and total polar surface area (which apart from the molecular weight are also a result of the two-dimensional composition and three-dimensional conformation of the molecule) are expressed with real values [Sugaya, 2014]. The majority of fingerprints are based on the interpretation of molecules as undirected graphs with labelled vertices (atoms with labels *C*, *O*, *H*, *N*, *S*, etc.) and labelled edges (chemical bonds with labels *1* for *single*, *2* for *double*, *3* for *triple*, and *a* for *aromatic*). With *bond* we refer to the connection between atoms in molecules, whereas *binding* relates to the docking of one molecule to another. Novel bonds can emerge from the docking process. The graph character of molecules (compare Figure 3.1) needs to be documented appropriately such that fingerprints can be determined or calculated correctly. The *structure data file* (SDF) format of a molecule is an adjacency table with label information for atoms and bonds as well as the three-dimensional relative positions of atoms. This adjacency table is sometimes complemented with a list of further properties, such as identifier strings or the molecular weight. A number of other chemical file formats exist, for example the single-line notations *simplified molecular input line entry specification* (SMILES) or *SMILES arbitrary target specification* (SMARTS)[1]. The formats can be converted into each other via software tools, e.g., via *Open Babel*[2].

[1] Daylight Theory Manual `www.daylight.com/dayhtml/doc/theory/`

[2] `openbabel.org`

A large amount of publicly available or commercial molecular fingerprints exists [Bender and Glen, 2004, Bender et al., 2009, Koutsoukasa et al., 2013]. Depending on their original purpose of use, each fingerprint is equipped with a specialised composition of structural or physico-chemical features. There are prediction tasks that can be modelled and solved well via physico-chemical properties [Liu et al., 2006]. However, for the assessment of bioactivity of molecular compounds, structural features turned out to be very successful [Bender et al., 2009]. A unique classification of structural descriptors is difficult because of the ambiguity of certain standard fingerprint formats and the lack of completeness in view of the fingerprint variety. In our experiments below we use fingerprints from the following prominent fingerprint types [Heikamp, 2014]

- *Predefined keys*,
- *Pharmacophore fingerprints*, and
- *Extended connectivity fingerprints*,

which are not unique nor complete, but sufficiently reflect the existing diversity for our purposes.

Predefined keys (or structural fingerprints) represent a collection of fixed features in a binary bit string indicating presence or absence of the respective feature in a molecule. Features are molecular patterns, such as *atom types*, *rings* and *aromatic systems*, *functional groups*, or other substructures. The 166-bit fingerprint *Maccs* was originally designed by Molecular Design Limited (MDL) Inc. Information Systems [Durant et al., 2002]. The features of pharmacophore fingerprints encode two- or three-dimensional arrangements of atom types [Heikamp, 2014]. We will use the three-point pharmacophore *GpiDAPH3* that considers any set of the three possible atom types *donor*, *acceptor*, and *atom in aromatic pi system* [Bender et al., 2009]. In a sense, pharmacophore fingerprints are keyed fingerprints of fixed size as well. Extended connectivity (or combinatorial) fingerprints list circular atom environment features [Rogers and Hahn, 2010, Heikamp, 2014] strongly related to the concept of *Weisfeiler-Lehman* labels (compare also Section 3.1.5). In contrast to the first two groups, extended connectivity fingerprints are neither fixed in length nor in the set of features, but depend on the features that are found in the precise set of compounds at hand. *ECFP4* and *ECFP6* consider a radius (or *Weisfeiler Lehman* depth) of 2 or 3, respectively. All utilised fingerprints in the experimental sections can be calculated with *Molecular Operating Environment* (MOE)[3] or Pipeline Pilot (PP)[4] software.

Affinity-annotated compounds can be found and composed to appropriate datasets from different publicly accessible databases. *BindingDB*[5], *ChEMBL*[6], and *PubChem*[7] contain thousands of ligands in standard chemical file formats (compare above). In addition to affinities against protein targets obtained from bioassays, the databases contain further information on the comprised compounds such as *absorption, distribution, metabolism, and excretion* (ADME) parameters [Fröhlich et al., 2005, Heikamp, 2014]. Structural, functional, and relational information on proteins are contained, e.g., in the *protein data bank* (PDB)[8] database for biological macromolecules. The amino acid sequence,

[3] Chemical Computing Group `www.chemcomp.com`
[4] BIOVIA `www.3dsbiovia.com`
[5] `www.bindingdb.org`
[6] `ww.ebi.ac.uk/chembl/`
[7] `https://pubchem.ncbi.nlm.nih.gov/`
[8] `www.rcsb.org/`

connectivity of peptide chains, and 3D structure of proteins can be useful in docking algorithms (see Section 1.3.4) and for the quantitative assessment of the relation between protein targets (compare Chapter 5).

1.3.3 Kernels for Molecular Graphs

The concept of kernel functions already appeared in Section 1.2.1 on multiple views and Sections 1.2.3, 1.2.4, and 1.2.5 on the classes of multi-view learning. Although the precise definition will follow below in Section 2.5 on machine learning basics, for the time being we understand kernel functions as generalised scalar products that deliver a quantitative assessment of the similarity between learning objects. Furthermore, different kernels relate to different feature representations of these objects. In this sense, molecular fingerprints induce a kernel function and vice versa. Kernel functions are applied successfully in a variety of learning algorithms (the so-called kernel methods) such as the ones investigated in the present thesis. Frequently, complex structured data objects are in the focus of a certain learning task, e.g, trees, graphs, strings or whole text passages [Collins and Duffy, 2001, Ralaivola et al., 2005]. If an appropriate kernel function for complex inputs is available, kernel methods can be utilised to solve the learning task at hand. For obvious reasons, molecules can be identified with graph objects, where atoms and bonds are labelled vertices and edges, respectively. A kernel function as measure of similarity for molecular graphs enables the comparison between different compounds. This in turn forms the basis for the application of the molecular similarity principle, i.e., to assume and to exploit that structurally similar compounds also exhibit similar properties [Bender and Glen, 2004]. The application of the similarity principle finally leads to a quantitative structure-activity relationship model (compare Section 1.3.4 below) for the prediction of the molecule's behaviour. However, also other objects that are in the focus of machine learning attempts have graph structure, for example, the scheme of biochemical pathways, the network of hyperlinks or citations, and the connections in social networks [Kondor and Lafferty, 2002].

A literature survey of Gärtner [2003] on kernels for structured data appeared nearly concurrently to the emergence of modern virtual screening and affinity prediction. Gärtner et al. [2003] defined graph kernels by counting walks with equal start and end node and of the same length or common labelled sequences. Kondor and Lafferty [2002] developed and applied so-called diffusion kernels based on matrix exponentiation. In addition to linear patterns like paths or trees, the cyclic pattern kernel [Horváth et al., 2004, Horváth, 2005] also took cyclic patterns within the graph into account to assess the similarity between molecules. With respect to the decomposition of graphs into cyclic and acyclic components we refer to Section 3.1.3 below. In their overview of graph kernels and machine learning approaches in chemical informatics, Ralaivola et al. [2005] suggest to apply conventional and novel fingerprinting techniques based on graph structures in connection with the *Tanimoto*, *MinMax*, and *Hybrid* kernel function. An optimal assignment kernel was presented by Fröhlich et al. [2005], who suggested to firstly compare vertices and edges between direct neighbours within a certain neighbourhood radius and then calculate the actual assignment by maximising the vertex and the edge similarities. The application of support vector classification or regression in combination with kernels for molecular graphs became more and more important for virtual screening and has been investigated by a number of authors. For example, Geppert et al. [2008] utilised the linear kernel and standard molecular fingerprints for ligand prediction with support

vector machines. 2D walk and tree kernels [Gärtner et al., 2003, Ramon and Gärtner, 2003, Kashima et al., 2004] and 3D pharmacophore kernels [Mahé and Vert, 2009] were considered in this scenario as well. Gaüzère et al. [2014] performed classification and regression experiments using small subgraphs (treelets) as well as cyclic and chiral information. Finally, the target-ligand kernel is a product kernel for molecular compounds and considered protein targets. It will be discussed in detail in Section 5.2 below.

1.3.4 Virtual Screening

Chemoinformatics applies techniques and knowledge from computer science and chemistry in order to extract, process, and extrapolate meaningful information from chemical structures. Pioneering work in that research field was done by Hansch et al. [1962] already half a century ago. Chemoinformatics considerably gained in importance in recent years because of the rapidly increasing amount of data with complex chemical information and machine learning algorithms capable of detecting non-linear patterns [Lo et al., 2018]. HTS techniques already work very fast and efficient in practice but are still time-consuming and cost-intensive, in particular, if one keeps in mind that there are quasi infinitely many conceivable compound candidates. Additionally, just like all practical experiments it might produce incorrect outcomes such as false positives or false negatives [Heikamp, 2014]. Although it cannot replace practical experiments completely, *(in-silico) virtual screening* is the attempt to simulate HTS with computational methods virtually without any expenditure of time and money. It denotes the screening of large molecular databases (see Section 1.3.2 above) for compounds with a certain bioactivity and is one of the central applications in chemoinformatics [Shoichet, 2004, Irwin, 2008]. A successful application of such techniques would support the empirical screening process by preliminary selection tests, prioritisation of chemical candidates, and finally also by reducing animal experiments [Maunz and Helma, 2008].

Virtual screening can be divided into *ligand-based virtual screening* and *structure-based virtual screening approaches*. Structure-based approaches calculate a *scoring function* [Ain et al., 2015] via computational methods. Amongst others, these methods comprise SVR [Li et al., 2011], *random forests* [Ballester and Mitchell, 2010, Liu et al., 2013, Li et al., 2014], *partial least squares regression* and *artificial neural networks* [Wang et al., 2010b, Speck-Planche and Cordeiro, 2014, Stepniewska-Dziubinska et al., 2018, Ferreira and Andricopulo, 2019, Tetko and Engkvist]. The scoring function is derived from 3D structure information of the involved molecules [Ortiz et al., 1995, Zhou and Skolnick, 2012] and models the protein-ligand docking process via thermodynamical statistics. In contrast, ligand-based approaches utilise information of known ligands and their affinities to obtain a binding model (more details below). For the reasons explained above, virtual screening plays an important role for the discovery and design of novel *lead* compounds for drugs as a central concern in medicine [Jacob et al., 2008]. In Chapter 3, 4, and 5 we focus on the affinity prediction task with ligand-based approaches in order to obtain a so-called *quantitative structure activity relationship* (QSAR) model [Maunz and Helma, 2008, Cherkasov et al., 2013].

Ligand-based virtual screening for the prediction of protein-ligand bindings is based on the *molecular similarity principle* or *similarity property principle* (SPP), which states that similar molecules are likely to have similar activities [Bender and Glen, 2004, Bender et al., 2009, Geppert et al., 2010]. Apart from this commonality, the precise approaches

differ in computation method and learning task. An overlap of methodologies and tasks makes a unique structuring of virtual screening very difficult.

- The inter-molecular relation can either be considered as a classification problem, i.e., whether a ligand binds to a protein or not which we will refer to as *ligand prediction*. More specifically, it can also be modelled as a regression problem with positive real-valued affinities or related quantities (compare Section 1.3.1) which we will denote *affinity prediction*. Although the classification model is a strong simplification of the reality, this approach predominates the literature [Burbidge et al., 2001, Erhan et al., 2006, Geppert et al., 2008, Jacob et al., 2008, Geppert et al., 2009, Ning et al., 2009, Wassermann et al., 2009a, Ullrich et al., 2010, Vogt and Bajorath, 2010, Sugaya, 2013]. Only few publications go beyond the sheer suggestion of ligand candidates with a regression approach [Bock and Gough, 2002, 2005, Sugaya, 2014, Balfer and Bajorath, 2015]. In the present thesis we focus on the regression task of affinity prediction.

- Another distinctive feature strongly related to the modelling of the binding process is the type of the actual outcome of the prediction model. In addition to the classification or affinity annotation of single compounds, an order or ranking of a set of compounds [Bock and Gough, 2002, Geppert et al., 2008] or a compound representation as structured output [Ullrich et al., 2010] can be outcomes as well.

- With respect to computation, the majority of algorithms can be assigned to *similarity search* [Willett et al., 1998, Sheridan and Kearsley, 2002, Willett, 2006, Geppert et al., 2010, Vogt and Bajorath, 2010] or classical *machine learning* [Burbidge et al., 2001, Geppert et al., 2008, Mahé and Vert, 2009] using molecular fingerprint representations, where the transition between both fields are smooth. Similarity search denotes a ranking of compounds with respect to their similarity value (*Tanimoto coefficient*) compared to one or multiple active reference compounds [Geppert et al., 2008]. More details on machine learning and molecular fingerprints can be found in Chapter 2 and Section 1.3.5 below.

- The availability and amount of labelled compounds with respect to proteins induces a grouping of applied algorithms. There are supervised algorithms that rely on sufficient labelled training compounds [Lo et al., 2018]. So-called semi-supervised algorithms operate in the learning scenario of few labelled training examples and many unlabelled instances [Ning et al., 2009, Kondratovich et al., 2013]. Also unsupervised algorithms play an important role. Actually, *orphan screening* denotes the search for ligands of protein targets without known training ligands [Geppert et al., 2009, Wassermann et al., 2009a, Ullrich et al., 2010].

- Finally, algorithms can be categorised depending on whether they investigate the compound activity with respect to one single target [Geppert et al., 2008] or against multiple targets. The latter is strongly related to the questions about the binding preference of a compound to one particular target over other targets, also known as *selectivity* [Wassermann et al., 2009b, Heikamp, 2014].

Apart from ligand or affinity prediction there are other learning tasks in the field of virtual screening using similarity search and machine learning. Strongly related is the detection of bioactivities with *support vector classification* (SVC) or SVR models based on *ligand efficiency* indices [Sugaya, 2013, 2014]. These are alternative quantities to

assess a compound's drug potency. *Scaffold hopping* is another task connected with the identification of ligand affinities. It denotes the identification of structurally different active molecules [Wassermann et al., 2009a]. For the development of novel drugs it is rather less interesting to find molecules very similar to already known ligands. Instead, molecules with a different molecular scaffold are more likely to be candidates for further research [Heikamp, 2014]. *Binding sites* or *pockets* are the protein substructures where the actual interaction with the ligand occurs. The identification of similar binding sites via protein structure-based fingerprints can be used to find structurally diverse ligands from binding information in databases [Wood et al., 2012]. The protein targets themselves can be categorised into classes depending on their functionality. Machine learning algorithms such as *support vector machines* (SVMs) [Burges, 1998] were applied successfully to the classification of database proteins and, hence, to protein function prediction [Cai et al., 2003, Lanckriet et al., 2004c, Tsuda et al., 2005]. Last but not least, also DNA is in the focus of virtual screening. *Genes* are DNA sectors which are correlated with a certain protein expression activity and have been classified successfully via machine learning algorithms [Vert and Kanehisa, 2002].

As we have shown above, virtual screening techniques can be employed in a wide range of applications. However, it exhibits intrinsic limitations [Shoichet, 2004], particularly for affinity prediction.

1. Firstly, a variety of molecular fingerprints exists and it is neither obvious which one to choose for a given task, nor whether the representation comprises the features necessary to explain the desired activity [Sheridan and Kearsley, 2002, Heikamp, 2014].

2. Similar to an inapt molecular representation, also scaffold hopping [Geppert et al., 2010, Heikamp, 2014] and *activity cliffs* [Medina-Franco et al., 2009, Balfer and Bajorath, 2015] can lead to a failure of the SPP. Activity cliffs are in a sense the opposite of scaffold hopping as they describe big changes in activity for small structural differences.

3. If two ligands bind to the same protein target but the binding occurs at two different binding sites, e.g., in the case of allosteric inhibition [Nelson and Cox, 2001], the SPP will not be helpful to infer about the affinity from one ligand to another [Heikamp, 2014].

4. A more general problem of the prediction scenario is the typically small number of known ligands and corresponding affinities [Geppert et al., 2008, Jacob et al., 2008]. Given a considered protein target, it is expensive to obtain practically verified labelled training examples.

5. Activities against multiple protein targets or the search for target-selective compounds represent further challenges in the context of affinity prediction [Wassermann et al., 2009a, Heikamp, 2014].

6. Orphan screening as an unsupervised learning problem represents a challenge to computational methods per se.

We address the proposed limitations of the affinity prediction task in items 1., 4., and 6. with our approaches proposed in the main chapters below.

1.3.5 State-of-the-Art Ligand-Based Approaches

Affinity prediction as an instance of ligand-based virtual screening is in the focus of the present thesis with respect to the practical application. For this reason, we dedicate this chapter to the machine learning methodologies applied in this field so far. The state-of-the-art of the recent two decades has been reviewed yet [Burbidge et al., 2001, Geppert et al., 2010, Cherkasov et al., 2013, Heikamp, 2014, Heikamp and Bajorath, 2014, Lo et al., 2018]. In this section, we summarise the literature on ligand-based approaches which is relevant for the present thesis.

Virtual screening can be considered a major application for SVMs nearly since their breakthrough at the end of the last century. In particular, its explicit classification variant *support vector classification* (SVC) as well as SVM-based ranking strategies and similarity search [Sheridan and Kearsley, 2002, Willett, 2006, Vogt and Bajorath, 2010] play a predominant role in the literature (compare also Section 1.3.4 above). For historical reasons, SVM is frequently used as synonym for SVC, although in the strict sense it is a generic term for a class of algorithms. Sugaya [2013] performed SVC using training data based on a threshold on ligand efficiency indices (as an alternative to K_i- or IC_{50}-values) to predict ligands. Geppert et al. [2008] investigated SVC-based and different similarity search-based ranking strategies to find new ligands at the presence of various known active compounds. SVC and similarity search methods were applied both for ligand prediction in the standard supervised case with labelled training data and for orphan screening by Geppert et al. [2009]. Multi-class SVC experiments were conducted by Wassermann et al. [2009b] who categorise ligands into selective, non-selective, and inactive ones. For example, the experiments with a *k-nearest neighbour* approach in Geppert et al. [2008] showed that there is a smooth transition from similarity search to machine learning algorithms like SVC. *Decision trees* and *Bayesian classifiers* [Burbidge et al., 2001, Geppert et al., 2010] are examples of alternative machine learning algorithms that have been applied for the prediction of ligands as well. The learning task of ligand prediction was also already handled in a *structured output prediction* scenario where the ranking of compounds itself was the output of the algorithm [Ullrich et al., 2010]. For the quality assessment of classification models in virtual screening, performance measures like sensitivity, specificity, accuracy, and recovery rate are typical [Geppert et al., 2008, 2009, Sugaya, 2013, Heikamp, 2014].

In addition to ligand prediction as a classification approach, Fröhlich et al. [2005] considered affinity prediction with a regression model to describe the protein-ligand complex. Affinity prediction references in the regression scenario are still rare taking into account the overall number of publications in the field of ligand prediction and virtual screening. However, the majority of authors use SVR and variants for the prediction of ligand affinities [Bock and Gough, 2002, Liu et al., 2006, Sugaya, 2014]. Balfer and Bajorath [2015] studied inherent problems of SVR models, in particular to predict high affinities and discontinuities in activity landscapes (activity cliffs). Maunz and Helma [2008] predicted toxic activities with local SVR models. Besides SVR also least squares regression variants [Ding et al., 2013, Abbasi et al., 2017] and random forests [Abbasi et al., 2017, Kundu et al., 2018] are applied frequently in ligand-based approaches, but were not able to replace SVR for the prediction of affinities so far. Recently, various types of artificial neural networks were applied successfully to affinity prediction [Jiménez et al., 2018, Öztürk et al., 2018, Ferreira and Andricopulo, 2019, Tetko and Engkvist]. Artificial neural networks and deep learning will prospectively gain more and more importance for chemoinformatics in the near future. Frequently, variants of the *mean squared error*

are applied as performance measure for the considered regression task [Bock and Gough, 2002, Liu et al., 2006, Maunz and Helma, 2008]. To a small extend, also alternative evaluation criteria like *Pearson's correlation coefficient* [Sugaya, 2014] or R^2-values [Balfer and Bajorath, 2015] are investigated.

All of the methods mentioned so far utilise fingerprints for the representation of the small molecular compounds (see Section 1.3.2). A big number of fingerprint descriptors with structural and physico-chemical molecule features exist [Bender et al., 2009] that can be applied for virtual screening and related tasks [Wassermann et al., 2009b, Sugaya, 2013]. The choice of the optimal representation of molecular instances for the respective learning problem is a central issue in virtual screening and strongly relates to the research on fingerprint design [Heikamp and Bajorath, 2012], recombination [Nisius and Bajorath, 2009, Nisius, 2010, Nisius and Bajorath, 2010], compression and feature selection [Geppert et al., 2010]. Fingerprints can easily be included in kernel methods which are able to detect non-linear dependencies in data. To this aim, a variety of kernel functions (compare Section 2.5 and Table 2.1) have already been applied to molecular instances in this field so far, for example, the linear, Tanimoto, Gaussian, and the polynomial kernel [Liu et al., 2006, Wassermann et al., 2009a,b, Balfer and Bajorath, 2015]. Again, the application of the Tanimoto coefficient as a way to express similarities for compound ranking approaches [Geppert et al., 2008, Heikamp, 2014] shows close connections between machine learning and similarity search. According to the *kernel trick* the explicit knowledge of the corresponding feature representation is not an issue as long as the calculation of kernel values for involved learning objects is possible. Fröhlich et al. [2005] and Mahé and Vert [2009] exploited this property of kernel methods with their alignment and structure kernel algorithms. Kernel functions were not only applied to small molecular compounds as potential ligand candidates but also to proteins. The target-ligand kernel appearing in Jacob et al. [2008], Jacob and Vert [2008], Vert and Jacob [2008], Geppert et al. [2009], Wassermann et al. [2009a] is a product kernel for proteins and small molecular compounds for which different kernel functions for proteins are investigated.

A special case of ligand or affinity prediction in the context of virtual screening is orphan screening. In contrast to the problems and solution methods above, orphan screening denotes the search for novel ligands of proteins (and their respective affinities) for which no labelled binding information is available at the time of prediction. The most prominent approach in this scenario is the application of *target-ligand kernels* in combination with SVMs [Jacob et al., 2008, Jacob and Vert, 2008, Geppert et al., 2009, Wassermann et al., 2009a]. Here, labelled information of related targets are included in the learning process via a product kernel for targets and ligands. Ning et al. [2009] also investigated the usefulness of unlabelled compounds in a semi-supervised approach, as well as *multi-task learning* and *multi-ranking*. Erhan et al. [2006] applied artificial neural networks and collaborative filtering to orphan screening. Only little effort was done so far in the research field of ligand affinity prediction in an orphan screening setting [Bock and Gough, 2005].

1.3.6 Multiple Views in Chemoinformatics

Multi-view learning is not a completely new learning paradigm in chemoinformatics. For this reason, we present existing approaches in this section. However, to the best of our

knowledge it was not applied to virtual screening and in particular to affinity prediction at all.

A transductive multi-view approach was applied to learn the kernel linear coefficients corresponding to MKL (see Section 1.2.3) and the desired labels for protein function prediction [Lanckriet et al., 2004c] and transmembrane protein identification [Lanckriet et al., 2004a]. In this context, kernels based on protein sequence alignments, protein-protein interactions, and gene expression measurements served as views. Tsuda et al. [2005] solved a protein classification problem by learning a linear combination of protein networks. The multiple protein networks put different protein properties into relation. In a sense, they represent a sparse variant of Gram matrices. The algorithm is a multi-view variant of graph-based learning and works more efficient than the MKL variant of Lanckriet et al. [2004b] as, in contrast to the Gram matrix of kernel functions, networks are in general sparse. Gaüzère et al. [2014] performed MKL using a kernel based on a bag of small molecular subgraphs. The inhibitory potential of *short interfering RNA* in a viral gene expression pathway is measured via its efficacy, which is similar to the affinity of general molecular compounds. Qiu and Lane [2008] utilised an MKL regression approach to predict this efficacy value. Vert and Kanehisa [2002] improved a gene classification task based on microarray RNA expression data as one representation of instances with further information on the respective genes. To this aim, they used a graph of genes extracted from a chemical reaction database where two genes are connected if they catalyse subsequent reactions in a biochemical pathway. They applied *kernel canonical correlation analysis* to perform microarray feature extraction directed from the gene representation in the reaction graph.

1.4 Thesis Outlook

In the present chapter we introduced topic and contribution of this thesis and positioned it both within the research field of multi-view machine learning and the chemoinformatics application of ligand affinity prediction. The second preliminary chapter is followed by three main chapters dedicated to affinity prediction in different learning scenarios and adequate machine learning approaches to solve them.

In Chapter 2 we give an overview of relevant concepts for the thesis from the research field of machine learning. More precisely, here we deliver the foundation to investigate and later apply multi-view kernel algorithms to learn a real-valued prediction function. The section on notation where we emphasise the indexing with respect to different views as well as labelled and unlabelled data, is followed by the presentation of how the learning procedure theoretically occurs. We explain the different phases of learning and how complexity measures from learning theory can be used to control the generalisation performance of function classes and algorithms. We introduce techniques from convex optimisation which will be the basis for the solution of the optimisation problems in the main chapters. Furthermore, we define the central concept of a kernel function and corresponding function spaces. We discuss important properties of kernels and illuminate how kernels are related to feature representations of learning objects. Chapter 2 finishes with kernelised single-view regression and dimensionality reduction techniques.

Chapter 3 addresses affinity prediction in a supervised scenario with multiple kernel learning under particular consideration of the graph structure of learning objects. It is

based on our publication *Ligand Affinity Prediction with Multi-Pattern Kernels* [Ullrich et al., 2016b] at the *Discovery Science* conference in 2016. At first, we review basics from graph theory and comment on how graph patterns and labels that cleverly condense information on graphs were used to define existing kernel functions for graphs. We go a step further and construct a kernel linear combination of popular graph kernels calculated for different depths of a high-level graph labelling procedure. This multi-pattern kernel is canonically included in objectives from multiple-kernel learning that will be presented here. In the subsequent empirical section we investigate the role of different graph patterns for their respective predictive ability. We evaluate and discuss the usefulness of the multi-pattern kernel for supervised affinity prediction and an automatic weighting of various kernel functions.

The subsequent main chapter focuses semi-supervised affinity prediction due to our workshop paper *Ligand-Based Virtual Screening with Co-Regularised Support Vector Regression* [Ullrich et al., 2016a] at the *International Conference on Data Mining* in 2016 and our publication *Co-Regularised Support Vector Regression* [Ullrich et al., 2017] at the *European Conference on Machine Learning* in 2017. Via the approach of co-regularisation unlabelled data in addition to labelled data are used for the prediction of ligand affinities to compensate for a small number of labelled examples. Co-regularisation allows for a comparison of predictions for unlabelled objects utilising multiple data representations. In Chapter 4 we propose a novel co-regularised kernel algorithm and variants of it. The different versions either arise from modified risk functionals for labelled and unlabelled instances that lead to a diminished number of variables, or from a fused kernel function which provides the co-regularised algorithm with a single-view character. Finally, we empirically evaluate the novel algorithms by a comparison to existing co-regularised approaches and other single- and multi-view approaches for the practical task of affinity predictions.

We consider an unsupervised learning scenario for affinity prediction in Chapter 5 which we firstly presented in our workshop paper *Corresponding Projections for Orphan Screening* [Giesselbach et al., 2018] at the *Neural Information Processing Systems* conference in 2018. Here, projection-based approaches are applied to transfer information from related learning tasks with training data to an unsupervised prediction problem. Initially, we review an important baseline from chemoinformatics using product kernels for a prediction model. Afterwards we introduce two novel kernel algorithms to infer a predictor for a protein without labelled training examples. The first of which directly relates similarities in the space of tasks and hypotheses and the second applies a dimensionality reduction technique to fuse task and hypothesis space. In contrast to the linear combination and co-regularisation approach from Chapters 3 and 4, these algorithms can be formulated and applied in a single- and multi-view version. Subsequently, we practically evaluate the approaches for the unsupervised affinity prediction task.

Finally, in Chapter 6 we conclude the thesis with a summary of the achievements and a list of future directions. The conclusion is followed by an appendix chapter where we present long technical proofs, detailed information on the used datasets, and supporting pseudocode formulations for a number of the applied algorithms. Closing this outlook section and preceding to the main part of the thesis, we determine conventions with respect to citation. We give references to definitions, lemmas, and theorems either directly or in the immediate neighbourhood. For general concepts, like *empirical risk minimisation* or *positive semi-definiteness*, we chose a reference with an appropriate formulation and context. The content based on our publications [Ullrich et al., 2016b,a,

2017] and [Giesselbach et al., 2018] is presented in an elaborate version. A detailed description can be found in the main chapters below. The Lemmas 5.5, 5.7, 5.9, and 5.11 as well as Definitions 5.6, 5.8, 5.10, and 5.12 are novel.

Chapter 2

Machine Learning Preliminaries

Machine learning is the process of drawing conclusions and making predictions from known information contained in data. It is strongly related with *data mining*, the actual data exploration and extraction process of unknown information. Data mining applies techniques from machine learning. Machine learning, inversely, requires data mining techniques and results. Both can be seen as a subdiscipline of *knowledge discovery*, a general term for the amplification of knowledge by computer programs [Mitchell, 1997, Hastie et al., 2001, MacKay, 2003, Flach, 2012]. As an important computer science discipline, machine learning arised from the efforts and findings in connection with *artificial intelligence* that in its modern sense bases upon the fundamental work of Turing [1950] and others in theoretical informatics. It has numerous practical applications and utilises theory and methods from various mathematical fields, such as optimisation, statistics and probability theory, functional analysis, and complexity theory. Navigation systems, speech and object recognition, fraud detection, and search engines are only a few example tasks for machine learning techniques. Since the end of the 20th century, machine learning is flourishing and getting more and more important because of the establishment of the world wide web and sophisticated electronic devices with computer technology in basically every sector of life. The coherent appearance of huge amounts of data and the increasing potential of computers in calculation and storage poses a big chance and a big challenge to machine learning.

Machine learning can be categorised generally via different criteria which will be focused in Section 2.2 below. Generally, the common aim of machine learning attempts is to learn a predictor function (or *model*) f that maps objects from an instance space $\mathcal{X}$ into a label space $\mathcal{Y}$

$$f : \mathcal{X} \to \mathcal{Y}, \tag{2.1}$$

such that the error that can be expected for future predictions is minimal. The elements of $\mathcal{X}$ are the learning objects, i.e., the instances we want to receive novel insights for by the help of algorithms. The label space $\mathcal{Y}$ comprises the possible outcomes of the model to be learned. In order to avoid trivial or bad results, machine learning approaches are accompanied with a performance measure and an error functional that directs the learning procedure by rewarding good outcomes and punishing bad ones.

This work focuses on the inductive task of learning a regression function for unseen instances. However, aspects of transductive algorithms and transfer learning will play a

role in Chapters 4 and 5 as well. These and other central notions of machine learning will be explained in the two subsequent sections. In the case of regression, typically the distance between true label and prediction value will be used to assess the quality of a model function [Schölkopf and Smola, 2002]. How an appropriate machine learning modelling can be performed via risk minimisation in theory and practice will be shown in Section 2.3 on learning theory. In the present work, we obtain a predictor function via kernel methods using multiple data representations (or views) on data. In Sections 2.4 and 2.5 we show how convex optimisation in combination with a special candidate space for predictor functions induces the beneficial properties of kernel methods. These kernel methods will be applied to solve the considered task of ligand affinity prediction. At the end of this chapter in Sections 2.6 and 2.7 we will present standard regression algorithms and dimensionality reduction methods. These constitute the basis for the multi-view techniques in the three main chapters below.

2.1 Notation

We initiate the chapter on preliminaries with notational conventions that can be used as a reference in the remainder of the thesis. Some of the mathematical concepts mentioned here might not be completely clear at this point, but are either understood intuitively or will be explained precisely in the sections and chapters below.

In the following, we will denote $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{H}$ *instance space*, *label space*, and *hypothesis* or *candidate space*. Single objects will be denoted with $x \in \mathcal{X}$ or $z \in \mathcal{X}$, depending on whether they are instances with or without known label $y \in \mathcal{Y}$. If $\mathcal{X} \subseteq \mathbb{R}^d$ and $x_1, \ldots, x_n \in \mathcal{X}$, by X we address a finite subset of $\mathcal{X}$ in form of a matrix

$$X = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{11} & \ldots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \ldots & x_{nd} \end{pmatrix}. \tag{2.2}$$

The vector $Y = (y_1, \ldots, y_n)^T$ contains the corresponding labels $y_1, \ldots, y_n$. For regression problems the label space are the real numbers $\mathcal{Y} = \mathbb{R}$. More precisely, in the affinity prediction scenario the non-negative real numbers $\mathbb{R}^+$ are used as label space. With respect to machine learning, the concept of a view on (or a representation of) data is the central theme of the present thesis. One particular view will be indexed with v. We will consider M different views. The feature map $\Phi : \mathcal{X} \to \mathcal{H}$ symbolises the representation of an abstract object $x \in \mathcal{X}$ in an appropriate feature space. Hence, for a given view, Φ_v is the v-th feature map and $\Phi_v(\mathcal{X})$ is the v-th view applied on data instances. A predictor function $f : \mathcal{X} \to \mathcal{Y}$ assigns a label $y \in \mathcal{Y}$ to an object $x \in \mathcal{X}$ that holds some interesting information about the object. Again, for a given view v the function $f_v : \Phi_v(\mathcal{X}) \to \mathcal{Y}$ is the v-th (single-)view predictor function. We point out that f_v can either be the result of the v-th single-view regression method (*single-view predictor*) or one of the simultaneous outcomes $f_1, \ldots, f_M$ of a multi-view method (*view predictor*). We will present multi-view optimisation problems below, where the minimisation or maximisation should be performed with respect to multiple different views simultaneously. For the sake of simplicity, we abbreviate

$$\min_{f_1 \in \mathcal{H}_1, \ldots, f_M \in \mathcal{H}_M} \quad \text{with} \quad \min_{f_v \in \mathcal{H}_v}$$

and, analogously,

$$\max_{\alpha_1,\ldots,\alpha_M \in \mathbb{R}^n} \quad \text{with} \quad \max_{\alpha_v \in \mathbb{R}^n}$$

in order to refer to the optimisation with respect to all $v = 1, \ldots, M$. Also for the sake of simplicity, we abbreviate $v \in \{1, \ldots, M\}$ and $u, v \in \{1, \ldots, M\}$ with $v \in [\![M]\!]$ and $(u, v) \in [\![M]\!]^2$, respectively. Frequently, variables below carry double or even triple indices of the kind $\alpha_{v,i}$ or $\gamma_{u,v,j}$, where typically $u, v \in \{1, \ldots, M\}$ are view indices, $i \in \{1, \ldots, n\}$ is the index over labelled examples, and $j \in \{1, \ldots, m\}$ the index over unlabelled instances. We will abbreviate this double or triple indices with α_{vi} or γ_{uvj}.

A kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a generalised similarity measure for objects $x \in \mathcal{X}$ [Vert et al., 2004] and can be related to a view via k_v, analogous to a predictor functions f_v or a feature map Φ_v. The Gram matrix K_v of a kernel function k_v consists of the kernel values

$$K_v = \Big(k_v(x_i, x_j)\Big)_{i,j \in 1,\ldots,n}$$

for $x_1, \ldots, x_n \in \mathcal{X}$, or

$$K_v = \Big(k_v(x_i, x_j)\Big)_{i,j \in 1,\ldots,n+m}$$

for $x_1, \ldots, x_{n+m} \in \mathcal{X}$. It will be stated in the respective sections which definition of K_v will be applied. In the latter case, we use a decomposition of $K_v \in \mathbb{R}^{(n+m)\times(n+m)}$ into an upper $L_v = (K_v)_{i,j=1}^{n,n+m}$ and a lower $U_v = (K_v)_{i,j=n+1,1}^{n+m}$ submatrix.

2.2 The Concept of Learning and Tasks

In this section we introduce very general terms and categorisation attempts of machine learning that are relevant for the present work. According to Mitchell [1997], (machine) learning can be defined as follows.

Definition 2.1 (Learning)**.** [Mitchell, 1997] A computer program is said to *learn* from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

An attempt to align this definition with the concept behind Equation 2.1 is that the machine learning approach is the computer program written with the intention to learn. For this purpose, the machine learning approach requires at least partial knowledge about the instance space $\mathcal{X}$ and label space $\mathcal{Y}$ as well as the relation between both, joined together as experience E. The output of a machine learning approach is the model f which performs task T. The whole process is directed via the *performance measure* P. In a regression approach the instance space $\mathcal{X}$ could be the set of connected labelled graphs that represent molecules, the label space $\mathcal{Y}$ are affinity values from $\mathbb{R}$, and the model from an appropriate function space is evaluated via a performance measure for regression. Let $f : \mathcal{X} \to \mathbb{R}$ be a regression function and $(x_1, y_1), \ldots, (x_n, y_n) \subseteq \mathcal{X} \times \mathbb{R}$ be pairs of instances and corresponding real-valued labels. The *root mean squared error*

(RMSE) of f with respect to the data sample is defined as

$$\mathrm{RMSE}(f) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2}.$$

The described scenario applies to the learning task of affinity prediction that we focus on in the following main chapters. By the properties of E, T, and P we can divide machine learning into different groups of algorithms. However, the decompositions will often be incomplete or overlapping. For example, if we consider the learning task as the distinguishing feature, commonly considered approaches are

- *regression* (real-valued output, e.g., affinity value),
- binary or multi-class *classification* (finitely many classes, e.g., the gender of a person),
- *structured output prediction* (complex output object, e.g., an order of molecular graphs),
- *density estimation* (parameters of a probability distribution, e.g., mean and standard deviation of data points),
- *dimensionality reduction* (lower-dimensional representation of data points, e.g., 2D projection of high-dimensional data),
- *clustering* (decomposition of data in groups, e.g., symptom-based disease classes),
- *association rule mining* (relations between dataset variables, e.g., product suggestion based upon a shopping basket), and
- *reinforcement learning* (sequence of actions, e.g., walk through a labyrinth).

As already mentioned above, the list above is not only an incomplete but also ambiguous. For instance, density estimation could also be assumed a structured output prediction. Another example are *support vector machines* (SVM) algorithms that can be designed for (one- or multi-class) classification, regression, and structured output prediction.

Another categorisation that emphasises the generalisation performance of the learning task discriminates between *inductive* and *transductive learning*. For inductive approaches we learn a universal model f that can be applied to any object of the instance space $\mathcal{X}$, particularly to novel elements of $\mathcal{X}$ unknown at learning time. Whereas, in the transductive learning scenario all training and testing instances as a subset of $\mathcal{X}$ are known a priori [Schölkopf and Smola, 2002]. The availability and quality of training data can also be lifted as grouping criterion with respect to experience. During the learning phase of a model we call data instances $x \in \mathcal{X}$ with known label $y \in \mathcal{Y}$ *labelled data*, whereas instances x without label are named *unlabelled data*. Generally, we distinguish between *supervised* and *unsupervised learning* approaches. For supervised learning labelled data must be available, however, this is not necessary for unsupervised learning. If some training molecules with binding affinity are known, the regression example from above is a typical supervised learning task. The notion of unsupervised learning in this thesis often appears in connection with so-called *transfer learning* [Pan and Yang, 2010].

Transfer learning can best be explained in terms of task and experience as well. It consists in the knowledge transfer drawn from solving a task T by experience E to a related task T' with experience E'. As it is the case in the considered unsupervised learning scenario of affinity prediction in the present thesis, the experience E' for the related task does not necessarily include the availability of labelled training data. A special case of the supervised learning scenario is denoted with *semi-supervised learning*. In addition to labelled data, here we expect to be aware of unlabelled data instances, for example, molecules without known affinity value. Although semi-supervised learning is actually a part of supervised learning we will treat supervised learning, semi-supervised learning, and unsupervised learning independently and dedicate each scenario one of the following chapters.

2.3 Learning Theory

Being aware of general ideas of machine learning, we now have to work out in more detail how a model with desirable generalisation properties can be derived. According to Shalev-Shwartz and Ben-David [2014], a successful learner needs to be able to generalise from examples to unseen instances and to have prior knowledge on the scenario (*inductive bias*). To this aim, we will treat aspects of both *statistical learning theory* [Vapnik, 1999] and *computational learning theory*. They deal with the general problem of "Given a task T, experience E, and a performance measure P, how can we derive a *good* model?" and address questions of the kind "Having T, E, and P, how difficult is it to derive a *good* model?" With our explanations we will focus on the case of supervised and semi-supervised regression as performed in Chapters 3 and 4. In some respects, the unsupervised regression problem from Chapter 5 is transformed into a supervised learning task as well. The presented choice of loss functions, risks, and complexity measures is directed towards the regression task we intend to solve and the techniques we apply for this purpose [Schölkopf and Smola, 2002].

2.3.1 Empirical Risk Minimisation

To the aim of solving a machine learning task and finding an optimal prediction model, a *loss function* is typically applied to measure the appropriateness of a single model. We will use the name *loss* for both the function itself and the loss function's output.

Definition 2.2 (Loss function)**.** [Schölkopf and Smola, 2002] Let $\mathcal{Y}$ be a label space. The non-negative function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ is called a *loss function* if

$$\ell(y, y) = 0$$

for all $y \in \mathcal{Y}$.

In particular, given a data example $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and a model $f : \mathcal{X} \to \mathcal{Y}$. Then the loss function $\ell(y, f(x))$ will output zero if the observation y agrees with the model prediction $f(x)$. Assume, there is another model $f' : \mathcal{X} \to \mathcal{Y}$. If the two predictions $f(x)$ and $f'(x)$ are equal, the loss $\ell(f(x), f'(x))$ would output zero as well. For a regression task T a good model f will be characterised by a preferably small distance between the prediction value $f(x)$ and the true label $y \in \mathbb{R}$ of instance $x \in \mathcal{X}$. The loss ℓ should

return a small value in case the prediction was good, and a low value in the opposite case. Therefore, the respective loss function for regression will take the distance $|y - f(x)|$ into account. A *convex loss function* is a convex function ℓ with input $|y - f(x)|$, i.e., which is convex in the distance between y and $f(x)$. If necessary, we assume the considered loss functions in the present thesis to have this convexity property. In the following, we present the two *distance-based loss functions* [Steinwart and Christmann, 2008] for regression that will accompany us through the whole thesis.

Definition 2.3 (ε-insensitive and squared loss)**.** [Steinwart and Christmann, 2008] Let $\varepsilon > 0$ be a constant and $y_1, y_2 \in \mathcal{Y}$. The *ε-insensitive loss* is defined as

$$\ell_\varepsilon(y_1, y_2) = \max\{0, |y_1 - y_2| - \varepsilon\}. \tag{2.3}$$

The function

$$\ell_2(y_1, y_2) = |y_1 - y_2|^2 \tag{2.4}$$

is known as *squared loss*.

We postulate ε to be greater than zero in order to distinguish it from the absolute loss ℓ_{abs} below, although $\varepsilon = 0$ would be possible in the definition of ℓ_ε as well. The squared loss function ℓ_2 is also commonly known as *least squares loss*. Because of its relation to the ℓ_2-norm we use the symbol ℓ_2, which should not be confused with the space of sequences ℓ_2. The latter continuously penalises gaps between the two inputs y_1 and y_2, whereas for the ε-insensitive loss intervals between y_1 and y_2 smaller than ε do not cause a loss value greater than zero. The different loss functions influence on the final properties of the respective learning algorithms. *Absolute distance* $\ell_{\text{abs}}(y_1, y_2) = |y_1 - y_2|$ and *squared ε-insensitive loss* $\ell_\varepsilon^2(y_1, y_2) = \max\{|y_1 - y_2| - \varepsilon, 0\}^2$ are other related examples of loss functions for regression.

Having defined a loss function that evaluates predictions for single instances, one would like to assess the overall quality of a prediction model. Assume data examples $(x, y) \in \mathcal{X} \times \mathcal{Y}$ are generated via the joint distribution $\mathcal{D} = \mathbb{P}(x, y)$. One would be interested in the minimisation of the *risk functional* (or *expected risk*) [Vapnik, 1999, Schölkopf and Smola, 2002]

$$R(f) = \mathbb{E}_{\mathcal{D}}(\ell(y, f(x)) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, f(x))\, d\mathbb{P}(x, y), \tag{2.5}$$

which collects and weighs the loss of all possible data tuples. However, as the underlying probability distribution $\mathbb{P}$ is unknown, the expected risk can be estimated via the *empirical risk*

$$R_{emp}(f) = \hat{\mathbb{E}}(\ell(y, f(x)) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i)) \tag{2.6}$$

and training examples $(x_1, y_1), \ldots, (x_n, y_n)$ drawn from distribution $\mathcal{D}$. In order to find an appropriate or optimal predictor function f amongst a multitude of functions one has to fix or restrict the set of potential candidates. We call the function space $\mathcal{H}$ the *hypothesis space* or *candidate space*. The aim is to find the best function of the hypothesis space with respect to the empirical risk, which leads us to the *empirical risk minimisation* inductive principle.

Definition 2.4 (ERM)**.** [Schölkopf and Smola, 2002] Let $\mathcal{H}$ be a space of functions mapping from $\mathcal{X}$ to $\mathcal{Y}$ with norm $\|\cdot\|_{\mathcal{H}}$ and ℓ be a loss function. The optimisation

$$\min_{f\in\mathcal{H}} R_{reg}(f) = \min_{f\in\mathcal{H}} \sum_{i=1}^{n} \ell(y_i, f(x_i)) \tag{2.7}$$

is called *empirical risk minimisation* (ERM).

If the hypothesis space is rich and the data is noisy or does not carry sufficient information, the learning task might still be intractable or not solvable in a satisfactory manner [Vapnik, 1999, Schölkopf and Smola, 2002, Shawe-Taylor and Cristianini, 2004]. In this context, *overfitting* denotes the overly adaption of a predictor function to the observations, disregarding the true functional relationship between inputs and outputs. These effects can for example be suppressed by a further limitation or restriction on the functions of the candidate set, known as *regularisation*. A prominent example of regularisation is the inclusion of the function norm into the considered objective to optimise. The following definition is a specification of Definition 2.4.

Definition 2.5 (RRM)**.** Let $\mathcal{H}$ be a candidate space as introduced above. Let, furthermore, $g : \mathbb{R} \to \mathbb{R}$ be a strictly monotonically increasing function. The functional

$$R_{reg}(f) = g(\|f\|_{\mathcal{H}}) + \sum_{i=1}^{n} \ell(y_i, f(x_i)), \tag{2.8}$$

is called *regularised empirical risk* and its minimisation with respect to f is called *regularised risk minimisation* (RRM).

The regularised risk functional R_{reg} is the starting point for a variety of machine learning algorithms, where loss function ℓ and regularising term $g(\|f\|_{\mathcal{H}})$ vary from approach to approach. All machine learning algorithms considered in the main chapters below follow the *(regularised) ERM principle* or a related approach. The details can be found in Chapters 3, 4, and 5. In Definition 2.5 we omit the factor $\frac{1}{n}$ from the empirical risk R_{emp} because of the flexibility of the function g, for example, if $g(\cdot) = \nu(\cdot)^2$ for a trade-off parameter $\nu > 0$. We will use the term *error* synonymously for risk.

2.3.2 Rademacher Complexity

Although not known precisely, it is possible to control the empirical risk of a predictor function, i.e., its qualification to generalise to arbitrary data instances [Shawe-Taylor and Cristianini, 2004]. One would prefer a function class as candidate space such that for every function the difference between training and true error is small (known as *uniform convergence*).

Definition 2.6 (Empirical Rademacher complexity)**.** [Bartlett and Mendelson, 2002] Let $x_1, \ldots, x_n \in \mathcal{X}$ be a random sample of instances drawn i.i.d. from distribution $\mathcal{D}$ and $\mathcal{H}$ be a function class. With $\sigma = (\sigma_1, \ldots, \sigma_n)$ we denote n independently identically distributed Rademacher random variables. The *empirical Rademacher complexity* of $\mathcal{H}$ is defined as

$$\hat{\mathcal{R}}_n(\mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{f\in\mathcal{H}} \left| \frac{2}{n} \sum_{i=1}^{n} \sigma_i f(x_i) \right| \right].$$

The empirical Rademacher complexity is a measure of a function class to fit random data. This property is also known as *capacity* of a function class. Another measure of function class capacity (or complexity) that is independent of the data's probability distribution is the so-called *Vapnik-Chervonenkis dimension*. A big capacity of a function class implies a big capacity to find patterns in random noise. For this reason, a small empirical Rademacher complexity is desirable for the function class $\mathcal{H}$. In Definition 2.6, the randomness should be represented by the Rademacher random variables σ. The following theorem supports the ERM principle, as the difference between expected risk and empirical risk can be controlled via the empirical Rademacher complexity.

Theorem 2.7. *Let $\delta \in (0,1)$. Assume $\mathcal{H}$ is a class of functions $f : \mathcal{X} \to \mathcal{Y}$ and ℓ a loss function mapping into $[0,1]$ without loss of generality. For every $f \in \mathcal{H}$*

$$\mathbb{E}_{\mathcal{D}}(\ell(y, f(x)) \leq \hat{\mathbb{E}}(\ell(y, f(x)) + \hat{\mathcal{R}}_n(\mathcal{H}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}$$

holds true with probability greater or equal $(1-\delta)$.

The theorem was proven by Shawe-Taylor and Cristianini [2004]. Applying Theorem 2.7, a bound on the empirical Rademacher complexity finally gives us the opportunity to compare different approaches with respect to their theoretical generalisation performance. An example of a bound on the empirical Rademacher complexity depending on the algorithm's parameters and the data sample can be found in Section 4.3.5.

2.3.3 Phases of Learning

As we have seen above, the expected risk can be bounded via Rademacher complexity $\hat{\mathcal{R}}_n$ and empirical risk R_{emp}. In the present section, we address the issue of how the learning process can be directed in practice such that the empirical risk of the learned model f becomes minimal given the algorithm, the candidate space, the data, and the limitations of optimisation. Indeed, the successful accomplishment of a machine learning task T requires the two phases *training* and *testing*. During training the available information or experience E is used to actually learn (or train) a model or predictor function, for example via the ERM principle. We will refer to the available data during the training phase as *training data*, regardless whether the data is labelled or unlabelled, or of a completely other kind (such as similarity values between protein targets like in Chapter 5). An optimisation procedure of the algorithm's parameters is also included in the learning phase. It is necessary to pick the best assignment of parameter values for the task at hand. In the testing phase the learned model has to be evaluated with respect to its prediction performance using test data. In this connection, the performance measure P is not always equal to the applied loss function. Via the loss function the predictor function can be equipped with desirable properties such as the sparsity in the case of ε-insensitive loss. However, with respect to the performance P one is interested in the actual discrepancy between true and prediction value (least squares loss in the case of regression). Usually, the entire available data is divided into training data and test data. In the case of supervised or semi-supervised learning, the known labels are compared with the predictions in order to calculate a performance measure.

In order to choose the best parameter assignment and to assess the quality of the learned model as good as possible, i.e., to give a good estimate of its expected risk, we need to

introduce some randomness in the data. It is principally fixed once drawn from some unknown distribution. This randomness in the learning procedure can be achieved via the so-called *k-fold cross-validation* (CV) technique. Here, the whole data is split into k folds and every fold is used once as test data and the respective union of the remaining $k-1$ folds as training data. The final performance is then the average over the k empirical error values. Assume we are at fold k_0. For every combination of possible parameters another k'-fold CV is executed including only the training data of fold k_0. The parameter combination with the lowest error value in average is then chosen for the learning procedure in fold k_0. In general, the best combination will vary from fold to fold depending on the k_0-th training data. If a parameter is real-valued, one should fix a set of typical parameter values out of $\mathbb{R}$ or $\mathbb{R}^+$ and perform a *grid search* over the chosen representatives. Apart from the standard k-fold CV procedure, also other CV schemes are employed in special learning scenarios. For example, inverse CV uses only one fold for training and the other $k-1$ folds for testing (compare Brefeld et al. [2006] and Section 4.4). In particular, if the number of folds k does not induce the desired fraction of training and test data (e.g., 10 folds and 30% of training data) it is more useful to randomly draw the training data for each fold instead of splitting the data into folds.

2.4 Optimisation Theory

We will apply the RRM principle from above in Chapter 3 and Chapter 4, which means that we minimise a functional including the empirical risk which is typically convex. The related approach for the objective function in Chapter 5 also turns out to be convex. Algorithms for kernel methods are generally often formulated in terms of convex optimisation problems as convex problems have a single global optimum [Lanckriet et al., 2004b]. Therefore, the present section is an extract of the theory of *convex optimisation* and provides the basics and efficient solution tools for this group of well-studied optimisation problems. Our explanations below mainly follow the book of Cristianini and Shawe-Taylor [2000]. Details are also taken from the very comprehensive reference on convex optimisation by Boyd and Vandenberghe [2004]. We begin with the notions of convex and affine functions.

Definition 2.8 (Convex function)**.** [Cristianini and Shawe-Taylor, 2000] A function $f : \mathbb{R}^d \to \mathbb{R}$ is *convex* if for any $x_1, x_2 \in \mathbb{R}^d$

$$f(\tau x_1 + (1-\tau)x_2) \leq \tau f(x_1) + (1-\tau) f(x_2)$$

holds true for all $\tau \in (0,1)$. In case of a strict inequality the function is called *strictly convex*.

The convexity property of a function f of d variables is equivalent with its *Hessian matrix* of second derivatives being positive semi-definite [Boyd and Vandenberghe, 2004].

Definition 2.9 (Affine function)**.** [Cristianini and Shawe-Taylor, 2000] Let $A \in \mathbb{R}^{k \times d}$ be a real-valued matrix and $b \in \mathbb{R}^k$ be a vector. A function $f : \mathbb{R}^d \to \mathbb{R}^k$ with $f(x) = Ax + b$ is said to be *affine*.

We will consider minimisation problems with additional requirements concerning the optimisation variables. In the constrained *primal problem* below, f is the *objective*

function, $g_l(x) \leq 0$, $l = 1, \ldots, n_g$, are called the *inequality constraints*, and $h_{l'}(x) = 0$, $l' = 1, \ldots, n_h$, the *equality constraints*. If x^* is a solution of Equation 2.9 then $f(x^*)$ is called the *optimal value* of the problem.

Definition 2.10 (Convex optimisation problem)**.** [Cristianini and Shawe-Taylor, 2000] If the objective function $f : \mathcal{X} \to \mathbb{R}$, the set $\mathcal{X}$, and the inequality constraint functions g_l are convex and the equality constraint functions $h_{l'}$ are affine, the minimisation

$$\min_{x \in \mathbb{R}^d} f(x) \quad \text{s.t.} \left\{ \begin{array}{c} g_l(x) \leq 0 \\ h_{l'}(x) = 0 \end{array} \right\}_{l \in [\![n_g]\!],\ l' \in [\![n_h]\!]} . \tag{2.9}$$

is a *convex optimisation problem*.

A set $\mathcal{X} \subseteq \mathbb{R}^d$ is said to be convex if $\theta x + (1-\theta)x'$ is an element of $\mathcal{X}$ for all $x, x' \in \mathcal{X}$ and all $\theta \in (0, 1)$. We know that for unconstrained problems with differentiable objective function f, the partial derivative set to zero

$$\frac{\partial f(x^*)}{\partial x} = 0$$

is a necessary condition for x^* being a minimum of f. It turns into a sufficient condition if f is additionally convex [Cristianini and Shawe-Taylor, 2000]. The point x^* is the global minimum if f is strictly convex. If we aim at solving problems with equality and inequality constraints we need to expand a bit further and study *Lagrangian theory* and introduce the concept of *duality*. At first, we define an important auxiliary function.

Definition 2.11 (Lagrangian function)**.** [Cristianini and Shawe-Taylor, 2000] Suppose we have an optimisation problem with objective function f, inequality constraint functions $g_l, l = 1, \ldots, n_g$, and equality constraint functions $h_{l'}, l' = 1, \ldots, n_h$. We denote $L : \mathbb{R}^d \times \mathbb{R}^{n_g} \times \mathbb{R}^{n_h} \to \mathbb{R}$ with

$$L = L(x, \alpha, \beta) = f(x) + \sum_{l=1}^{n_g} \alpha_l g_l(x) + \sum_{l'=1}^{n_h} \beta_{l'} h_{l'}(x), \tag{2.10}$$

the *Lagrangian* of the optimisation problem. The real numbers α_l and $\beta_{l'}$ are the *Lagrangian multipliers*.

In this context, the pointwise infimum $\theta : \mathbb{R}^{n_g} \times \mathbb{R}^{n_h} \to \mathbb{R}$ given as

$$\theta(\alpha, \beta) = \inf_{x \in \mathbb{R}^d} L(x, \alpha, \beta)$$

is known as (*Lagrange*) *dual function*. For the sake of simplicity, we omit the variables of the Lagrangian function $L = L(x, \alpha, \beta)$ in calculations below.

The optimal value $f(x^*)$ of Equation 2.9 is bounded from below by the dual function $\theta(\alpha, \beta)$ if $\alpha_l \geq 0$ for all $l = 1, \ldots, n_g$ [Boyd and Vandenberghe, 2004]. The following definition applying this lower bound will lead us to the central solution strategy of *dualisation* for many relevant optimisation problems.

Definition 2.12 (Dual problem)**.** [Cristianini and Shawe-Taylor, 2000] Assume a (primal) problem as stated in Equation 2.9. The maximisation problem

$$\max_{\alpha,\beta} \theta(\alpha, \beta)$$
$$\text{s.t. } \{\alpha_l \geq 0\}_{l \in [\![n_g]\!]}$$

is called its (*Lagrangian*) *dual problem*. The difference between the optimal value of the primal problem and the optimal value of the dual problem $f(x^*) - \theta(\alpha^*, \beta^*)$ is called the *duality gap*.

By the lower bound property of the dual function, the duality gap is always non-negative, which is known as *weak duality*. If the duality gap vanishes we have the case of *strong duality*. We will exploit later that for convex optimisation problems the strong duality theorem holds true. The theorem can be found in Cristianini and Shawe-Taylor [2000].

Theorem 2.13 (Strong duality theorem)**.** *Suppose the equality and inequality constraint functions of a convex optimisation problem are all affine. Then the duality gap is zero, i.e.,* $f(x^*) = \theta(\alpha^*, \beta^*)$.

Various settings that imply strong duality as well as the Karush-Kuhn-Tucker conditions introduced below were discussed by Boyd and Vandenberghe [2004]. Consequently, optimisation problems that fulfill the preconditions from above can equivalently be solved via their dual formulation. It turns out that this fact will be applicable for the majority of machine learning algorithms considered below (compare, in particular, Chapter 3 and 4). The solution strategy for dual problems is a consequence of the theorem of *Kuhn* and *Tucker* [Cristianini and Shawe-Taylor, 2000, Boyd and Vandenberghe, 2004].

Theorem 2.14. *We consider a convex problem with affine equality and inequality constraint functions as in Equation 2.9. Necessary and sufficient conditions for a point* $x^* \in \mathbb{R}^d$ *being optimal for the primal problem is the existence of a pair* $(\alpha^*, \beta^*) \in \mathbb{R}^{n_g} \times \mathbb{R}^{n_h}$ *such that*

$$\frac{\partial L(x^*, \alpha^*, \beta^*)}{\partial x} = 0, \tag{2.11}$$
$$\frac{\partial L(x^*, \alpha^*, \beta^*)}{\partial \beta} = 0, \tag{2.12}$$
$$\alpha_l^* g_l(x^*) = 0, \tag{2.13}$$
$$g_l(x^*) \leq 0, \tag{2.14}$$
$$\textit{and} \qquad \alpha_l \geq 0, \tag{2.15}$$

for $l = 1, \ldots, n_g$.

In the case of no inequality constraints, i.e., $a_l = 0$ for all $l = 1, \ldots, n_g$, the theorem above equals the *Lagrange* theorem and reduces to Equations 2.11 and 2.12. It will be applied, e.g., in Section 2.6.1. The conditions in Equations 2.11 to 2.15 are denoted as *Karush-Kuhn-Tucker conditions* (KKT conditions). Equation 2.13 is also known as *KKT complementary condition*. Its special role will be explained in Section 2.6.2.

Particular optimisation problems are difficult to solve in their initial (primal) formulation. A constructive solution approach is the following. At first, the Lagrangian L is

build according to Equation 2.10. Subsequently, from the derivatives of L with respect to the problem variables and to the dual variables corresponding to equality constraints (compare Equations 2.11 and 2.12) one derives a dual formulation, which can be solved efficiently. The additionally added equality and inequality constraints from the application of KKT conditions are necessary for the correct solution of the optimisation problem and the ranges of the solution parameters (compare Section 2.6.2). We will apply the techniques from convex optimisation in order to find appropriate predictor functions. In order to distinguish predictor functions from objectives, we will denote the objective function in the following chapters with $\mathcal{Q}$, the predictor function with f, and the complexity class symbol with $\mathcal{O}$.

2.5 Kernel Methods

In Section 2.3 on learning theory we have seen that a specification of the candidate space for the predictor function gives the opportunity to have an influence on the generalisation performance of a learning algorithm. To benefit from this result, in the present section we describe a class of function spaces called *reproducing kernel Hilbert spaces*. Each of these spaces is canonically related to a *kernel function* in a way that will be explained in detail below. With *kernel methods* we denote learning algorithms that apply reproducing kernel Hilbert spaces as candidate spaces.

In principle, kernel methods always comprise two steps [Shawe-Taylor and Cristianini, 2004].

1. A mapping of the considered objects is performed into an appropriate linear feature space, and
2. the actual learning algorithm is a search for linear patterns in that feature space.

Because of the *kernel trick*, which will be introduced below, the explicit calculation of the feature representation becomes unnecessary. For this reason, the first step is often performed only indirectly by a reformulation of the algorithm's objectives in terms of kernel values. As the linear methods in feature space are often well-known approaches, such as linear regression, the *kernelised* algorithms can be solved efficiently. As we will see, the combination of linear methods with the benefits of kernel functions make kernel methods powerful machine learning tools.

In the empirical sections below we will mostly utilise the linear kernel, which is the inner product of instances in form of d-dimensional vectors. We present all algorithms in the kernelised version to facilitate the applications of general kernel functions. The linear kernel case is then a special case of the kernelised formulation.

Initially, the central concept of a *Hilbert space* from functional analysis is followed by the definition of the kernel function.

Definition 2.15 (Hilbert space)**.** [Werner, 1995] A mapping $\langle \cdot, \cdot \rangle : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called *inner product*, if it is linear in both arguments, symmetric and $\langle x, x \rangle \geq 0$ for all $x \in \mathcal{X}$ and $\langle x, x \rangle = 0$ if and only if $x = 0$. A *Hilbert space* $\mathcal{H}$ is a complete vector space with norm $\| \cdot \|_{\mathcal{H}}$ such that there is an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ for which $\sqrt{\langle x, x \rangle_{\mathcal{H}}} = \|x\|_{\mathcal{H}}$ is valid.

Definition 2.16 (Kernel function)**.** [Steinwart and Christmann, 2008] Let $\mathcal{X}$ be a set of instances. The function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a *kernel*, if there is a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathcal{X} \to \mathcal{H}$ such that

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}} \tag{2.16}$$

for all $x, x' \in \mathcal{H}$. The function Φ is called *feature map* and the space $\mathcal{H}$ in this context is called *feature space*. The matrix

$$K = \big(k(x_i, x_j)\big)_{i,j=1}^{n} \tag{2.17}$$

is the *Gram matrix* of kernel k with respect to instances $x_1, \ldots, x_n \in \mathcal{X}$.

Actually, for a given kernel k there are infinitely many isometric isomorphic feature maps Φ and Hilbert spaces $\mathcal{H}$ such that Equation 2.16 holds true [Minh et al., 2006]. In the case of $\mathcal{H} = \mathbb{R}^d$, the name feature space is very intuitive as instances $x \in \mathcal{X}$ are mapped to d feature values $\Phi(x)$. This leads us to the definition of a *view* as a central concept of this thesis. It formalises the fact that a view on data is basically a particular feature map of data instances.

Definition 2.17 (View)**.** Let $\mathcal{X}$ be an arbitrary instance space and $\mathcal{H}$ a feature space. The representation of an instance space by a feature map $\Phi(\mathcal{X})$

$$\Phi : \mathcal{X} \to \mathcal{H}$$

is called a *view* on data. We will denote a set of M feature representations

$$\Phi_1 : \mathcal{X} \to \mathcal{H}_1, \ldots, \Phi_M : \mathcal{X} \to \mathcal{H}_M \tag{2.18}$$

as *multiple views* of the instance space $\mathcal{X}$. We refer to the respective view by its index $v \in \{1, \ldots, M\}$.

In addition to the feature space representation, the property of *positive semi-definiteness* will play an important role for kernel functions.

Definition 2.18 (Positive semi-definiteness)**.** [Steinwart and Christmann, 2008] Let $\mathcal{X}$ be an arbitrary instance space. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called *positive semi-definite* if and only if

$$\sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0 \tag{2.19}$$

for all $n \in \mathbb{N}$, $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$, and $x_1, \ldots, x_n \in \mathcal{X}$.

In the strict positive case the function is said to be *positive definite*. It turns out that the characteristic of positive semi-definiteness and the property of a function to be a kernel function are actually equivalent as a consequence of the following theorem [Steinwart and Christmann, 2008].

Theorem 2.19. *A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel according to Definition 2.16 if and only if it is symmetric and positive semi-definite.*

TABLE 2.1: Examples of kernel functions

Name	**Kernel Function** $(x, x' \in \mathbb{R}^d)$
Linear kernel	$k(x, x') = \langle x, x' \rangle$
Tanimoto kernel	$k(x, x') = \frac{\langle x,x' \rangle}{\langle x,x \rangle + \langle x',x' \rangle - \langle x,x' \rangle}$
Polynomial kernel	$k(x, x') = (\langle x, x' \rangle + c)^d$, $c \geq 0$
Gaussian kernel	$k(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$, $\sigma > 0$

As an interpretation of their definition, kernel functions can be regarded generalised similarity measures [Lanckriet et al., 2004c, Vert et al., 2004] between two objects x and x' from an instance space $\mathcal{X}$ of interest. In contrast to mathematical measures, a kernel function is in general not normalised. Regardless, we use the expression *similarity measure* below in order to exemplify the concept of kernel functions. A number of established kernel functions for vectors from $\mathbb{R}^d$ can be found in Table 2.1, where $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$ are the Euclidean norm and scalar product in $\mathbb{R}^d$. Kernels for graphs objects play an important role in chemoinformatics. An introduction to graph kernels and related work can be found in Section 1.3.3 above and the introduction of Chapter 3 below. The kernel property is preserved under summation of two kernel functions and multiplication of a kernel with a positive constant. Furthermore, the tensor product of two kernel functions is a kernel function again [Steinwart and Christmann, 2008]. Analogous closure properties for Gram matrices are valid. From the definition it is obvious that kernel functions and Gram matrices are strongly related to the concept of covariance functions of random variables and covariance matrices (compare also Section 2.7). For more details on the stochastic interpretation of kernel functions and their origin in the context of integral operators consult the literature on kernel theory [Aronszajn, 1950, Schölkopf and Smola, 2002, Shawe-Taylor and Cristianini, 2004, Minh et al., 2006]. The property of a function to be a kernel comes along with the positive semi-definiteness of the corresponding kernel matrices. A symmetric matrix $K \in \mathbb{R}^{n \times n}$ is said to be positive semi-definite if and only if for all $\alpha \in \mathbb{R}^n$

$$\alpha^T K \alpha \geq 0 \tag{2.20}$$

holds true. Hence, a function k is a kernel if all its Gram matrices are positive semi-definite. There are optimisation problems below where the Hessian matrix is equal to the Gram matrix of a kernel function. Consequently, this Hessian matrix is positive semi-definite. The corresponding optimisation has a convex objective function according to Definition 2.8 above. For the resulting convex optimisation problem we can apply the solution techniques from Section 2.4.

As mentioned already above, every kernel function induces a function space. These kind of functions spaces are chosen as candidate space in kernel methods.

Definition 2.20 (Reproducing kernel Hilbert space)**.** [Steinwart and Christmann, 2008] A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called the *reproducing kernel* of the *reproducing kernel Hilbert space* (RKHS) $\mathcal{H}_k$ if and only if

1. $k(x, \cdot) \in \mathcal{H}_k$ for all $x \in \mathcal{X}$, and
2. $f(x) = \langle k(x, \cdot), f \rangle_{\mathcal{H}_k}$ for all $x \in \mathcal{X}$ and all $f \in \mathcal{H}_k$.

The second property is also known as *reproducing property*. Finally, it can be shown that the property of a function to be a *kernel* and to be a *reproducing kernel* are indeed equivalent [Steinwart and Christmann, 2008]. The RKHS $\mathcal{H}_k$ is a feature space of kernel k, which can be seen via the *canonical feature map*

$$\Phi_k(x) = k(x, \cdot) \quad , \quad x \in \mathcal{X},$$

together with the reproducing property, as we may conclude

$$k(x, x') = \langle k(x, \cdot), k(x', \cdot) \rangle_{\mathcal{H}_k} = \langle \Phi_k(x), \Phi_k(x') \rangle_{\mathcal{H}_k}$$

for all $x, x' \in \mathcal{X}$. In contrast to the feature maps and feature spaces, every reproducing kernel has a uniquely defined RKHS and vice versa [Steinwart and Christmann, 2008]. The RKHS $\mathcal{H}_k$ of the reproducing kernel k is one of its corresponding infinitely many feature spaces $\mathcal{H}$ according to the canonical feature map. For this reason, from now on we will omit the index k in the RKHS as long as the corresponding kernel is obvious. There is also a subset of reproducing kernels called *Mercer kernels* which will not be discussed here in more detail.

The already mentioned *kernel trick* describes the fact that the calculation of a kernel value can be substituted by an inner product in an appropriate linear feature space and vice versa. For this reason, the kernel trick enables the application of principally arbitrary linear algorithms in feature space. Moreover, it is possible to avoid the calculation of the potentially non-linear and infinite-dimensional feature map Φ if only the necessary kernel values are known or can be calculated. Hence, alternative formulations of algorithms can be generated by just exchanging the kernel function by an inner product or another kernel function [Schölkopf and Smola, 2002].

As kernel functions correspond to feature representations in a canonical way, the multi-view scenario from Definition 2.17 is equivalent with having a number of kernel functions

$$k_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \ldots, k_M : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

available for instances from $\mathcal{X}$. Again, we realise that handling multi-view problems with kernel methods does not require the explicit knowledge of the respective feature representation if one is able to calculate the kernel function.

The subsequent *representer theorem* supplies us with a convenient solution tool for a class of optimisation problems in kernel methods [Schölkopf et al., 2001, Steinwart and Christmann, 2008]. It guarantees a representation of solution functions as linear combinations of the RKHS's reproducing kernel. Consequently, the representer theorem is the basis for the elegant solution techniques in the context of support vector machines and related algorithms we will consider below.

Theorem 2.21 (Representer theorem)**.** *[Schölkopf et al., 2001] We consider an instance space $\mathcal{X}$ and examples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function and $\mathcal{H}$ be the RKHS of kernel k with norm $\| \cdot \|_{\mathcal{H}}$ and inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Assume we intend to solve*

$$\min_{f \in \mathcal{H}} c\left(y_1, f(x_1), \ldots, y_n, f(x_n)\right) + g\left(\|f\|_{\mathcal{H}}\right) \tag{2.21}$$

for an arbitrary cost function $c : (\mathcal{Y} \times \mathcal{X})^n \to \mathbb{R}$ and strictly monotonically increasing regularising function $g : \mathbb{R}^+ \to \mathbb{R}$. Then a solution f of Equation 2.21 has got a

representation in form of

$$f(\cdot) = \sum_{i=1}^{n} \pi_i k(x_i, \cdot) \tag{2.22}$$

for appropriate $\pi_1, \ldots, \pi_n \in \mathbb{R}$.

Schölkopf et al. [2001] used the term *cost function* in the sense of a loss function generalisation in order to express the loss sustained for the labelled examples $(x_1, y_1), \ldots, (x_n, y_n)$. For the relation between cost and loss refer also to Steinwart and Christmann [2008]. Notice that the minimisation in Equation 2.21 is just an RRM according to Equation 2.8. The subsequent proof is a version of the proof by Schölkopf et al. [2001].

Proof. Let $f \in \mathcal{H}$ be the minimising function of the optimisation in Equation 2.21. We consider the canonical feature map of kernel k with $\Phi(x) = k(x, \cdot)$ for $x \in \mathcal{X}$ and the decomposition of $\mathcal{H}$ into

$$S = \text{span}\,\{\Phi(x_i) : x_1, \ldots, x_n \text{ are the training instances}\}$$

and its orthogonal complement $S^{\perp}$. We may write f always as $f = f_0 + f_1$ such that $f_0 \perp f_1$ and

$$f_0(\cdot) = \sum_{i=1}^{n} \pi_i \Phi(x_i)(\cdot) \in S$$

for $\pi_1, \ldots, \pi_n \in \mathbb{R}$ and $f_1 \in S^{\perp}$. Because of the reproducing and orthogonality property one concludes for the function values $f_1(x_i)$ of the training instances x_i, $i = 1, \ldots, n$, that

$$0 = \langle f_0, f_1 \rangle_{\mathcal{H}} = \langle \sum_{i=1}^{n} \pi_i k(x_i, \cdot), f_1 \rangle_{\mathcal{H}} = \sum_{i=1}^{n} \pi_i \langle k(x_i, \cdot), f_1 \rangle_{\mathcal{H}} = \sum_{i=1}^{n} \pi_i f_1(x_i) \tag{2.23}$$

holds true for every choice of coefficients $\pi_1, \ldots, \pi_n \in \mathbb{R}$. For this reason, $f_1(x_i) = 0$ holds true for $x_1, \ldots, x_n \in \mathcal{X}$ and, hence, the cost function c in Equation 2.21 is unaffected by f_1. Furthermore, as f can be decomposed into the two orthogonal functions f_0 and f_1, the norm term in the objective function can be written as

$$g(\|f\|_{\mathcal{H}}^2) = g(\|f_0\|_{\mathcal{H}}^2 + \|f_1\|_{\mathcal{H}}^2).$$

Again, a non-zero function f_1 would only increase the norm in Equation 2.21, for which reason we conclude the desired representation of f as an element of S. □

We will present all algorithms in kernelised form, i.e., we use RKHSs of appropriate kernel functions as candidate spaces. Therefore, we will omit the additional word *kernel* in the algorithm's names. For examples, we use *support vector regression* instead of *kernel support vector regression*. Based on the representer theorem, we will formulate kernelised objectives and obtain their solutions with respect to the view predictor functions in terms of the kernel linear combination's coefficients π from Equation 2.22. For a Gram matrix $K \in \mathbb{R}^{n \times n}$ of kernel k

$$K\pi = (f(x_1), \ldots, f(x_n))^T \tag{2.24}$$

holds true for instances $x_1, \ldots, x_n \in \mathcal{X}$.

The single-view regression methods introduced in the following section apply Theorem 2.21 for the derivation of their solution. The definition and solution of kernelised RLSR and kernelised SVR build the basis for the multi-view methods in the subsequent chapters. In order to solve multi-view optimisation problems in a kernelised scenario, we aim at a representation of the view predictors as a kernel linear combination as well. In the following chapters, we will consider different multi-view optimisation settings with respect to the availability of labelled and unlabelled data and, hence, with respect to the optimisation objectives. We will prove multi-view versions of the representer theorem in Chapters 3 and 4 and argue for the application of Theorem 2.21 in Chapter 5.

2.6 Single-View Regression

Regression denotes the practical task of finding a real-valued function of one or more input variables. To approximate binding affinities with a feature vector that represents a molecular compound as input is one out of numerous applications for regression techniques. Many single- and multi-view regression algorithms arise from the ERM principle proposed in Section 2.3.1 or related approaches. However, the precise methods differ in the applied regularisation and loss function. In this section we introduce two well-known and effective single-view regression techniques. They form the basis for regression methods that apply multiple views on data which are in the focus of the subsequent main chapters.

2.6.1 Regularised Least Squares Regression

Least squares regression approaches are prominent regression techniques. The first single-view method we present utilises the ℓ_2-norm as loss function and the Hilbert space norm of the predictor function as a regularisation term.

In the subsequent optimisation problem formulations, ν will be used as the trade-off parameter between regularisation term and various loss terms. As the proposed methods originate from different research works, it is not used consistently. In Chapter 3 it is the parameter associated with the labelled error, whereas in Chapter 4 we will use it as regularisation parameter associated with the norm term. It is necessary to adapt formulas or parameter values for the comparison of practical experiments if the parameters in the approaches are combined with different terms in the RRM formulation. For arithmetical reasons, a factor of $1/2$ is included occasionally as parameter joined with the ℓ_2-norm. Furthermore, it is common to define the empirical risk with a factor of $1/n$ in order to reduce the influence of the actual number of examples n. We avoided this factor and compensated it by an appropriate range of selectable parameter values.

The definitions and lemmas in the following Sections 2.6.1 and 2.6.2 are modified and adapted from the introduction of *kernel ridge regression* and *ε-insensitive regression* in [Cristianini and Shawe-Taylor, 2000] and [Shawe-Taylor and Cristianini, 2004].

Definition 2.22 (Regularised least squares regression)**.** Let $\mathcal{H}$ be an RKHS with kernel function k. Assume we have training examples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ and a

regularisation parameter $\nu > 0$. The optimisation problem

$$\min_{f \in \mathcal{H}} \nu \|f\|_{\mathcal{H}}^2 + \sum_{i=1}^{n} |y_i - f(x_i)|^2 \tag{2.25}$$

is called *regularised least squares regression* (RLSR).

The linear case of RLSR is also known as *ridge regression* [Shawe-Taylor and Cristianini, 2004]. Because of the squared loss function this problem is already quadratic in the parameterised form and we can derive a solution analytically.

Lemma 2.23. *Let K be the Gram matrix of kernel k over the training examples and $Y = (y_1, \ldots, y_n)^T$ be the label vector. Let the parameter $\nu > 0$ be chosen, such that the inverse of the matrix $K + \nu \mathbf{I}_n$ exists. Then the solution of kernel regularised least squares regression is*

$$\pi = (K + \nu \boldsymbol{I}_n)^{-1} Y, \tag{2.26}$$

where the predictor function equals $f(x) = \sum_{i=1}^{n} \pi_i k(x_i, x)$.

An appropriate choice of the parameter $\nu > 0$ furthermore ensures the smallest eigenvalue of $K + \nu \mathbf{I}_n$ to be large enough and, hence, a good condition of this matrix.

Proof. From the representer theorem we directly obtain a parameterised version of the solution function according to Equation 2.24. The problem can then be rewritten as

$$\min_{\pi \in \mathbb{R}^n} \nu \pi^T K \pi + \|Y - K\pi\|^2.$$

The derivative of the objective $\mathcal{Q}(\pi)$ with respect to π equals

$$\frac{\partial \mathcal{Q}}{\partial \pi} = 2\nu K\pi - 2K(Y - K\pi)$$

and should be zero for optimality. Hence, we conclude $\pi = (K + \nu \mathbf{I}_n)^{-1} Y$, which finishes the proof. □

2.6.2 Support Vector Regression

The second basic regression approach utilises the ε-insensitive loss function ℓ_ε from Equation 2.3 in its primal formulation of the ERM principle. The desired predictor function f should again be an element of an RKHS $\mathcal{H}$ with kernel k.

Definition 2.24 (Support vector regression)**.** Let $\mathcal{H}$ be an RKHS and let $\varepsilon > 0$ and $\nu > 0$ be model parameters. Given training examples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, the optimisation problem

$$\min_{f \in \mathcal{H}} \nu \|f\|_{\mathcal{H}}^2 + \sum_{i=1}^{n} \max\{0, |y_i - f(x_i)| - \varepsilon\} \tag{2.27}$$

is called *support vector regression* (SVR).

The name *support vector* will be motivated below. Because of the absolute value in the definition of the loss function we cannot apply the same technique we used for RLSR. However, in its dual version SVR according to Equation 2.27 can be solved as a quadratic program in the dual variables. To derive this dual version, we need to reformulate the ε-insensitive loss. An equivalent reformulation can be achieved by the introduction of so-called *slack variables* as done in the proof of the following lemma.

Lemma 2.25. *Assume training instances $x_1, \ldots, x_n \in \mathcal{X}$ and the corresponding vector of labels Y. Let K be the Gram matrix over training instances of kernel k with RKHS $\mathcal{H}$ and hyperparameter $\nu > 0$. The problem*

$$\begin{aligned} \max_{\alpha,\hat{\alpha}\in\mathbb{R}^n} \quad & -\frac{1}{4\nu}(\alpha-\hat{\alpha})^T K(\alpha-\hat{\alpha}) + (\alpha-\hat{\alpha})^T Y - \varepsilon(\alpha+\hat{\alpha})^T \mathbf{1}_n, \\ \text{s.t.} \quad & \mathbf{0}_n \leq \alpha, \hat{\alpha} \leq \mathbf{1}_n \end{aligned} \tag{2.28}$$

is the SVR dual optimisation problem, where for $x \in \mathcal{X}$ the predictor can be written as $f(x) = \sum_{i=1}^n \pi_i k(x_i, x)$, such that $\pi = \frac{1}{2\nu}(\alpha - \hat{\alpha}) \in \mathbb{R}^n$.

In the proof below we apply Theorem 2.21 and the techniques from Lagrangian duality presented in Section 2.4. This procedure will be performed again in Chapter 4 for the solution of multi-view regression approaches.

Proof. For the solution of Equation 2.27 we may directly apply the representation in Equation 2.22 and deduce a representation of the solution function $f(\cdot) = \sum_{i=1}^n \pi_i k(x_i, \cdot)$ with coefficients $\pi_1, \ldots, \pi_n \in \mathbb{R}$, leading to a parameterised variant of SVR

$$\min_{\pi\in\mathbb{R}^n} \nu\pi^T K\pi + \mathbf{1}_n^T \max\{\mathbf{0}_n, |Y - K\pi| - \varepsilon\mathbf{1}_n\},$$

where the max function acts component-wise

$$\max\{\mathbf{0}_n, |Y - K\pi| - \varepsilon\mathbf{1}_n\} = \begin{pmatrix} \max\{0, |y_1 - (K\pi)_1| - \varepsilon\} \\ \vdots \\ \max\{0, |y_n - (K\pi)_n| - \varepsilon\} \end{pmatrix} \in \mathbb{R}^n.$$

We reformulate the primal kernelised problem via the inclusion of slack variables $\xi, \hat{\xi} \in \mathbb{R}^n$ and obtain

$$\begin{aligned} \min_{\pi\in\mathbb{R}^n,\ \xi,\hat{\xi}\geq 0_n} \quad & \nu\pi^T K\pi + (\xi+\hat{\xi})^T \mathbf{1}_n \\ \text{s.t.} \quad & \left\{ \begin{array}{rcl} Y - K\pi & \leq & \varepsilon\mathbf{1}_n + \xi \\ K\pi - Y & \leq & \varepsilon\mathbf{1}_n + \hat{\xi} \\ \xi, \hat{\xi} & \geq & \mathbf{0}_n \end{array} \right\}. \end{aligned} \tag{2.29}$$

By means of the Lagrangian multipliers $\alpha, \hat{\alpha}, \beta, \hat{\beta} \geq 0_n$ we can introduce the constraints into the objective and formulate the Lagrangian L (see Equation 2.10)

$$\begin{aligned} L =& \nu\pi^T K\pi + (\xi+\hat{\xi})^T \mathbf{1}_n \\ & + \alpha^T(Y - K\pi - \varepsilon\mathbf{1}_n - \xi) + \hat{\alpha}^T(K\pi - Y - \varepsilon\mathbf{1}_n - \hat{\xi}) \\ & - \beta^T\xi - \hat{\beta}^T\hat{\xi}. \end{aligned}$$

According to the KKT condition in Equation 2.11 the partial derivatives of L with respect to primal and slack variables

$$\frac{\partial L}{\partial \xi} = \nu \mathbf{1}_n - \beta - \alpha, \ \frac{\partial L}{\partial \hat{\xi}} = \nu \mathbf{1}_n - \hat{\beta} - \hat{\alpha}, \text{ and } \frac{\partial L}{\partial \pi} = 2\nu K\pi - K(\alpha - \hat{\alpha})$$

need to be zero which induce the inequality constraints on the dual variables $\mathbf{0}_n \leq \alpha, \hat{\alpha} \leq \mathbf{1}_n$, the relation between primal and dual variables $\pi = \frac{1}{2\nu}(\alpha - \hat{\alpha})$, and the dual objective $\mathcal{Q}(\alpha, \hat{\alpha})$

$$\mathcal{Q}(\alpha, \hat{\alpha}) = -\tfrac{1}{4\nu}(\alpha - \hat{\alpha})^T K(\alpha - \hat{\alpha}) + (\alpha - \hat{\alpha})^T Y - \varepsilon(\alpha + \hat{\alpha})^T \mathbf{1}_n, \tag{2.30}$$

which finishes the proof. □

It is also possible to use the square of the loss vector in the problem formulation of Equation 2.27, which leads to a very similar optimisation problem [Shawe-Taylor and Cristianini, 2004]. This is referred to as *squared ε-insensitive loss regression*. RLSR is a special case of it for $\varepsilon = 0$.

Interestingly, the KKT *complementary condition* in Equation 2.13 is not used for the derivation of the dual formulation in Equation 2.28. Nevertheless, it determines important characteristics of SVR and related algorithms which we will encounter, e.g., in Chapter 4. In particular, the KKT complementary condition can be consulted to substantiate the expression *support vector*. Actually, it postulates that the product between inequality constraint and corresponding Lagrangian multiplier must be zero at the solution. In the case of the SVR formulation in Equation 2.29 we obtain

$$\alpha_i(y_i - (K\pi)_i - \varepsilon - \xi_i) = \alpha_i g_i(\pi) = 0 \tag{2.31}$$

$$\hat{\alpha}_i((K\pi)_i - y_i - \varepsilon - \hat{\xi}_i) = \hat{\alpha}_i \hat{g}_i(\pi) = 0 \tag{2.32}$$

$$\beta_i \xi_i = (1 - \alpha_i)\xi_i = 0 \tag{2.33}$$

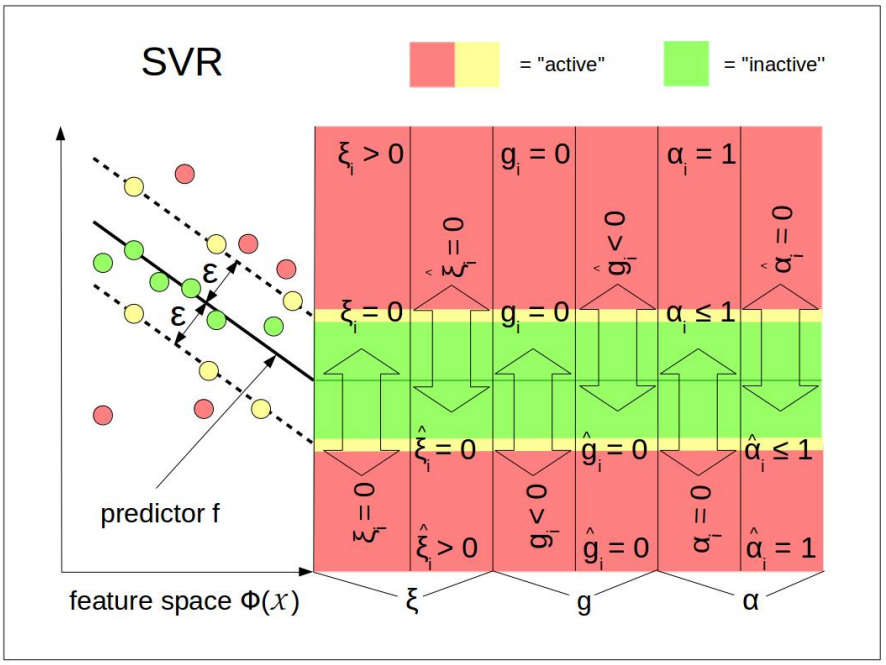
$$\hat{\beta}_i \hat{\xi}_i = (1 - \hat{\alpha}_i)\hat{\xi}_i = 0. \tag{2.34}$$

The primal SVR formulation in Equation 2.29 also implies that ξ_i should be the smallest non-negative value such that $g_i(\pi) \leq 0$ holds true. If $y_i - f(x_i) < \varepsilon$ (prediction and label have a distance strictly smaller than ε) it follows that $g_i(\pi) < 0$ and, therefore, the corresponding multiplier must be zero ($\alpha_i = 0$) in order to satisfy Equation 2.31. In the case of $y_i - f(x_i) = \varepsilon$ the complementary condition in Equation 2.31 is fulfilled directly with $g_i(\pi) = 0$, for which reason it is sufficient that $\xi_i = 0$ and $\alpha_i \leq 1$ (as a consequence of $\partial L / \partial \xi = 0$ in the proof above). If the solution of the predictor function in the original setting was assumed to be of the form

$$f(\cdot) = \sum_{i=1}^{n} \pi_i k(x_i, \cdot) + b$$

for $b \neq 0$ (other than the representation in Equation 2.22), an instance x_i with $\alpha_i < 1$ would enable the calculation of a function bias $b \in \mathbb{R}$ via $y_i - \sum_{i=1}^{n} \pi_i k(x_i, \cdot) + b - \varepsilon = 0$. If y_i is farther than ε apart from the prediction $f(x_i)$ also $g_i(\pi) = 0$ holds true, but with $\xi_i > 0$. Consequently, $\alpha_i = 1$ is valid according to Equation 2.33. Equivalent conclusions hold true for $\hat{\xi}_i$, $\hat{g}_i$, and $\hat{\alpha}_i$. Figure 2.1 gives an overview of the relations between inequality constraints ($g_i / \hat{g}_i$), corresponding multipliers ($\alpha_i / \hat{\alpha}_i$), and slack variables ($\xi_i / \hat{\xi}_i$) in the respective areas around the predictor function hyperplane.

FIGURE 2.1: Active and inactive inequality constraints (g), multipliers (α), and slack variables (ξ)



It shows the values taken by the ξ-, g-, and α-variables within the ε-margin, at the margin border, and apart from the predictor hyperplane. By construction, also $\alpha_i\hat{\alpha}_i = 0$ is valid, for which reason either α_i or $\hat{\alpha}_i$ is equal to zero. An inequality constraint is called *active* if it is valid in form of an equality and *inactive* in case of a strict inequality. Therefore, only training instances x_i with corresponding active inequality constraints, i.e., that lie at the ε-tube around the SVR predictor f or outside of it, have a non-zero multiplier α_i or $\hat{\alpha}_i$, respectively. These instances play an important role for the SVR model.

Definition 2.26 (Support vector)**.** [Schölkopf and Smola, 2002] We consider the representation of the solution function as kernel linear combination $f(\cdot) = \sum_{i=1}^{n} \pi_i k(x_i, \cdot)$ from Equation 2.22. A training instance $x_i \in \mathcal{X}$ is called *support vector* if its corresponding linear coefficient π_i is different from zero.

From $\pi_i = \frac{1}{2\nu}(\alpha_i - \hat{\alpha}_i)$ we see that only instances with corresponding active inequality constraints $g_i(\pi)$ or $\hat{g}_i(\pi)$ are necessary to calculate the prediction values of f. The margin size of ε determines the actual *sparsity* of the predictor, i.e., the number of coefficients π_i unequal to zero (compare also Section 4.3.4). This sparsity property is characteristic for SVR and related algorithms [Chan et al., 2007] and leads to an efficient model calculation and storage.

2.7 Dimensionality Reduction

In this section we will discuss machine learning tools to reduce the feature space dimension. Essential information with respect to objective criteria from typically high-dimensional feature space representations should be kept during the reduction procedure and unnecessary information discarded. Working with lower-dimensional feature spaces

reduces memory requirements and computing time. Moreover, it is a reasonable demand that the mapped objects maintain nearly the whole information compared to the objects in the original feature space. Dimensionality reduction as an unsupervised data-driven downscaling can be regarded a learning task itself or as a tool to solve another learning task as it is the case, for example, in Chapter 5. There are approaches which take multiple views on data into account to calculate an appropriate reduction model, e.g., *canonical correlation analysis* (CCA). However, the following two methods only consider one single feature space.

2.7.1 Johnson-Lindenstrauss Random Projection

Firstly, we present a random projection technique for dimensionality reduction based on the work of Dasgupta and Gupta [2003]. We consider an arbitrary but fixed data matrix $\Phi(X) \in \mathbb{R}^{n \times D}$, such that $\Phi(x_1)^T, \ldots, \Phi(x_n)^T$ are the rows according to Equation 2.2 and $\Phi : \mathcal{X} \to \mathbb{R}^{\mathbb{D}}$ is the feature map for instances from $\mathcal{X}$. The *Johnson-Lindenstrauss* (JL) [Dasgupta and Gupta, 2003] lemma states that for a well-defined projection mapping $f : \mathbb{R}^D \to \mathbb{R}^d$, $d \leq D$, the distance between instances from X remain approximately the same in the image space compared to the distance in the initial feature space. More formally, for two instances $x, x' \in X$

$$(1-\varepsilon)\|\Phi(x) - \Phi(x')\|^2 \leq \|f(\Phi(x)) - f(\Phi(x'))\|^2 \leq (1+\varepsilon)\|\Phi(x) - \Phi(x')\|^2 \qquad (2.35)$$

holds true, where $0 < \varepsilon < 1$ is a small error bound. For more details on the preconditions of the mapping f and the proof of the JL lemma we refer to Dasgupta and Gupta [2003].

As it will be applied in the empirical section of Chapter 5, we present an example for a precise projection f that fulfills the requirements of the JL lemma. We consider the data matrix $\Phi(X) \in \mathbb{R}^{n \times D}$ which we intend to map to a lower dimension d. Indeed, the mapping

$$f(\Phi(x)) = \frac{1}{\sqrt{d}} P^T \Phi(x), \qquad (2.36)$$

such that $P \in \mathbb{R}^{D \times d}$ and $\Phi(x) \in \mathbb{R}^D$ consists of *Bernoulli* random variables with a probability of success of $p = 0.5$ is a valid JL projection [Baraniuk et al., 2008] if $d \in \mathcal{O}((\ln n)\varepsilon^{-2})$ holds true.

2.7.2 Principal Component Analysis

For the introduction of the second unsupervised approach we explicitly consider the feature map $\Phi : \mathcal{X} \to \mathbb{R}^D$, where D is the dimension of the initial feature space for instances. The cumulative variance contained in the feature space components can be used as indicator of the information content or to monitor the information loss in the case of dimensionality reduction. Therefore, the idea of *principal component analysis* (PCA) [Schölkopf et al., 1997] is to learn an orthogonal transformation of the feature space such that the resulting projection of data keeps as much intrinsic variance as possible in decreasing order of the resulting components [Schölkopf and Smola, 2002]. This demand can be formulated as an eigenvector problem which we will introduce briefly in the following.

We assume the data representations to be centered, i.e., the mean of every feature space component is supposed to be zero. Let $\Phi(X) \in \mathbb{R}^{n \times D}$ be the data matrix in the initial feature space corresponding to instances $x_1, \ldots, x_n \in \mathcal{X}$

$$\Phi(X) = \begin{pmatrix} \Phi^T(x_1) \\ \vdots \\ \Phi^T(x_n) \end{pmatrix},$$

where $\Phi(x) \in \mathbb{R}^D$ for all $x \in \mathcal{X}$. Now we aim at a projection matrix $P \in \mathbb{R}^{D \times d}$ such that the projected data

$$\Phi(X)P \in \mathbb{R}^{n \times d}$$

exhibits the desired properties of maximal variance within a smaller number $d \leq D$ of projection image components. It turns out that the columns p of P are actually the eigenvectors of the empirical covariance matrix

$$C = \frac{1}{n}\Phi^T(X)\Phi(X) \in \mathbb{R}^{D \times D}$$

according to the eigenvector-eigenvalue equation $Cp = \lambda p$, where λ is an eigenvalue. This eigenvector problem can be solved as an ERM problem according to Equation 2.7. For the precise formulation and more details consult Schölkopf et al. [1997] and Schölkopf and Smola [2002].

With regard to the kernelised PCA formulation, let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function with canonical feature map $\Phi(x) = k(x, \cdot)$ as well as $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$ for all $x, x' \in \mathcal{X}$. If p is an eigenvector solution of the PCA optimisation, as a consequence of Theorem 2.21 there are coefficients $\pi_1, \ldots, \pi_n \in \mathbb{R}$ such that p has a representation in form of $p = \Phi^T(X)\pi$. Consequently, we obtain a kernelised formulation via

$$\Phi(X)P = \Phi(X)\Phi^T(X)\Pi = K\Pi, \tag{2.37}$$

where $K = \Phi(X)\Phi^T(X) \in \mathbb{R}^{n \times n}$ is the Gram matrix and $\Pi \in \mathbb{R}^{n \times d}$ a projection of K. If the inverse K^{-1} exists we conclude

$$\begin{aligned} \lambda p &= Cp \\ \lambda(\Phi^T(X)\pi) &= C(\Phi^T(X)\pi) \\ \lambda\Phi(X)(\Phi^T(X)\pi) &= \Phi(X)C(\Phi^T(X)\pi) \\ \lambda\Phi(X)\Phi^T(X)\pi &= \tfrac{1}{n}\Phi(X)\Phi^T(X)\Phi(X)\Phi^T(X)\pi \\ \lambda K\pi &= \tfrac{1}{n}K^2\pi \\ n\lambda\pi &= K\pi. \end{aligned}$$

Hence, the kernelised PCA algorithm is a modified eigenvector problem and its result Π is a projection of the kernel matrix K. Let $(\gamma, \tilde{\pi})$ be a pair of eigenvalue and corresponding eigenvector of K. The scaled eigenvectors $\pi = \tilde{\pi}/\sqrt{\gamma}$ build the columns of Π in Equation 2.37. The scaling of the eigenvectors $\tilde{\pi}$ arises from the requirement of an orthonormal basis in P from above, where $1 = p^T p$ is necessary. The final number of columns $d \leq \min\{n, D\}$ must be chosen depending on the practical purpose and the data itself. The PCA approach for dimensionality reduction can be applied with arbitrary kernel functions as the feature vectors are only required in form of inner products. For more

details on the derivation of PCA and its properties we refer to Schölkopf et al. [1997, 1998], Schölkopf and Smola [2002] and Shawe-Taylor and Cristianini [2004].

An equivalent formulation of the PCA eigenvector-eigenvalue problem will be used in Chapter 5. Let $M \in \mathbb{R}^{d \times d}$ be an arbitrary symmetric matrix. Hence, M has got a so-called *eigenvalue decomposition* [Werner, 1995] in form of

$$M = UDU^T, \tag{2.38}$$

where $U \in \mathbb{R}^{d \times d}$ is a unitary matrix with columns equal to the eigenvectors of M. This decomposition into U and D is denoted with *diagonalisation*. The factor $D \in \mathbb{R}^{d \times d}$ is a diagonal matrix such that the eigenvalues of M are the corresponding diagonal elements. Equation 2.38 is equivalent with $U^T MU = D$. Regarding the reformulation of the PCA optimisation we exploit the fact that the value of

$$\begin{aligned} &\max_{U \in \mathbb{R}^{d \times d'}} \operatorname{tr}(U^T MU) \\ &\text{s.t. } U^T U = \mathbf{I}_{d'} \end{aligned} \tag{2.39}$$

is reached if the columns of U are the eigenvectors corresponding to the d' largest eigenvalues of M [Werner, 1995]. Hence, the maximal value is the sum of the d' largest eigenvalues.

Chapter 3

Multiple Kernel Learning

We introduced ligand affinity prediction as an important problem from chemoinformatics in detail in Section 1.3.4 of the introductory chapter. In order to support the expensive identification of ligand affinities in practice and to plan experiments efficiently, machine learning methods can be used to predict affinity values via computational algorithms in the context of similarity-based virtual screening. SVR utilising a molecular fingerprint is the state-of-the-art method and was already tested successfully [Liu et al., 2006, Sugaya, 2014, Balfer and Bajorath, 2015]. This supervised approach for regression employs labelled instances in vectorial format in order to train an inductive model for future instances. Many publicly available or commercial fingerprint descriptors for small molecules exist. These fingerprints list (or count) diverse physico-chemical properties of the respective molecule, structural properties of their molecular graphs in 2D, or even 3D information [Bender et al., 2009]. The variety of data descriptions here is both a blessing and a curse. On the one hand, there are many different data representations available, which were originally designed towards different purposes. On the other hand, the variety of representations implies the need for a choice of the optimal one for the affinity prediction task. In the first instance, it is not obvious which molecular representation is optimal for a considered prediction task from chemoinformatics [Fröhlich et al., 2005]. A branch of chemoinformatics research considers fingerprint reduction and recombination techniques to design an optimal fingerprint and select the most informative features for prediction [Willett, 2006, Nisius and Bajorath, 2010, Heikamp and Bajorath, 2012].

In contrast to the described approaches, in the present chapter we investigate strategies to deal with the variety of descriptors by including multiple representations simultaneously. To this aim, a group of multi-view learning approaches named *multiple kernel learning* (MKL) trains a linear combination of predictor functions such that each function is related to a particular representation of data using labelled training data (see Section 1.2.3). Although multiple views are not completely new to chemoinformatics (compare Section 1.3.6), the application of supervised MKL approaches in combination with a systematic choice of graph patterns is novel in the field of ligand affinity prediction. These MKL approaches will be the first group of multi-view learning methods which we investigate in this thesis. Beyond our focused task of affinity prediction, the proposed approaches below are generally applicable for learning tasks with

- instances that can be interpreted as graphs,

- multiple data representations with appropriate similarity measure (kernel functions),
- a real-valued label, and
- sufficient labelled examples.

The following real-world applications display examples which illustrate the described learning scenario.

Example 3.1. ***(Drug discovery)*** *The interaction of chemical substances with each other, such as the binding of a small molecule to a protein, needs to be tested practically in a time- and cost-consuming process. However, the efforts made in this research field are justified by the fact that many drugs act as protein ligands. Documented laboratory results meanwhile led to the formation of huge molecule databases with ligands and their respective protein affinity. Various kinds of molecular fingerprint descriptors have been developed (see Section 1.3.4) and can be used to represent small molecules differently. This information can be used to learn binding models of proteins in supervised algorithms.*

Example 3.2. ***(Temperature forecast)*** *The development of climate has now been recorded for decades in great detail and nearly on a worldwide basis. In view of a dramatical increase of the temperature on earth its forecast is no longer only important for the weather of the following days. The temperature at a certain location in the world is recorded together with a variety of characteristic information, such as physical information (air pressure, humidity, cloudiness, wind strength and direction), geographical information (soil conditions, temperature zone, and vegetation), local information (GPS position, height, hillside situation), different wavelength sensors from satellite data, and neighbourhood information. Using multi-view learning the temperature can be forecasted taking various information sources on climate and environment into account.*

Example 3.3. ***(Condition monitoring and predictive maintenance)*** *In assembly and production a number of input parameters describe the process conditions. Additionally, accompanying equipment like microphones or acceleration sensors record the progress of the production process and the quality of the involved tools and products. To the aim of a maximal product quality and resource efficiency the prediction of present and future tool condition parameters is an important application of supervised multi-view algorithms.*

Apart from the algorithmic aspect of this chapter, we additionally address the topic of view generation and analysis for graph data, such as small molecules. Actually, both structural and neighbourhood information are crucial for the capacity of small molecules to be a ligand and for the strength of the binding [Ralaivola et al., 2005, Gaüzère et al., 2014]. For example, the presence of a benzene ring or that of an alcoholic group and their relative positions influence the chemical properties of the compound at hand. None of the existing fingerprints that collect structural information, however, captures both all circular and tree patterns of the molecular graph independent of size and the adjacency and connectivity information of atoms within the graph structure. To the aim of an optimised graph representation for the practical task at hand we propose to investigate and systematically combine graph kernels that incorporate relevant patterns for structural and neighbourhood information. To be more precise, we consider the feature set of the *cyclic pattern kernel* (CPK) [Horváth et al., 2004, Horváth, 2005], the

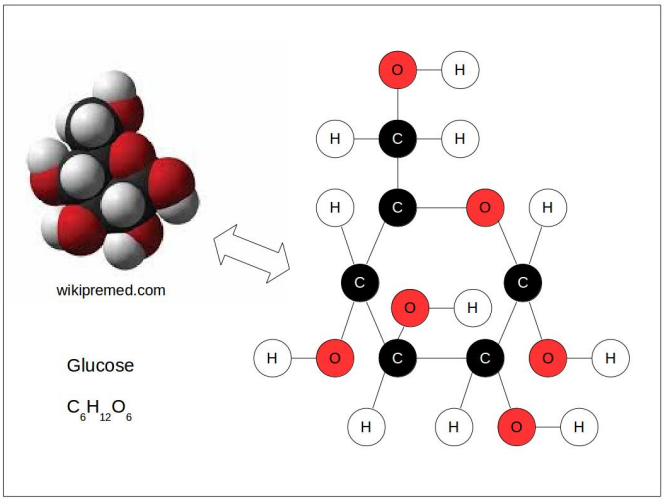
feature set of shortest path (SP) kernels [Borgwardt and Kriegel, 2005], and *Weisfeiler-Lehman* (WL) labels [Shervashidze et al., 2010]. The WL algorithm assigns (new) labels to each vertex in the graph that depend on the surrounding vertices up to a certain distance h. CPK decomposes a graph into the set of contained cycles ($\mathcal{C}$) and remaining tree components ($\mathcal{T}$) formed by edges that do not belong to cycles. Shortest path features ($\mathcal{P}$) collect the shortest paths from one vertex to another. Finally, we also consider the labels of the atoms ($\mathcal{L}$) themselves as features. In order to supply structural patterns with additional neighbourhood information we determine cycles, trees, shortest paths, and labels based on the WL labelled graph of depth h, resulting in $4 \cdot h$ feature sets, each in a binary or counting feature representation [Ralaivola et al., 2005]. However, it is neither clear which of them to keep in the application scenario of affinity prediction, nor obvious how important the components are for the success of prediction. We propose and evaluate a systematic view generation and analysis process to obtain an optimised choice and weighting of the mentioned feature sets applying a *multi-pattern kernel* (MPK) [Ullrich et al., 2016b]. Being a kernel linear combination, the MPK can directly be included in an MKL algorithm. [Fröhlich et al., 2005] We employ two formulations and corresponding solutions of MKL, *learning kernel ridge regression* (LKRR) by Cortes et al. [2009] (which we will denote ℓ_2-MKL) and another ε-insensitive loss MKL variant of Vishwanathan et al. [2010] (ε-MKL). Both algorithms optimise a linear combination of kernel functions corresponding to the provided data representations or views. The linear combination of functions is included in a regularised empirical risk functional with squared loss function (ℓ_2-MKL) or ε-insensitive loss (ε-MKL). The resulting multi-view SVR and multi-view RLSR algorithms and MPKs are applied to find the graph feature combination that achieves the best prediction results. In the case of the linear kernel this is equivalent with utilising a novel fingerprint representation with differently weighted pattern components, such that the weighting highlights the importance of the pattern group for the affinity prediction task.

The present chapter is based on our publication [Ullrich et al., 2016b]. It is structured as follows. At first we deliver all necessary tools from the theory of graph kernels in Section 3.1. After we discussed how an aromatic edge label can be determined automatically from the molecule's structure formula, we consider some properties of set kernels. Subsequently, we introduce three important representatives of graph or set kernels, accompanied with the pattern classes cycles, trees, shortest paths, and WL labels in Sections 3.1.3, 3.1.4, and 3.1.5. The pattern classes will then be used in the following definition of multi-pattern kernels for molecular learning instances in Section 3.2. In the third section, we present solutions for an SVR and an RLSR multiple kernel learning variant. Finally, in the empirical Section 3.4.1 we analyse multiple kernel learning with MPKs for affinity prediction. We show that in comparison with single-view baselines and standard molecular fingerprints we can indeed take profit from the proposed multi-view approach for our considered practical task.

3.1 Graph Kernels

The concept of graph kernels was introduced in Section 1.3.3 above. Similar to molecular fingerprints, there is a variety of graph kernels [Gärtner, 2003]. We consider graph kernels which are based on different feature representations of structural patterns for molecules. The following definition is modified from [Horváth et al., 2004].

FIGURE 3.1: Glucose molecule in 3D representation and as a graph



Definition 3.1 (Graph)**.** Let Σ be a set of labels which can be ordered linearly. The tuple $G = (V, E)$ together with the *labelling function* $\lambda_G : V \cup E \to \Sigma$ is called a *labelled undirected graph* if V is a finite set of *vertices* and $E \subseteq V^2$ are the *edges*. The labelling function λ_G assigns a label to every edge and vertex.

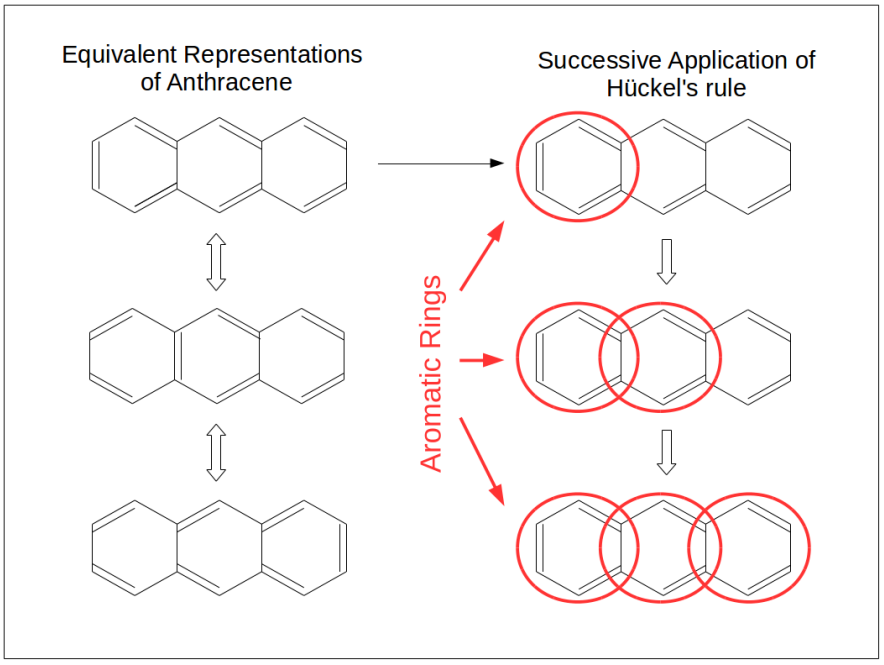
The edges E as a subset of V^2 in undirected graphs are assumed to be sets of the form $\{v_1, v_2\}$ such that $v_1, v_2 \in V$. In the graph scenario, we will denote the instance space $\mathcal{G}$ (instead of $\mathcal{X}$) in order to emphasise that the learning objects are graphs. A function $k : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ is called a *graph kernel* if it is a kernel function and the instance space $\mathcal{G}$ is a set of graphs. The presented kernels below are examples of graph kernels and their calculation is based on the determination of structurally interesting subgraphs, also called *graph patterns*. In the ligand affinity prediction scenario kernels for graphs are of particular interest, because molecules can be considered as undirected labelled graphs (compare also Section 1.3.4). Their atoms and covalent bonds serve as nodes and vertices, respectively, whereas atom and bond types are the labels. This is illustrated for the *glucose* molecule in Figure 3.1 [Ullrich et al., 2016b]. Before we start with the introduction of graph kernels and graph patterns, we illustrate how we tackled the challenge of identifying and labelling the special bond type of an aromatic bond. Furthermore, we briefly consider the concept of set kernels and their relation to pattern feature vectors of graph kernels.

3.1.1 The Aromatic Bond

Potential ligands are small organic molecules that typically carry single and double bonds (labels *1* and *2*). Very rarely also triple bonds occur (label *3*), for example in the class of alkynes [Nelson and Cox, 2001]. There is a fourth type of bonds that plays an important role in our considered molecules and that widely influences their binding properties. Therefore, this bond type should be marked with an extra label *a*. In fact, small (organic) molecules frequently exhibit *aromatic* ring systems. The so-called π-electrons are outer atom shell electrons which are not involved in a bond to another atom. An aromatic ring is a ring of atoms in which the π-electrons of the atoms are delocalised, i.e., the π-electrons are no longer assignable to one fixed atom orbital

but to a group of orbitals. In this case all bonds in the ring are said to be aromatic. As a consequence of this, the ring has a planar structure (in contrast for example to the jagged glucose ring), is more stable, and supplies the corresponding substances with novel characteristics. Typically, the aromatic molecule parts are represented as conjugate (alternating) single and double bonds in structural formulas of molecules. In Figure 3.2 three equivalent representations of the molecule *anthracene* with different arrangements of single and double bonds are shown. Because of these differences and the simplification tendencies in order to omit redundant information it is not always trivial to detect aromatic bonds in molecular graph representations such as the SDF (compare Section 1.3.4). A very helpful tool is *Hückel's rule* that classifies a molecular ring to be aromatic if the sum of its π-electrons is $4n + 2$ for some $n \in \mathbb{N} \cup \{0\}$ [Nelson and Cox, 2001]. For the above mentioned reasons this rule is not always applicable in a straightforward manner. To understand the following explanatory example one should be aware that the vertices of rings in organic molecules are carbon atoms. Furthermore, the double bonds are represented with two lines and have 2 π-electrons, whereas single bonds are represented as single lines and have no π-electron. Anthracene consists of three aromatic 6-rings in a row. Figure 3.2 shows that in each of the representations of anthracene one of the three rings has 6 π-electrons and the respective others only 4 π-electrons. Via Hückel's rule only one ring would be classified correctly as aromatic and it depends on the considered representation which ring.

FIGURE 3.2: Hückel's rule applied to the anthracene molecule



To overcome this problem, we introduce a Hückel's rule heuristic in order to easily detect aromatic rings ad hoc in molecular graph representations with high accuracy. It is based on the fact, that aromatic structures can be detected by a successive application of Hueckel's rule (compare Figure 3.2). We formulated this practical procedure as pseudocode in Algorithm 1 that can be found in Appendix C. Aromatic ring systems usually consist of molecular rings with 5 or 6 atoms (*5- or 6-cycles*) that are aromatic themselves. Therefore, we systematically test each of the found simple cycles for their aromaticity. The Hückel's rule heuristic is accompanied with the following two conventions:

- If a simple cycle is identified to be aromatic, all its edges immediately get the label *a* for *aromatic*.
- If an edge once got label *a*, it keeps this label during all further loops of the algorithm.

The most frequent case are cycles that fulfill Hückel's rule for $n = 1$, i.e., have 6 π-electrons. For 5-cycles with only one and 6-cycles with only two double bonds it is helpful to keep in mind that this already enforces the planarity of the ring and therefore, promotes the aromaticity. Two π-electrons can be contributed by a double or aromatic bond (labels *2* or *a*) or by a free electron pair delivered from a nitrogen, oxygen, or sulfur heteroatom, i.e., a vertex with label *N*, *O*, or *S*. In Figure 3.2 we see an example of how the aromatic character of one ring induces the aromaticity of the other, resulting in equivalent molecule representations and the repeat-until-loop in the algorithm below. The reduced representations of structure formulas with single and double bonds in Figure 3.2 imply a *carbon atom C* at every vertex and a *hydrogen atoms H* at every spare valence electron.

To the best of our knowledge, this algorithm classifies the majority of existing aromatic rings or ring systems correctly. Nevertheless, being a heuristic implementation of Hückel's rule, Algorithm 1 fails for very special structures. One prominent example is the *porphyrin* ring system in *hemoglobin* or *chlorophyll* that has an aromatic 16-ring which is only partially detected to be aromatic by Algorithm 1. Even more, there are aromatic ring systems that do not satisfy Hückel's rule at all, e.g., the polycyclic superphenalene. However in this rare cases, the union of other labels and patterns already describes the respective molecules well. And also in case of false positive aromatic rings, e.g., as in the case of *phenalene*, the structure of the respective molecule is at least similar to an aromatic one and therefore remarkable. Algorithm 1 can easily be modified and refined, for instance by admitting other heteroatoms or expanding the search to other cycle sizes, e.g., to the *cyclopropenyl* cation (fulfills Hückel's rule with $n = 0$) or to the 16-cycle of porphyrin mentioned above.

3.1.2 Pattern Feature Vectors

The definitions of graph kernels below are based on the assumption that molecular graphs can be represented appropriately via a set of features. In a sense, these kernels are actually set kernels and we start with a consideration of *sets*, *multisets*, and different concepts of *set intersections*. For every set A of elements from an instance space $\mathcal{X}$ the *indicator function* $\mathbb{1}_A$ with

$$\mathbb{1}_A(x) = \begin{cases} 0 & : \quad x \notin A \\ 1 & : \quad x \in A \end{cases}$$

assigns the membership of $x \in \mathcal{X}$ to A. *Multisets* are a generalisation of sets where elements may appear multiply in one set. The elements of a multiset A can be viewed as tuples $(x, m_A(x))$ of the actual object x and its *multiplicity* $m_A(x)$, which is the frequency an object x appears in A [Singh et al., 2007]. The multiplicity function $m(\cdot)$ should not be confused with the number of m of unlabelled instances in semi-supervised learning (see Chapter 4). For a multiset A the indicator function is a *multiplicity function* with $\mathbb{1}_A(x) = m_A(x)$ and describes the structure of the multiset A [Singh et al.,

2007]. Operations between sets or multisets can be expressed in terms of indicator or multiplicity functions as well. If $A, B \subseteq \mathcal{X}$ are sets, the indicator function of their intersection is

$$\mathbb{1}_{A\cap B}(x) = \mathbb{1}_A(x) \cdot \mathbb{1}_B(x) = \min\{\mathbb{1}_A(x), \mathbb{1}_B(x)\}, \tag{3.1}$$

where the product and the minimum of set indicator function values are always equal. In the case of two multisets A and B, the product and the minimum of the multiplicity function values are in general different. Due to this, also different concepts of multiset intersections $A \wedge B$ and $A \sqcap B$ with

$$\mathbb{1}_{A\wedge B}(x) = \min\{\mathbb{1}_A(x), \mathbb{1}_B(x)\} \quad \text{and} \quad \mathbb{1}_{A\sqcap B}(x) = \mathbb{1}_A(x) \cdot \mathbb{1}_B(x) \tag{3.2}$$

can be considered, of which $A \wedge B$ describes the standard multiset intersection.

Let $\mathcal{A}$ be a class of graph patterns (for a precise definition see below). Without loss of generality, the number d of patterns is finite. Then $\mathcal{A}(G)$ denotes the respective subset (or sub-multiset) of patterns occurring in the labelled undirected graph $G \in \mathcal{G}$. We represent $\mathcal{A}(G)$ in form of a d-dimensional feature vector via a bijection between *dimension component* of the feature vector $i \in \{1, \ldots, d\}$ and the ith graph pattern. The *binary feature vector*

$$(\Phi_{\mathcal{A}}(G))_i = \begin{cases} 1 & : \text{ pattern } i \text{ occurs in } G \\ 0 & : \text{ pattern } i \text{ does not occur in } G \end{cases} \tag{3.3}$$

and the *counting feature vector*

$$(\Phi_{\mathcal{A}}(G))_i = \text{multiplicity of pattern } i \text{ in } G. \tag{3.4}$$

correspond to $\mathcal{A}(G)$ as a set or multiset, respectively. It will be stated explicitly which one is meant in the following applications. Based on the introduction of set and multiset intersections as well as binary and counting feature vectors, we define three graph kernels. The following definition and lemma are modified from [Ralaivola et al., 2005].

Definition 3.2 (Intersection, multiset intersection, and counting kernel)**.** Let $G, G' \in \mathcal{G}$ be labelled undirected graphs. Let $\mathcal{A}$ be a class of patterns.
(*i*) If $\mathcal{A}(G)$ and $\mathcal{A}(G')$ are sets

$$k_{\cap,\mathcal{A}}(G, G') = \langle \Phi_{\mathcal{A}}(G), \Phi_{\mathcal{A}}(G') \rangle \tag{3.5}$$

is called an *intersection kernel*.
(*ii*) In the case of multisets $\mathcal{A}(G)$ and $\mathcal{A}(G')$

$$k_{\wedge,\mathcal{A}}(G, G') = \min\{\Phi_{\mathcal{A}}(G), \Phi_{\mathcal{A}}(G')\} \quad \text{and} \tag{3.6}$$
$$k_{\sqcap,\mathcal{A}}(G, G') = \langle \Phi_{\mathcal{A}}(G), \Phi_{\mathcal{A}}(G') \rangle \tag{3.7}$$

denote a *multiset intersection kernel* and a *bag-of-words* or *counting kernel*, where the minimum in (3.6) is defined component-wise.

The normalised variants with respect to the set union of $k_{\cap,\mathcal{A}}$ and $k_{\wedge,\mathcal{A}}$ are also known as *Tanimoto kernel* (compare Table 2.1) and *MinMax kernel* [Ralaivola et al., 2005].

Lemma 3.3. *All kernels in Equations (3.5) to (3.7) and their normalised variants from above are well-defined.*

Proof. To start with, all kernels in Equations (3.5) to (3.7) are symmetric by definition. If the dimension d of the feature vector is finite, the intersection kernel $k_{\cap,\mathcal{A}}$ and the counting kernel $k_{\sqcap,\mathcal{A}}$ are positive semi-definite and symmetric because of the properties of the inner product in $\mathbb{R}^d$. For the proof of the positive semi-definiteness of $k_{\cap}$ we refer to the reasoning of Horváth et al. [2004]. The proof for $k_{\sqcap}$ is very similar, but uses the multiplicity function instead of the indicator function. For $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ and pattern feature multisets $\mathcal{A}_1, \ldots, \mathcal{A}_n$ of arbitrary labelled undirected graphs $G^1, \ldots, G^n$ we obtain

$$\begin{aligned}\sum_{i,j=1}^{n} \alpha_i \alpha_j k_{\sqcap,\mathcal{A}}(G^i, G^j) &= \sum_{i,j=1}^{n} \alpha_i \alpha_j \sum_{x \in \mathcal{X}_{\mathcal{A}}(i,j)} \mathbb{1}_{\mathcal{A}_i}(x) \mathbb{1}_{\mathcal{A}_j}(x) \\ &= \sum_{x \in \mathcal{X}_{\mathcal{A}}(i,j)} \left(\sum_{i=1}^{n} \alpha_i \mathbb{1}_{\mathcal{A}_i}(x) \right)^2 \geq 0,\end{aligned} \tag{3.8}$$

where $\mathcal{X}_{\mathcal{A}}(i,j) = \{x \in \mathcal{X} : \mathbb{1}_{\mathcal{A}_i}(x) > 0 \text{ and } \mathbb{1}_{\mathcal{A}_j}(x) > 0\}$, which shows the desired property. The scenario for the multiset intersection kernel $k_{\wedge}$ is different as it cannot be represented as a product of indicator or multiplicity functions. However, in order to proof the positive semi-definiteness of $k_{\wedge}$, we apply Theorem 21.11. [Klenke, 2006] and use that $\min\{s,t\}$, is the covariance function of a centered *Gaussian process* B named *Brownian motion*, which means that

$$\mathbb{E}B(s)B(t) = \mathrm{Cov}(B(s), B(t)) = \min\{s,t\} \tag{3.9}$$

holds true for $s, t \geq 0$. Hence, we conclude with Equations 3.2, 3.6, and 3.9 that

$$\begin{aligned}\sum_{i,j=1}^{n} \alpha_i \alpha_j k_{\wedge,\mathcal{A}}(G^i, G^j) &= \sum_{i,j=1}^{n} \alpha_i \alpha_j \sum_{x \in \mathcal{X}_{\mathcal{A}}(i,j)} \min\{\mathbb{1}_{\mathcal{A}_i}(x), \mathbb{1}_{\mathcal{A}_j}(x)\} \\ &= \sum_{i,j=1}^{n} \alpha_i \alpha_j \sum_{x \in \mathcal{X}_{\mathcal{A}}(i.j)} \mathbb{E}B_x(i) B_x(j) \\ &= \mathbb{E} \sum_{x \in \mathcal{X}_{\mathcal{A}}(i,j)} \left(\sum_{i=1}^{n} \alpha_i B_x(i) \right)^2 \geq 0.\end{aligned}$$

which proves the positive semi-definiteness of the multiset intersection kernel. For more information regarding stochastic processes we refer to Klenke [2006]. The normalised variants of the intersection, multiset intersection, and counting kernel are valid kernels as well, which is proven in Proposition 6 of Ralaivola et al. [2005]. □

3.1.3 The Cyclic Pattern Kernel

Now we are well-prepared to introduce the *cyclic pattern kernel* of Horváth et al. [2004]. Its construction requires further fundamental definitions of graph substructures in the following.

Definition 3.4 (Walk, path)**.** [Horváth et al., 2004] Let $G = (V, E)$ with the labelling function λ_G be a *labelled undirected graph*. A *walk* w is a k-sequence of edges

$$w = e_1, e_2 \ldots, e_k = \{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\},$$

such that $e_1, \ldots, e_k \in E$. This walk is a *simple path* if all $v_0, \ldots, v_k$ are pairwise distinct.

On the basis of walks and paths we introduce connected graphs as well as cycles and trees which will play an important role as graph patterns in the following.

Definition 3.5 (Cycle, tree)**.** [Horváth et al., 2004] A walk is a *simple cycle* if all $v_0, \ldots, v_k$ are pairwise distinct, except from $v_0 = v_k$. If for every pair of vertices there is a connecting simple path, the graph G is said to be *connected*. A cycle-free connected graph is called a *tree*.

Two simple cycles are assumed to be isomorphic if their edge sequences are cyclic permutations of each other.

Definition 3.6 (Biconnected component)**.** [Horváth et al., 2004] Let $G = (V, E)$ be a connected graph. A vertex $\tilde{v}$ is an *articulation vertex* of G if its elimination, i.e., $\tilde{V} = V \setminus \{\tilde{v}\}$ and correspondingly reduced set of edges $\tilde{E}$, results in a graph $\tilde{G} = (\tilde{V}, \tilde{E})$ that is not connected. A *biconnected component* of G is a maximal subgraph of G whose vertex set contains no articulation vertex.

Clearly, a biconnected component is either a subgraph in which every pair of nodes is connected via at least two different paths or it is an edge $e \in E$. Hence, biconnected components induce an edge-disjoint (and simple cycle-disjoint) decomposition of the connected graph G. An edge e either belongs to a biconnected component with more than one edge or the edge forms a biconnected component itself. In the first case, for every two edges e_1 and e_2 of the biconnected component, there is a simple cycle $w_{1,2}$ in G that contains e_1 and e_2. In the second case, the biconnected component does not contain a simple cycle and is called a *bridge*. In order to well-define the *cyclic* and *tree patterns* of a connected undirected graph, we need a *canonical representation* function r, i.e., a unique notation for both simple cycles and trees build from bridges that otherwise might be enumerated with different names. For a simple cycle C we consider all cyclic permutations (positive and negative orientation) of its labelled sequence of vertices and edges. We fix the canonical representation $r(C)$ to be the lexicographically smallest cyclic permutation (compare Figure 3.3). In the present scenario of trees T build from bridges, the representation $r(T)$ is more complex because the tree is free, i.e., has no vertex marked as root (compare Figure 3.4). Basically, its canonical representation is the rooted labelled tree T with lexicographically smallest systematic name string [Horváth et al., 2004]. Illustrating examples for the canonical representation of cycles and trees can be found in Figures 3.3 and 3.4.

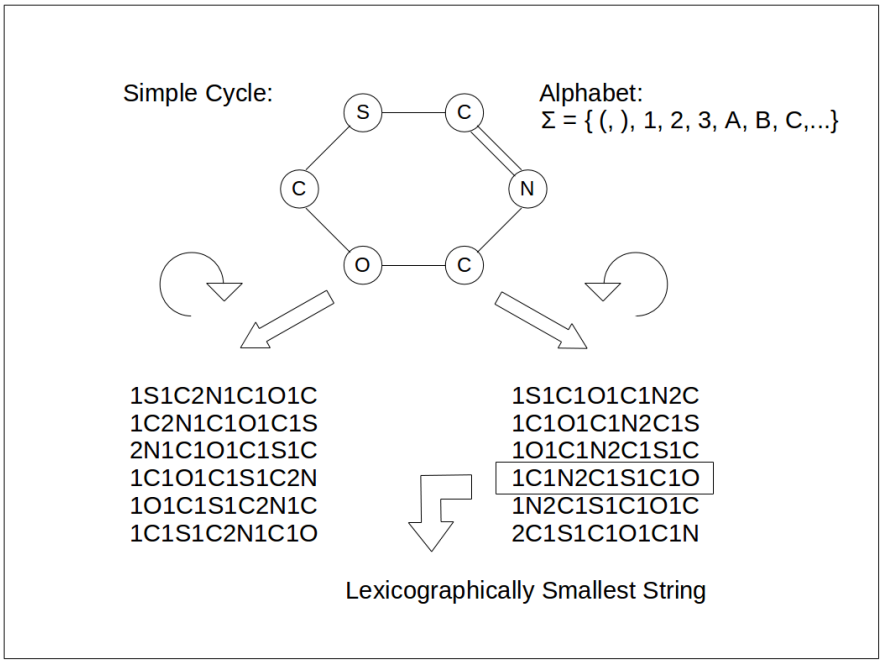
Definition 3.7 (Cyclic patterns)**.** [Horváth et al., 2004] Let r be the canonical representation function from above. The set of all *simple cycles* of an undirected labelled graph G is denoted by $\mathcal{S}(G)$. Accordingly, the set

$$\mathcal{C}(G) = \{r(C) : C \in \mathcal{S}(G)\}$$

are the *cyclic patterns* of a graph G.

Two graphs G and G' are said to be *isomorphic* if there is a bijection $\varphi : V \to V'$ such that $\{v, w\} \in E$ implies $\{\varphi(v), \varphi(w)\} \in E'$, as well as $\lambda(v) = \lambda'(\varphi(v))$ and $\lambda(\{v, w\}) = \lambda'(\{\varphi(v), \varphi(w)\})$. The canonical representation r of cycles is defined such that two isomorphic cycles $C_1, C_2 \in \mathcal{C}(G)$ always get the same unique identifier, i.e., $r(C_1) =$

FIGURE 3.3: Canonical representation of a simple cycle



$r(C_2)$. Hence, if there are two isomorphic cycles in $\mathcal{S}(G)$, their canonical representation will be the same and there is only one representative pattern in $\mathcal{C}(G)$ [Ullrich et al., 2016b]. It might be of interest to know how often an isomorphic pattern appears in a graph. In this case, we can also define the cyclic patterns as multiset

$$\mathcal{C}(G) = \{(r(C), m(r(C))) : C \in \mathcal{S}(G)\},$$

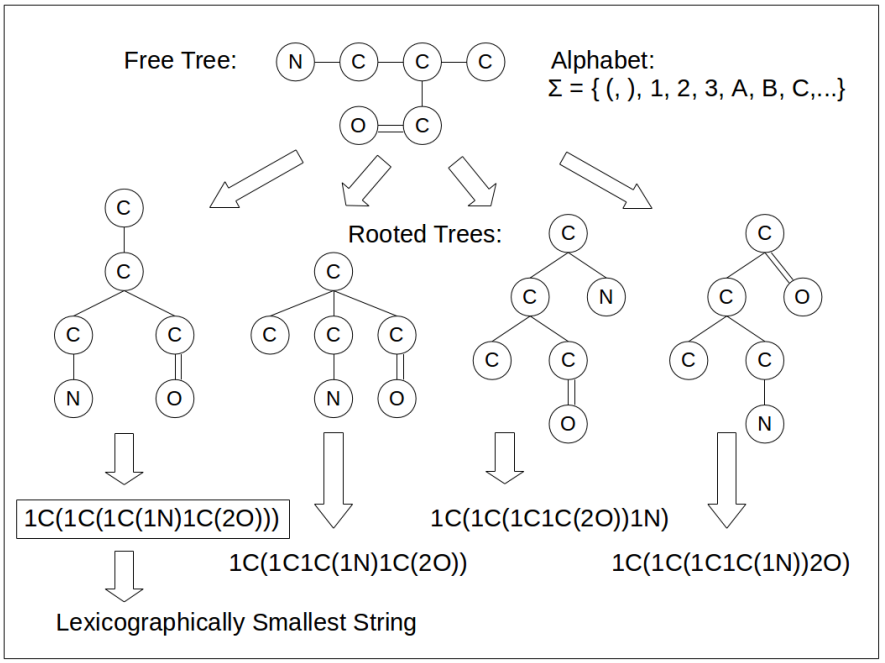


FIGURE 3.4: Canonical representation of a free tree

where $m(r(C))$ is the actual multiplicity of pattern $r(C)$ in G.

Definition 3.8 (Tree patterns)**.** [Horváth, 2005] With $\mathcal{B}(G)$ we call the set of all *bridges* of G. The *tree patterns* of G are defined by

$$\mathcal{T}(G) = \{r(T) : T \text{ is a connected component of } G \text{ with edges from } \mathcal{B}(G)\}.$$

The subgraph of G build from the edges in $\mathcal{B}(G)$ will be called *forest* (of trees).

Analogous to the cyclic patterns, the tree patterns can also be assumed a multiset

$$\mathcal{T}(G) = \{(r(T), m(r(T))) : T \text{ is a connected subgraph of } G \text{ with edges from } \mathcal{B}(G)\}.$$

Having defined cyclic and tree patterns we introduce the first important graph kernel for our further considerations.

Definition 3.9 (Cyclic pattern kernel)**.** [Horváth et al., 2004] Assume G and G' are two undirected labelled graphs. The *cyclic pattern kernel* (CPK) is defined as

$$k_{CP}(G, G') = |\mathcal{C}(G) \cap \mathcal{C}(G')| + |\mathcal{T}(G) \cap \mathcal{T}(G')|. \tag{3.10}$$

As an intersection kernel k_{CP} is well-defined according to Lemma 3.3 above. The feature space of cyclic and tree patterns is theoretically infinitely large. Nevertheless, for our practical purposes regarding molecular fingerprints in Section 3.4, suppose we have a database of graphs $\mathcal{G}$ with finite cyclic and tree pattern sets $\mathcal{C}(\mathcal{G})$ as well as $\mathcal{T}(\mathcal{G})$, where $\mathcal{C}(\mathcal{G})$ and $\mathcal{T}(\mathcal{G})$ are the canonical generalisations of $\mathcal{C}(G)$ and $\mathcal{T}(G)$, respectively. The following lemma is modified from [Horváth et al., 2004] and a direct consequence of the previous definitions.

Lemma 3.10. *Let $\mathcal{G}$ be a database of connected undirected labelled graphs with cyclic and tree patterns $\mathcal{C}(\mathcal{G})$ and $\mathcal{T}(\mathcal{G})$ of cardinalities $d_\mathcal{C}$ and $d_\mathcal{T}$. Let $G \in \mathcal{G}$ be a labelled undirected graph. We consider the enumeration $c_1, \ldots, c_{d_\mathcal{C}}$ and $t_1, \ldots, t_{d_\mathcal{T}}$ of the elements of $\mathcal{C}(\mathcal{G})$ and $\mathcal{T}(\mathcal{G})$. The vectors $\Phi_\mathcal{C}(G)$ and $\Phi_\mathcal{T}(G)$ denote the $d_\mathcal{C}$- and $d_\mathcal{T}$-dimensional binary or counting pattern feature vectors of G with respect to cycles and trees according to Equation 3.3. Having finite feature vectors, the CPK in Equation 3.10 can be expressed equivalently as linear kernel*

$$k_{CP}(G, G') = \langle \Phi_\mathcal{C}(G), \Phi_\mathcal{C}(G') \rangle_{d_\mathcal{C}} + \langle \Phi_\mathcal{T}(G), \Phi_\mathcal{T}(G') \rangle_{d_\mathcal{T}}$$

for $G, G' \in \mathcal{G}$.

Horváth et al. [2004] showed that enumerating k elements from $\mathcal{C}(G)$ of a graph G with n vertices is NP-hard. That means it belongs to the class of problems that can be solved with a non-deterministic *Turing machine* in polynomial time and the solution of other NP-problems can be reduced to the solution of the NP-hard problem [Turing, 1950]. Based on a result of Read and Tarjan [1975] on the computation of simple cycles, Horváth et al. [2004] proved that for a database $\mathcal{G}$ with at most n_{max} vertices and m_{max} edges, as well as

$$|\mathcal{S}(G)| \leq k \quad \text{for all} \quad G \in \mathcal{G} \tag{3.11}$$

the calculation of $\mathcal{C}(G)$ and $\mathcal{T}(G)$ has time complexity $\mathcal{O}(|\mathcal{G}|((k+2)n_{max}+2m_{max}))$. The so-called *bounded cyclicity* from Equation 3.11 makes many practical problems feasible in terms of computation.

3.1.4 Shortest Path Kernel

Given a labelled undirected graph $G = (V, E)$, we enhance the expressiveness of tree patterns $\mathcal{T}(G)$ and cyclic patterns $\mathcal{C}(G)$ by computing shortest paths between pairs of vertices [Ullrich et al., 2016b]. Because of the complexity of the general case, we only consider shortest paths within the forest of trees of the graph G, which means only between vertices that build the bridges $\mathcal{B}(G)$. To this aim we have to modify the graph structure with respect to the biconnected components. The vertices contained in cycles are

$$V_{\text{cycles}} = \{v : v \text{ is a vertex of a simple cycle in } \mathcal{S}(G)\}.$$

We define a contracted version $\overline{G} = (\overline{V}, \overline{E})$ of graph G by construction (compare also Algorithm 2 in Appendix C). Initially, the vertex set is $\overline{V} = V \setminus V_{\text{cycles}}$. Now we consider the decomposition of $\mathcal{S}(G)$ into biconnected components according to Definition 3.6. Let $B \in \mathcal{S}(G)$ be a biconnected component. As a (contracted) representative of B we add a new vertex v_B with a so far unused label l_{bc} to $\overline{V}$. We go on as described for every biconnected component in G. Consequently, the edge set of $\overline{G}$ is basically equal to the original set of bridges $\overline{E} = \mathcal{B}(G)$. Though, the link vertices of bridges with biconnected components from the original graph G are now contained in a vertex v_B for a biconnected component B and carry the label l_{bc}, which should be kept in mind regarding Equation 3.12 below.

Definition 3.11 (Contracted graph)**.** [Ullrich et al., 2016b] We refer to the notation and construction of $\overline{V}$ and $\overline{E}$ from above. If $G = (V, E)$ with labelling function λ_G is a labelled undirected graph, the graph $\overline{G} = (\overline{V}, \overline{E})$ is called the corresponding *contracted graph*. For the labelling $\lambda_{\overline{G}}$ of the contracted graph $\overline{G}$

$$\lambda_{\overline{G}}(v) = \begin{cases} l_{\text{bc}} & : \quad v = v_B \text{ for biconnected component } B \\ \lambda_G(v) & : \quad \text{else} \end{cases}$$

as well as

$$\lambda_{\overline{G}}(\{v, w\}) = \lambda_G(\{v, w\}) \tag{3.12}$$

holds true.

Because of Definition 3.6 the contracted graph is well-defined if the molecular compounds are planar and no two biconnected components are linked by an articulation vertex. Furthermore, the contracted graph is a labelled, undirected, and connected graph.

A *shortest path* between two nodes v and v' of a connected graph G is the one with the shortest length, i.e., edge number along the path. Shortest paths can be used to define graph kernels as well. The subsequent two definitions are modified from [Borgwardt and Kriegel, 2005].

Definition 3.12 (Shortest paths)**.** Let $G = (V, E)$ be a labelled undirected graph. We call

$$\mathcal{SP}(G) = \{P : P \text{ is a shortest path between vertices } v \text{ and } v' \text{ of } G\}$$

the *shortest paths* of G.

Definition 3.13 (shortest path kernel)**.** Let $\mathcal{SP}(G_1)$ and $\mathcal{SP}(G_2)$ be the shortest paths of G_1 and G_2, respectively. We denote

$$k_{SP}(G_1, G_2) = \sum_{P_1 \in \mathcal{SP}(G_1)} \sum_{P_2 \in \mathcal{SP}(G_2)} k_{\text{path}}(P_1, P_2) \tag{3.13}$$

shortest path kernel, where k_{path} is an appropriate kernel for paths.

A path kernel k_{path} can be defined, e.g., as a product of a kernel for vertices and one for edges [Schölkopf and Smola, 2002]. Between two nodes v and v' of a general connected graph G there might be different shortest paths connecting these vertices. However, as we only consider shortest paths between vertices in V_{forest} within the contracted graph, there is always only one unique shortest path. The transition from shortest paths to the corresponding patterns again requires a canonical representation which takes into account the order of vertex and edge labels. As paths are special trees, the canonical representation function r from above can be used here as well. It is simply the lexicographically shortest labelled edge sequence in one direction and reverse (compare Figure 3.4). In contrast to simple cycles and general trees, shortest paths have only two potential start vertices which simplifies their naming. The following definition is a variant of the one used in [Horváth et al., 2004].

Definition 3.14 (Shortest path patterns)**.** Let G be a labelled undirected graph with contracted graph $\overline{G}$ and r the canonical representation function from Section 3.1.3 above. We call

$$\mathcal{P}(G) = \{r(P) : P \in \mathcal{SP}(\overline{G})\}$$

the *shortest path patterns.*

Analogous to cyclic and tree patterns, the shortest path patterns can be assumed a set $\mathcal{P}(G)$ or a multiset

$$\mathcal{P}(G) = \{(r(P), m(r(P))) : P \in \mathcal{SP}(\overline{G})\},$$

where not only the existence but also the cardinalities of the found patterns are registered in the multiplicity function m.

We only consider shortest paths between vertices of found trees linked with a contraction vertex representing the former biconnected component of cycles. However, the determination of shortest paths $\overline{G}$ requires the initial decomposition of a graph G into biconnected components and bridges. The contracted graph $\overline{G}$ is again a tree and between every two vertices in a tree there is only one path, the shortest path. The *Floyd-Warshall algorithm* of Floyd [1962] for the determination of shortest paths has complexity $\mathcal{O}(n^3)$ where n is the number of vertices. As all vertices of a biconnected component B are reduced to a single vertex v_B this will be much faster in $\overline{G}$ than in the original graph G.

3.1.5 Weisfeiler-Lehman Graph Kernel

Another important type of graph kernels from the literature are the so-called *Weisfeiler-Lehman* (WL) graph kernels [Shervashidze et al., 2010]. They are defined for labelled graphs, where only the vertices carry labels. However, the involved *Weisfeiler-Lehman*

labelling can still be applied to labelled undirected graphs with edge labels according to Definition 3.1 by ignoring the edge labels. The WL labelling originates from the WL test of graph isomorphism [Shervashidze et al., 2010] and requires a recursively defined labelling function. It consists in a successive label expansion with label information from adjacent vertices. Suppose, we have an undirected graph $G = (V, E)$ with vertex and edge labels from an alphabet Σ according to the original labelling function λ_G. We explain the WL labelling function $\lambda_G^h : V \cup E \to \Sigma$ step-wise. The initial WL labelling (step $h = 0$) fulfills

$$\lambda_G\big|_V = \lambda_G^0\big|_V \quad \text{and} \quad \lambda_G\big|_E = \lambda_G^0\big|_E. \tag{3.14}$$

As the WL labelling should not affect the edge labels, the equality

$$\lambda_G\big|_E = \lambda_G^h\big|_E \tag{3.15}$$

is also valid for $h > 0$. The WL labels of vertices in step $h + 1$ are defined recursively. For a vertex $v' \in V$ we consider its label $\lambda_G^h(v')$ and the multiset of labels of its adjacent vertices

$$A_{v'} = \left\{ (l, m(l)) : l = \lambda_G^h(w) \text{ and } \{v', w\} \in E \right\}.$$

This multiset of adjacent labels should be sorted and appended to the leading label $\lambda_G^h(v')$ respecting their multiplicities. For example, if the label of vertex v' was N and the multiset of labels was $\{(C, 3), (O, 2), (N, 1), (S, 1)\}$, then the result would be *NCCCNOOS*, which is a word over the alphabet Σ. A renaming function $\mathfrak{r} : \Sigma^* \to \Sigma$ should assign an unused letter from Σ to the concatenated string from the set of words Σ^*. Finally, for the WL label of v in step $h + 1$

$$\lambda_G^{h+1}(v) = \mathfrak{r}\left\{ \text{concatenate}\left(\lambda_G^h(v) \ , \ \text{sort}\left(A_v\right) \right) \right\} \tag{3.16}$$

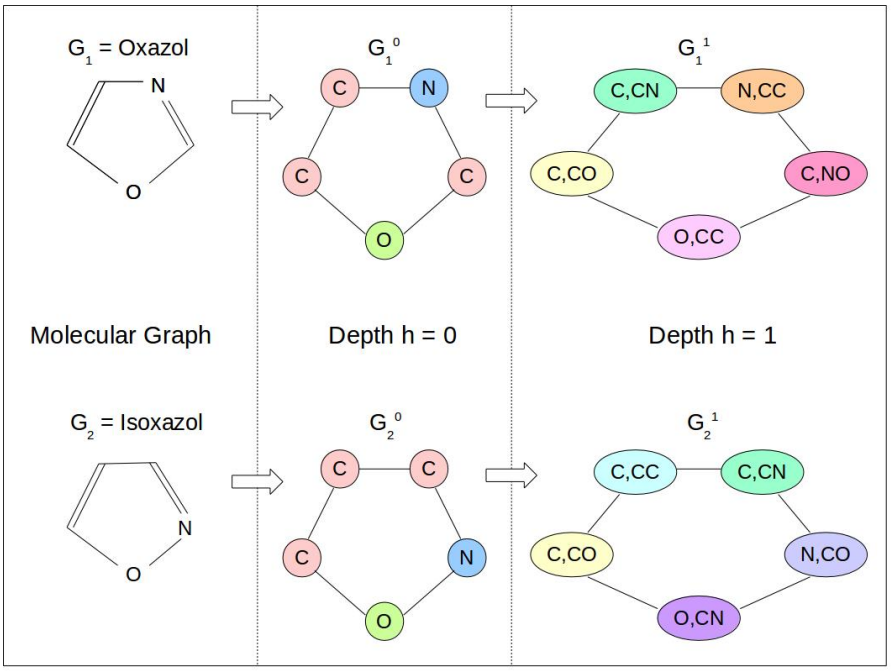
holds true. The calculation in Equation 3.16 requires a sorting algorithm and a renaming function $\mathfrak{r}$ for the concatenated label strings. Using additional symbols like commas and parantheses, it would be possible to omit the renaming function $\mathfrak{r}$. In favor of their information gain, WL labels could then become large depending on step size h and the precise structure of graph G. A short example of a WL labelling procedure can be found in Figure 3.5, which also shows a proof of graph non-isomorphism for the involved graphs. The emerging labels essentially cover all possible walks of length h coming from the respective vertex, i.e., the neighbourhood information of a vertex up to a maximal depth. Therefore, we will denote the step size h also by *depth* of the WL labelling procedure. The following two definitions are leaned to the work of Shervashidze et al. [2010].

Definition 3.15 (WL labelling)**.** The function λ_G^h, $h = 0, 1, 2, \ldots$ defined via Equations 3.14, 3.15, and 3.16 is called *Weisfeiler-Lehman labelling function.* Let $G = (V, E)$ with labelling function λ_G be a labelled undirected graph. We denote $G_h = (V, E)$ with labelling function λ_G^h the graph G's *Weisfeiler-Lehman labelling* (WL labelling) of *depth* h.

We will synonymously call G_h WL labelled graph (of depth h). The kernel defined in the definition below directly applies WL labelled graphs.

Definition 3.16 (Weisfeiler-Lehman kernel)**.** Let G^1 and G^2 be labelled and undirected graphs with corresponding sequences $G_0^1, G_1^1, G_2^1, \ldots$ and $G_0^2, G_1^2, G_2^2, \ldots$ of WL labellings.

FIGURE 3.5: Example of a WL labelling of depth $h = 1$ for two molecular graphs



Having an arbitrary kernel function $k : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ for labelled undirected graphs, a *Weisfeiler-Lehman kernel* (WLK) up to maximal depth H is defined as

$$k_{WL}^{H}(G^1, G^2) = \sum_{h=0}^{H} k(G_h^1, G_h^2). \tag{3.17}$$

The fourth group of graph patterns we consider will be the vertex labels generated in different depths of the WL labelling procedure. It is reasonable to assume that an information about presence or absence of atom types and their adjacency relations included in WL labels gives important information about the binding capacity for a potential ligand. The subsequent definition is an expansion of Definitions 3.7, 3.8, and 3.14 above for cyclic, tree, and shortest path patterns.

Definition 3.17 (Label patterns)**.** [Ullrich et al., 2016b] Let G_h be the WL labelling of depth $h \in \mathbb{N}$ of the labelled undirected graph $G = (V, E)$. The set

$$\mathcal{L}(G_h) = \{\lambda_G^h(v) : v \in V\}$$

is called *label patterns* of G_h.

If we are interested in the number of equal labels that appear in a certain labelled graph, we consider the multiset of labels

$$\mathcal{L}(G_h) = \{(l, m(l)) : l = \lambda_G^h(v) \text{ and } v \in V\}$$

analogous to cyclic, tree, and shortest path patterns. We do not need a canonical representation r or renaming function $\mathfrak{r}$ for the definition of $\mathcal{L}(G_h)$ as every label is assumed to be a (renamed) canonical representation by definition. We omit edge labels in Definition 3.17 as basically all small organic molecules contained in affinity datasets

exhibit single, double, and aromatic bonds and, hence, the vast majority of molecules would carry the same edge labels. Together with the label patterns $\mathcal{L}(G_h)$, we will also consider the sets or multisets $\mathcal{C}(G_h), \mathcal{T}(G_h), \mathcal{P}(G_h)$ of cyclic, tree, and shortest path patterns of a WL labelled graph G_h in the following section.

3.2 The Multi-Pattern Kernel

The concept behind the kernel class proposed below is to combine WL labelling with different graph patterns. As a consequence, the WL labels of depth h will, on the one hand, be the basis for the canonical representation of patterns and, on the other hand, they will be interpreted as additional pattern class. We will utilise the symbols $\mathcal{C}$, $\mathcal{T}$, $\mathcal{L}$, and $\mathcal{P}$ both as index and identifier for the respective graph pattern classes.

Definition 3.18 (Pattern kernel)**.** [Ullrich et al., 2016b] Let $\mathcal{V} \in \{\mathcal{C}, \mathcal{T}, \mathcal{P}, \mathcal{L}\}$ be a graph pattern class, k an appropriate set kernel, and G, G' be two labelled undirected graphs. The *cumulative pattern kernel* of depth H is defined via

$$k_{\mathcal{V}}^{H,cum}(G, G') = \sum_{h=0}^{H} k(\mathcal{V}(G_h), \mathcal{V}(G'_h)), \tag{3.18}$$

where either the set or the multiset $\mathcal{V}(\cdot)$ for the respective pattern class can be inserted. We call

$$k_{\mathcal{V}}^{H}(G, G') = k(\mathcal{V}(G_H), \mathcal{V}(G'_H)) \tag{3.19}$$

the *non-cumulative pattern kernel* of depth H.

The kernel k in Definition 3.18 is defined on pairs of graph sets. It would be possible to assume k to be a graph kernel in the sense of

$$k(\mathcal{V}(\cdot), \mathcal{V}(\cdot)) : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$$

as well, analogous to the introduction of CPK in Definition 3.9 above.

There are a couple of links between different graph kernels and their generation concepts which we list in the following. Firstly, the cumulative pattern kernel in Equation 3.18 can be considered a generalisation of the WLK in Equation 3.17 [Shervashidze et al., 2010]. Secondly, it can also be understood as an extension of the CPK in Equation 3.10 [Horváth et al., 2004] or the SPK in Equation 3.13 [Borgwardt and Kriegel, 2005]. Furthermore, the non-cumulative variant in Equation 3.19 is a generalisation of the non-negatively weighted WLK [Shervashidze et al., 2010]. Finally, the development of the ECFPd-fingerprints (compare Section 1.3.2) is very similar to the cumulative pattern kernel $k_{\mathcal{L}}^{H,cum}$ of depth H using label patterns. The diameter index d is twice as big as the depth parameter H. Further differences arise, e.g., from the hashing scheme of the respective labelling function [Rogers and Hahn, 2010].

We further enhance the diversity of the graph pattern-based molecular kernels by the following definition.

Definition 3.19 (Multi-pattern kernel)**.** [Ullrich et al., 2016b] Let $k_{\mathcal{V}}^{\theta_{\mathcal{V}}}$, $\mathcal{V} \in \{\mathcal{C}, \mathcal{T}, \mathcal{P}, \mathcal{L}\}$, be a pattern kernel according to Definition 3.18, where $\theta_{\mathcal{V}} = (H(\mathcal{V}), i_m(\mathcal{V}), i_c(\mathcal{V}))$ is a

multi-index of the pattern-dependent depth $H(\mathcal{V})$, the binary index for pattern sets or multisets $i_m(\mathcal{V})$, and the binary index for the cumulative or non-cumulative variant $i_c(\mathcal{V})$. Let $b_{\mathcal{C}}, b_{\mathcal{T}}, b_{\mathcal{P}}, b_{\mathcal{L}} \geq 0$ be non-negative weight coefficients. For two labelled undirected graphs G and G' the *multi-pattern kernel* (MPK) is defined as

$$k_{MP}(G, G') = \sum_{\mathcal{V} \in \{\mathcal{C}, \mathcal{T}, \mathcal{P}, \mathcal{L}\}} b_{\mathcal{V}} \cdot k_{\mathcal{V}}^{\theta_{\mathcal{V}}}(G, G'). \tag{3.20}$$

On the basis of this very flexible MPK, we point to the suggestion of Mack [2014] to utilise binary coefficients $b_{\mathcal{V}}$. But other than Mack [2014], we consider real-valued coefficients $b_{\mathcal{V}}$ in order to include pattern kernels in the magnitude of their importance. As the MPK in Equation 3.20 is a kernel linear combination, MKL approaches can be employed to determine appropriate coefficients for the learning task at hand. The index $\mathcal{V}$ together with a precise specification of the actual kernel will be summarised to the view index v. The index v is used in the context of multi-view approaches in the subsequent description of MKL algorithms as well as in Chapters 2, 4, and 5.

3.3 Regression with Kernel Linear Combinations

As already stated above, our intention is to solve a regression task by exploiting the diversity of MPKs. To this aim, we apply *multiple kernel learning* (MKL) [Gönen and Alpaydin, 2011] as a subfield of multi-view learning (see Section 1.2). The MKL approaches we consider have in common the search for a predictor function that relates to a linear combination of kernels $k_1, \ldots, k_M$

$$k_b = \sum_{v=1}^{M} b_v k_v, \tag{3.21}$$

where $b_1, \ldots, b_M \geq 0$. Because of the closure properties of kernels from Section 2.5, k_b is a kernel function again. Furthermore, MPKs from Definition 3.19 above are such kernel linear combinations with $b_v = b_{\mathcal{V}}$. Therefore, the pairing of MKL with MPKs allows for an optimal assignment of the coefficients $b_{\mathcal{V}}$, $\mathcal{V} \in \{\mathcal{C}, \mathcal{T}, \mathcal{P}, \mathcal{L}\}$, in Equation 3.20 and will be denoted with *MPK-MKL*. In this context, binary coefficients $b_{\mathcal{C}}, \ldots, b_{\mathcal{L}}$, as used by Mack [2014], would rather correspond to a choice than a weighting of pattern kernels for the regression task. The case of real-valued coefficients $b_{\mathcal{V}}$ allows for a better adjustment of the linear combination parameters of the respective pattern class $\mathcal{V}$ towards the precise learning task.

We already introduced the kernelised versions of RLSR (see Section 2.6.1) and SVR (see Section 2.6.2) as single-view approaches that operate in an implicit feature space. Now we consider two multi-view algorithms below which can be interpreted as RLSR and SVR utilising a kernel linear combination in order to solve the optimisation problem at hand. The following lemma substantiates the application of the representer theorem in the considered regularised risk functionals below.

Lemma 3.20. *Let $k_1, \ldots, k_M$ be kernel functions and $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ be labelled training examples. With $\mathcal{H}_b$ we denote the RKHS with reproducing kernel $k_b = \sum_{v=1}^{M} b_v k_v$, $b_1, \ldots, b_M \geq 0$, from Equation 3.21. For a loss function ℓ, an appropriate*

convex regularisation term $\Psi(b)$, and hyperparameters $\nu > 0$ and $\Lambda > 0$, the solution of

$$\min_{f \in \mathcal{H}_b, b \geq \mathbf{0}_M} \|f\|_{\mathcal{H}_b}^2 + \nu \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \Lambda \Psi(b) \tag{3.22}$$

has a representation in form of

$$f(x) = \sum_{i=1}^{n} \pi_i \sum_{v=1}^{M} b_v k_v(x_i, x), \tag{3.23}$$

which represents a kernel linear combination of k_b.

Proof. The optimisation problem in Equation 3.22 can be reformulated as

$$\min_{b \geq \mathbf{0}_M} \left(\min_{f \in \mathcal{H}_b} \|f\|_{\mathcal{H}_b}^2 + \nu \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \Lambda \Psi(b) \right), \tag{3.24}$$

where for fixed coefficients b the term $\lambda \Psi(b)$ is a constant. For the inner optimisation

$$f_b = \operatorname*{argmin}_{f \in \mathcal{H}_b} \|f\|_{\mathcal{H}_b}^2 + \nu \sum_{i=1}^{n} \ell(y_i, f(x_i))$$

we may apply the representer theorem and obtain a representation

$$f_b(x) = \sum_{i=1}^{n} \pi_i k_b(x_i, x) = \sum_{i=1}^{n} \pi_i \sum_{v=1}^{M} b_v k_v(x_i, x),$$

which is already the desired result from Equation 3.23. The final predictor f equals f_{b^*} for optimal coefficients $b^* \geq \mathbf{0}_M$ from Equation 3.24. □

Lemma 3.20 and its proof are analogous to Theorem 2.21 and the corresponding proof in a multi-view setting. We will apply two MKL approaches of Cortes et al. [2009] and Vishwanathan et al. [2010], respectively. Both approaches are examples of the optimisation in Equation 3.22 [Oneto et al., 2016]. More precisely, the ℓ_2-MKL algorithm in Section 3.3.1 uses the squared loss function and a box constraint for the linear combination parameters $b \geq \mathbf{0}_M$. The second ε-MKL algorithm introduced in Section 3.3.2 utilises the ε-insensitive loss and an ℓ_p-norm regularisation of the coefficients b.

In the present chapter, the Gram matrix K of a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ will generally be defined as

$$K = (k(x_i, x_j))_{i,j=1}^{n} \in \mathbb{R}^{n \times n}$$

for labelled training instances $x_1, \ldots, x_n \in \mathcal{X}$. The MKL approaches from Sections 3.3.1 and 3.3.2 below require the simultaneous calculation of both the kernel expansion coefficients π_i and the parameters b_v of the kernel linear combination in Equation 3.23. In contrast to RLSR and SVR, the MKL objectives cannot be formulated as quadratic program (QP).

3.3.1 Learning Kernel Ridge Regression

The first MKL approach for regression we consider utilises the squared loss function ℓ_2 for the empirical error of labelled training examples. The corresponding algorithm was introduced by Cortes et al. [2009] and is called *learning kernel ridge regression* (LKRR). The intention behind is to learn a predictor function f from an RKHS $\mathcal{H}_b$ with reproducing kernel $k_b : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which is a linear combination of view-related kernels $k_1, \ldots, k_M$ with linear coefficients $b_1, \ldots, b_M \geq 0$ according to Equation 3.21. LKRR is essentially an RLSR approach that applies a kernel linear combination with an additional regularisation constraint for the parameters b_v. Concerning the practical approach of ligand affinity prediction, the idea is to use an MPK for k_b and perform MPK-MKL experiments (see Section 3.4.2). For an appropriate assignment of indices this means, we will insert pattern kernels $k_{\mathcal{V}}^{\theta_{\mathcal{V}}}$ as view kernels k_v and linear combination coefficients $b_v = b_{\mathcal{V}}$ as used in Equation 3.21. For the sake of consistency, we will rename the LKRR algorithm of Cortes et al. [2009] in the following definition.

Definition 3.21 (ℓ_2-MKL)**.** [Cortes et al., 2009] Let $k_1, \ldots, k_M$ be kernel functions defined on an instance space $\mathcal{X}$ and k_b be the kernel linear combination according to Equation 3.21 with RKHS $\mathcal{H}_b$ and linear coefficients $b_1, \ldots, b_M \geq 0$. Let $(x_1, y_1), \ldots, (x_n, x_n)$ be labelled training examples from $\mathcal{X} \times \mathcal{Y}$. The optimisation

$$\min_{f \in \mathcal{H}_b,\ b \geq \mathbf{0}_M} \|f\|_{\mathcal{H}_b}^2 + \nu \sum_{i=1}^{n} |y_i - f(x_i)|^2, \qquad \text{s.t. } \|b - b^0\| \leq \Lambda \tag{3.25}$$

where $\nu, \Lambda > 0$ are hyperparameters and $b^0 \geq \mathbf{0}_M$ the initial linear coefficients, is called *ℓ_2-multiple kernel learning* (ℓ_2-MKL).

In contrast to the RLSR formulation in Equation 2.25, Cortes et al. [2009] used the parameter ν in Equation 3.25 attached to the empirical risk term.

Lemma 3.22. *[Cortes et al., 2009] Let $K_b, K_1, \ldots, K_M \in \mathbb{R}^{n \times n}$ be the Gram matrices of the kernel functions $k, k_1, \ldots, k_M$ and Y be the vector of real-valued labels. For $\pi \in \mathbb{R}^n$ we set*

$$w = (\pi^T K_1 \pi, \ldots, \pi^T K_M \pi)^T.$$

The solution f of the minimisation in Equation 3.25 has got a representation in terms of b and π corresponding to Equation 3.23. For $b \geq \mathbf{0}_M$, $\pi \in \mathbb{R}^n$, and initial linear coefficients $b^0 \geq \mathbf{0}_M$

$$b = b^0 + \frac{w}{\|w\|}$$

holds true, where $\pi = (K_b + 1/\nu \cdot \mathbf{I}_n)^{-1} Y$.

Proof. An extended version of the proof of Cortes et al. [2009] for Lemma 3.22 can be found in Appendix A. □

The presented solution for ℓ_2-MKL is not a closed formula as $b = b(\pi)$ and $\pi = \pi(b)$ are optimised simultaneously. Nevertheless, one can find the approximate solution via

Lemma 3.22 and an iterative algorithm [Cortes et al., 2009] that can be found in Appendix C.

3.3.2 ε-Insensitive Loss MKL

An alternative MKL regression approach proposed by Vishwanathan et al. [2010] utilises the ε-insensitive loss ℓ_ε for the calculation of the empirical risk. Furthermore, Vishwanathan et al. [2010] applied the ℓ_p-norm with $p > 1$ as regularisation term for the linear coefficients b in the kernel linear combination k_b according to Equation 3.21.

Definition 3.23 (ε-MKL)**.** [Vishwanathan et al., 2010] Let $\mathcal{H}_b$ be the RKHS of the kernel linear combination k_b from Equation 3.21, where $k_1, \ldots, k_M : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ are kernel functions. Let $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ be training examples. The optimisation

$$\min_{f \in \mathcal{H}_b} \frac{1}{2}\|f\|^2_{\mathcal{H}_b} + \nu \sum_{i=1}^{n} \max\{|y_i - f(x_i)| - \varepsilon, 0\} + \frac{\Lambda}{2}\|b\|_p^2 \tag{3.26}$$

is called *ε-multiple kernel learning* (ε-MKL), where $\varepsilon, \Lambda, \nu > 0$, $b \geq \mathbf{0}_M$ are hyperparameters.

Analogous to ℓ_2-MKL, the challenge for ε-MKL is to learn both the kernel expansion coefficients $\pi \in \mathbb{R}^n$ of Equation 2.22 and the kernel linear combination parameters $b \geq \mathbf{0}_M$ in Equation 3.20 for the predictor function f simultaneously. The solution for ε-MKL will be presented in the following lemma.

Lemma 3.24. *[Vishwanathan et al., 2010] We consider the view-related kernel functions $k_1, \ldots, k_M$ and corresponding Gram matrices $K_1, \ldots, K_M \in \mathbb{R}^{n \times n}$. Additionally, let k_b be the reproducing kernel from Equation 3.21 with RKHS $\mathcal{H}_b$. Assume, for hyperparameters $p > 1$ and $q > 1$ the relation $\frac{1}{p} + \frac{1}{q} = 1$ holds true. The solution f of ε-MKL from Equation 3.26 has got a parameterisation in form of*

$$f(\cdot) = \sum_{i=1}^{n} \pi_i \sum_{v=1}^{M} b_v k_v(x_i, \cdot).$$

The parameters $b \geq \mathbf{0}_M$ and $\pi \in \mathbb{R}^n$ can be determined via the dual optimisation

$$\begin{aligned} \max_{\alpha, \hat{\alpha} \geq \mathbf{0}_n} \; & -\frac{1}{8\Lambda} \left(\sum_{v=1}^{M} \left((\alpha - \hat{\alpha})^T K_v (\alpha - \hat{\alpha}) \right)^q \right)^{\frac{2}{q}} + (\alpha - \hat{\alpha})^T Y - \varepsilon(\alpha + \hat{\alpha}) \mathbf{1}_n, \\ & \text{s. t. } \{\mathbf{0}_n \leq \alpha, \hat{\alpha} \leq \nu \mathbf{1}_n\}, \end{aligned}$$

such that additionally

$$b_v = \frac{1}{2\Lambda} \left((\alpha - \hat{\alpha})^T K_v (\alpha - \hat{\alpha}) \right)^{\frac{q}{p}} \left(\sum_{v=1}^{M} \left((\alpha - \hat{\alpha})^T K_v (\alpha - \hat{\alpha}) \right) \right)^{\frac{1}{q} - \frac{1}{p}}, v = 1, \ldots, M,$$

and $\pi = \alpha - \hat{\alpha}$ is valid.

Proof. The regression case of Vishwanathan et al. [2010]'s proof can be found in Appendix A. □

TABLE 3.1: Dataset identifiers in preliminary single-view experiments (part A)

Protein	P23946	P09871	Q9Y5Y6	P42574	P07384
Identifier	DS1	DS3	DS5	DS7	DS9
Protein	P08709	P00750	P29466	P00747	P08246
Identifier	DS11	DS13	DS15	DS17	DS19

TABLE 3.2: Dataset identifiers in MKL experiments (parts B and C)

Protein	Q99895	P25774	P17655	P00740	P07339
Identifier	DS2	DS4	DS6	DS8	DS10
Protein	P43235	P07858	P07711	P00749	P07477
Identifier	DS12	DS14	DS16	DS18	DS20

3.4 Empirical Evaluation

In the present empirical section we investigate the benefit of using different molecular representations in a chemoinformatics problem. More precisely, we apply and analyse the MPK-MKL approach for ligand affinity prediction. The present section is based on the empirical section in [Ullrich et al., 2016b]. The presentation of results will be described in detail below.

3.4.1 Datasets, Implementation, and Experimental Setting

The experiments were performed with 20 datasets, such that each set contains ligands of one of 20 human proteins. A set contains between 90 and 986 ligands of the respective protein and every ligand is labelled with its real-valued affinity towards the respective protein (compare Section 1.3.1). More details on the used datasets can be found in Appendix B. We ordered the datasets by increasing numbers of ligands and divided them into two groups. The first group of datasets DS1, DS3, ..., DS19 listed in Table 3.1 together with its corresponding protein target ID was used for preliminary single-view experiments (A) to find appropriate compositions of graph patterns for the subsequent multi-view approaches. The main part of experiments concerned with MKL approaches for affinity prediction (B, C) was performed with another group of datasets DS2, DS4, ..., DS20 which is independent of the one used in part A. The second group of datasets is listed in Table 3.2 together with their respective protein IDs. More details on the precise setting of part A, B, and C can be found in Section 3.4.2 below.

Every ligand molecule was originally provided in SMILES format from which its labelled connected graph structure was calculated as SDF with the chemistry toolbox *Open Babel*[1] (see Section 1.3.4). We modified the SDF graph by introducing the edge label a for edges contained in aromatic ring systems using a Hückel's rule heuristic (compare

[1] available at `openbabel.org`

Section 3.1.1). Based on the modified labels we determined the WL graph labelling up to depth $H = 6$. Preliminary experiments showed that there was no gain in prediction quality using greater WL depths. For every depth h, we calculated the four previously discussed graph patterns *cycles, trees, shortest paths*, and WL *labels* themselves in form of binary and counting feature vectors (refer to Section 3.1.2). The SMILES format also allows for a calculation of standard molecular fingerprints. The formats Maccs, GpiDAPH3 and ECFP6 were applied in the practical experiments of Chapter 3.

We used the SMO-MKL software[2] based on libSVM[3] [Vishwanathan et al., 2010] for both our SVR and ε-MKL experiments. In contrast to ℓ_2-MKL in Section 3.3.1, where the squared loss function is applied to determine the empirical risk, the squared ℓ_p-norm implies a regularisation of the kernel linear coefficients b for ε-MKL. Vishwanathan et al. [2010] showed that their formulation of MKL using the squared ℓ_p-norm is differentiable and hence can be optimised efficiently using *sequential minimal optimisation* (SMO) [Platt, 1999]. We chose $p = 2$. For the ℓ_2-MKL and RLSR experiments we utilised our own implementation of Equation 2.26 and Algorithm 1 [Cortes et al., 2009]. The experimental framework and all figures were generated with *Python* 2.7[4], *Jupyter Notebook* [Kluyver et al., 2016] and *Matplotlib* [Hunter, 2007].

In order to evaluate the quality of the predictor function for regression f, we report the RMSE from Section 2.2 between a vector of predictions and the corresponding vector of true labels. In our experiments we performed a slightly modified k-fold CV scheme (compare Section 2.3.3) for both parameter tuning and training. More precisely, we randomly split a dataset with N instances k-times into a fraction p of training ($n = p \cdot N$) and another fraction $(1 - p)$ of test instances ($m = (1 - p) \cdot N$). In order to achieve an optimal parameter assignment, in each fold we k'-times split the n training instances again into $n' = p \cdot n$ training and $m' = (1 - p) \cdot n$ testing instances for the parameter tuning procedure. Without loss of generality, we assume that n and n' are integers. In our experiments we applied the modified 5-fold CV scheme for the training and tuning procedure and we used a fraction of $p = 0.8$ for the respective training instances. We applied the kernel k_{MP} according to Definition 3.19 together with ℓ_2-MKL and ε-MKL, and hence, MPK-MKL techniques. For this purpose, we used the intersection kernel $k_{\cap,\mathcal{V}}$ from Equation 3.5 and the counting kernel $k_{\sqcap,\mathcal{V}}$ from Equation 3.7 as set kernels in Definition 3.18. We calculated the linear kernel on binary or counting feature vectors for the pattern classes *cycles*, *trees*, *shortest paths*, and *WL labels* based on the WL labelled molecular graphs. For the reason of calculation stability of the used software, we normalised every kernel matrix K initially with its Frobenius matrix norm $\|K\|_2 = (\sum_{i,j=1}^{n} |k_{ij}|^2)^{1/2}$. In previous experiments, during the parameter tuning phase the trade-off parameter ν was always chosen large (independent of the offered range) and all algorithms were almost insensitive to the choice of Λ. Therefore, we fixed $\nu \in \{50.0, 100.0\}$ and $\Lambda = 1.0$. According to chemoinformatics expert knowledge with affinity prediction, we used $\varepsilon = 0.1$ [Balfer and Bajorath, 2015]. For ℓ_2-MKL we fixed $b_0 = (1/M, \ldots, 1/M)$ as initial kernel linear combination coefficients.

[2] available at `research.microsoft.com/en-us/um/people/manik/code/smo-mkl/download.html`

[3] `www.csie.ntu.edu.tw/~cjlin/libsvm/`

[4] `https://www.python.org/`

3.4.2 Results

We provided a multitude of systematical representations for molecules based on their graph structure that can be used for single- and multi-view kernel approaches. These pattern features augment the wide range of standard molecular fingerprints. The general question of this empirical section is whether there is a benefit for affinity prediction from the combination of multiple molecular descriptors via MKL approaches, in particular via MPK-MKL (compare Section 3.3).

A) Preliminary Single-View Experiments

In the first preliminary experiments we extract promising patterns or pattern combinations for the practical task of ligand affinity prediction only using RLSR and SVR. In a sense, these single-view experiments represent the initial part of the actual MPK-MKL procedure we propose in this empirical section and serve to handle the variety of graph patters and kernels.

For the graph pattern classes cycles, trees, shortest paths, and WL labels of different WL depths we used a cumulative and a non-cumulative feature vector variant which we refer to with *cum. pattern* or *pattern* in Figures 3.10 and 3.11. For the cumulative pattern variant, we considered all features based on all WL labelling depths up to some depth H in a concatenated feature vector. For the non-cumulative variant, we only included features of a fixed depth H. We present the results for different WL depths showing the average RMSEs with respect to the datasets with odd numbers DS1, DS3, ..., DS19 in Figures 3.10 and 3.11. We linked the data points with lines in order to better reveal the trend of RMSE values. A differentiated consideration of the particular datasets can be found in Figures 3.6 and 3.7.

The results of the preliminary phase can be found in Figures 3.10 (a) - (d) for counting features or the counting kernel and in Figures 3.11 (a) - (d) for binary features or the intersection kernel. We observe that the qualitative performance trend is very similar both for the application of RLSR and SVR and for the application of the intersection

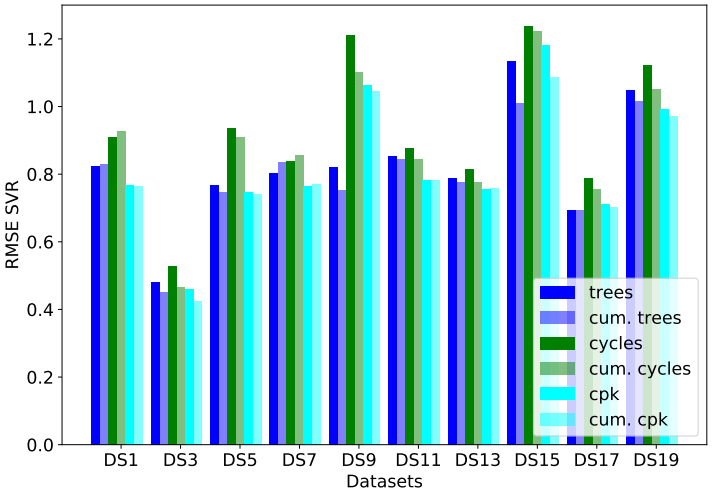
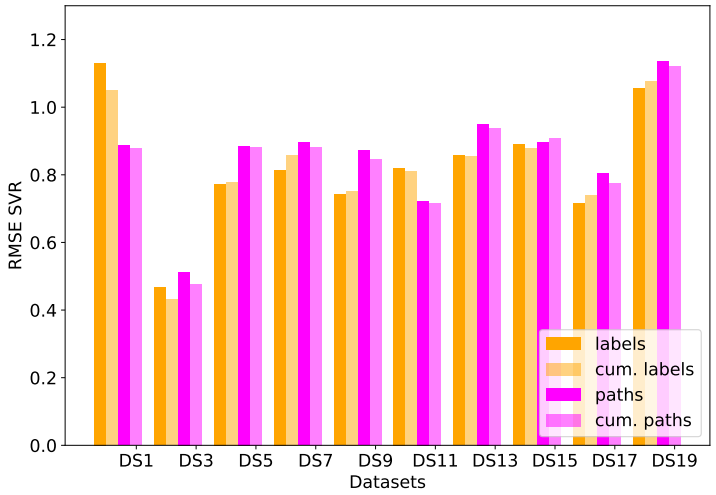


FIGURE 3.6: SVR performance with counting features of cycles and trees

FIGURE 3.7: SVR performance with counting features of labels and paths



and the counting kernel. Obviously, the non-cumulative patterns reach an RMSE minimum for the respective optimal WL depth and the RMSE increases again for greater depths. The RMSEs of the cumulative pattern features appear to converge to the optimal performance with increasing WL depth. However, the best RMSE is very similar for cumulative and non-cumulative patterns for both the binary and the counting feature vectors.

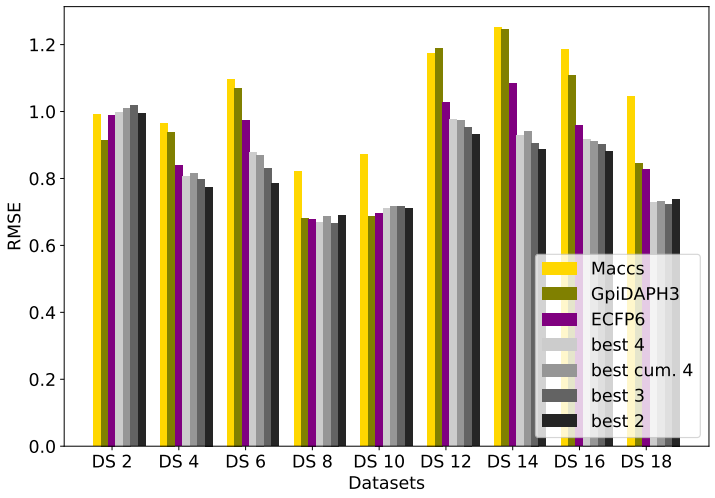
As the information for individual datasets is not apparent in the diagrams of Figures 3.10 and 3.11, for each cumulative and non-cumulative pattern variant we chose the best WL depth and extracted the performance for every dataset with odd number. The result for SVR and the counting kernel is shown in Figure 3.6 for cycles and trees as well as in Figure 3.7 for labels and shortest paths. We observe that both the relative performance of patterns and the comparison between cumulative and non-cumulative feature vectors can be found as a trend for the individual datasets as well.

B) MPK-MKL Experiments

The insights of the preliminary experiments are used for the MPK-MKL experiments in part B. Although, MKL allows for a simultaneous use of multiple kernels or feature vectors, we had to choose for a kernel subset for reasons of complexity of the general problem. We found that different pattern classes show different performance contributions for our considered regression task. Whereas, the application of cumulative and non-cumulative features led to very similar results for affinity prediction provided the respective optimal WL depth is known.

We propose the following experimental scheme for MPK-MKL. Firstly, the most promising combinations of M patterns with respect to WL depth and RMSEs (denoted with *best*) should be extracted in preliminary experiments as it was done in part A. In our practical experiments we considered $M = 2, 3, 4$. Secondly, these *best (cumulative)* M patterns should be used as views on data for ℓ_2-MKL and ε-MKL. The RLSR and SVR baseline approaches include binary and counting feature vector representations and the graph kernels *best SPK*, *best WLK*, and *best CPK*, where *best* indicates that we used

FIGURE 3.8: RLSR (coloured) and ℓ_2-MKL (grey) performance using the intersection kernel



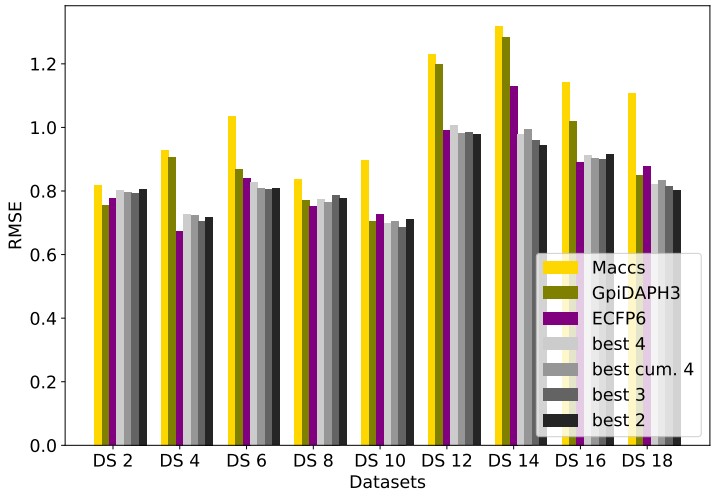
the best cumulative variant with respect to the optimal WL depth. Additionally, we compared the RMSEs of MPK-MKL with the performance of RLSR and SVR using standard molecular fingerprints.

The average RMSEs for datasets DS2, DS4, ..., DS20 are shown in Table 3.3. A differentiated presentation of RMSEs with respect to each dataset can be found in Figures 3.8 and 3.9. For all combinations of squared loss regularisation (RLSR and ℓ_2-MKL) and

View Combination	**Kernel**			
	intersection	counting	intersection	counting
Baselines	RLSR	RLSR	SVR	SVR
Maccs	1.056	1.082	1.078	1.046
GpiDAPH3	0.952	0.987	0.970	0.929
ECFP6	0.896	0.898	0.893	0.853
best SPK	0.873	0.870	0.886	0.830
best WLK	0.886	0.890	0.886	0.833
best CPK	0.989	0.996	0.981	0.950
MPK-MKL	ℓ_2-MKL	ℓ_2-MKL	ε-MKL	ε-MKL
best 4 patterns	0.838	0.851	0.881	0.832
best cum. 4 patterns	0.842	0.846	0.867	0.834
best 3 patterns	0.826	**0.843**	0.861	**0.824**
best 2 patterns	**0.813**	0.855	**0.859**	0.827

TABLE 3.3: Average RMSEs in MPK-MKL experiments (part B)

FIGURE 3.9: SVR (coloured) and ε-MKL (grey) performance using the counting kernel



ε-insensitive regularisation (SVR and ε-MKL) as well as the intersection and counting kernel, the MPK-MKL approaches with 2, 3, and 4 best pattern combinations outperform all baselines with standard graph kernels and molecular fingerprints. The respective MPK-MKL approaches with 2 or 3 best pattern combinations have the smallest RMSEs of the presented MKL results. Binary features led to slightly better results for ℓ_2-MKL (see Figure 3.8), whereas counting features are favourable for ε-MKL and the considered regression task (see Figure 3.9).

C) MKL Experiments with Standard Molecular Fingerprints

The performance of MKL for affinity prediction in combination with standard molecular fingerprints was investigated as well in the third part of the empirical evaluation. To this aim, we picked the three different standard fingerprints Maccs, GpiDAPH3, and ECFP6, such that each of them belongs to another type of binary molecular presenters (compare Section 1.3.4). As it represents the state-of-the-art approach for affinity prediction with standard molecular fingerprints, we opposed SVR to the SVR-type ε-MKL approach (see Section 3.3.2). We did not perform preliminary experiments here, as the molecular representation was given by the fingerprint format. More precisely, we utilised each combination of two as well as the set of all three fingerprints. For the experiments with standard molecular fingerprints, again we considered the datasets DS2, DS4, ..., DS20 in order to compare the results with the ones for MPK-MKL in part B above. As baselines for ε-MKL we included the single-view approaches *SVR (view)*, where *view* denotes the utilised fingerprint, and the multi-view approach *SVR (concat)*, which uses a concatenation of the respective fingerprints or views (compare also the empirical section of Chapter 4). Furthermore, we applied the linear kernel for the vectorial representations of molecules. We averaged the RMSEs over all datasets in order to present the results in Table 3.4 and to compare the results with the ones of Table 3.3.

We realise that the performance of SVR (concat) lies in the range of the RMSEs of the single-view SVR (view) approaches. For the optimal view choice SVR (concat) shows comparable results to SVR (view). In contrast, ε-MKL beats all baselines for all

Table 3.4: Average RMSEs in ε-MKL experiments with standard molecular fingerprints (part C)

View Combinations	**Methods**				
view 1/view 2(/view 3)	SVR (view 1)	SVR (view 2)	SVR (view 3)	SVR (concat)	ε-MKL
Maccs/ECFP6	1.081	0.884	-	0.921	**0.848**
Maccs/GpiDAPH3	1.050	0.921	-	0.918	**0.853**
GpiDAPH3/ECFP6	0.856	0.916	-	0.855	**0.768**
Maccs/GpiDAPH3/ECFP6	0.859	0.919	1.058	0.881	**0.824**

combinations of two or three standard molecular fingerprints with respect to averaged RMSEs.

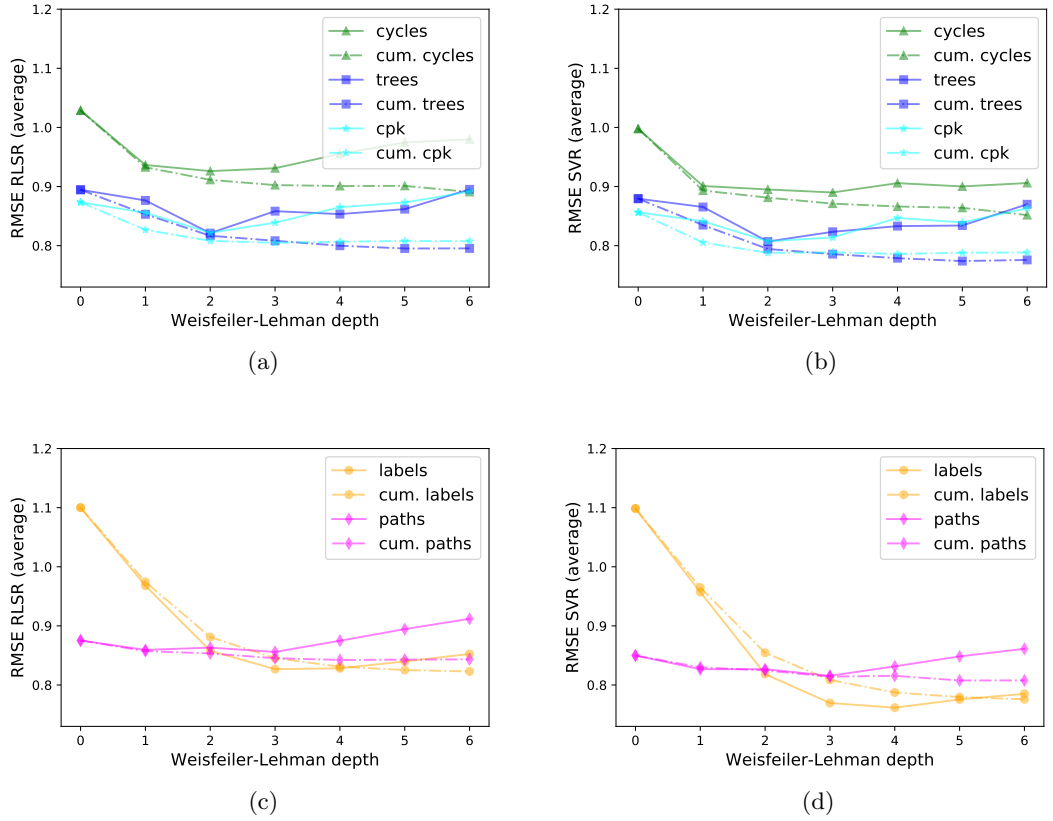
3.4.3 Discussion

The design of the standard molecular fingerprints that we used for our experiments is based on the 2D graph structure of the respective molecules. This is also the case for the presented graph pattern classes cycles, trees, shortest paths, and WL labels. However, in contrast to the graph pattern classes, the characteristic feature set of standard molecular fingerprints is in principle fixed. To gain more flexibility towards the regression task at hand, we proposed MPK-MKL as a systematic application of different graph pattern features within supervised multiple-kernel learning algorithms. More precisely, we compared ℓ_2-MKL and ε-MKL to single- and multi-view baselines using various graph pattern feature representations, standard graph kernels, and molecular fingerprints. In conclusion, MPK-MKL is a successful technique for the learning task of affinity prediction. We showed that the ligand affinity prediction performance can be improved compared to the state-of-the-art technique SVR using standard molecular fingerprints.

Obviously, affinity prediction profits from the simultaneous and systematic inclusion of important graph pattern classes into the learning process. Both investigated MKL algorithms outperform the respective single- and multi-view baselines (see Table 3.3 and Figures 3.8 and 3.9). The intersection kernel led to smaller RMSEs for both RLSR and SVR in the preliminary experiments. This was also the case for ℓ_2-MKL in the MPK-MKL experiments. In contrast, the counting kernel applied with ε-MKL outperformed the intersection kernel for MPK-MKL. In theory, the WL depths for the provided binary and counting feature vectors can become infinite. Nevertheless, previous experiments on our regression task have shown that all pattern classes had reached their performance optimum with RLSR and SVR at a WL depth of 6 or smaller. As the optimal values were similar, we preferred non-cumulative feature vector variants for the MPK-MKL experiments. In preliminary experiments, WL labels generally show the smallest RMSEs which underlines the importance of this pattern class for the prediction task at hand. For that reason, we propose to definitely include WL label patterns in view combinations of MPK-MKL approaches. We point to the fact that absolute RMSEs for preliminary, MPK-MKL, and MKL experiments with standard molecular fingerprints cannot be compared directly as we performed different runs and used independent datasets.

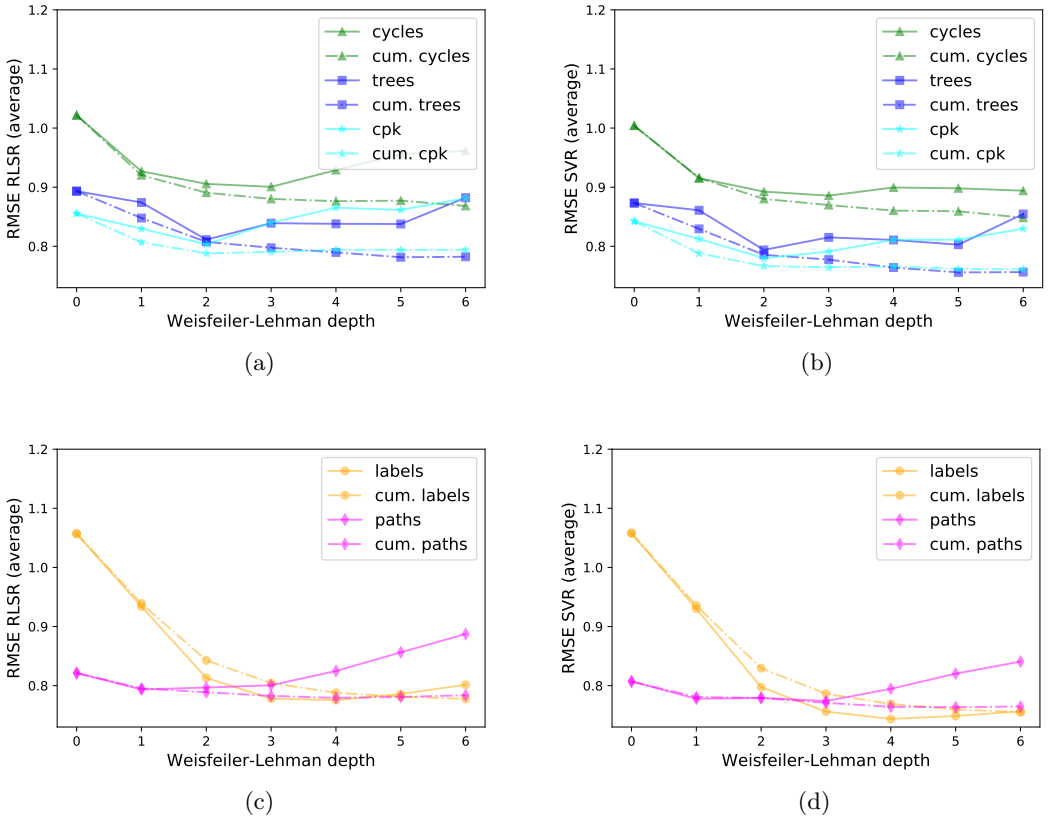
A drawback of the proposed MPK-MKL procedure is the big effort to generate and systematically choose feature representations as a preparation step for the actual MKL

FIGURE 3.10: Average RMSEs of RLSR (left) and SVR (right) in preliminary experiments (part A) based on the counting kernel



experiments. Because of the more difficult optimisation problem of ℓ_2-MKL and ε-MKL in comparison to RLSR and SVR, the running time of these algorithms is high. The actual choice of fingerprints is still an issue in this learning protocol and additionally the best WL depth h for each pattern remains an empirical issue. For reasons of complexity, we did not consider combinations of binary and counting or cumulative and non-cumulative feature vectors within one MKL algorithm. The application of different standard molecular fingerprints together with MKL is still expensive in terms of running time, but the effort reduces to choose a set of standard fingerprints, preferably from different fingerprint types. The results in Table 3.4 show that MKL in combination with standard molecular fingerprints competes with MPK-MKL for affinity prediction if appropriate fingerprint formats are utilised. It is up to the precise dataset, which approach is preferable and whether the improvement in prediction legitimates the increase in computing time. In summary, the application of MKL is a very promising technique in the research field of chemoinformatics, particularly for ligand affinity prediction. The approach of MPK-MKL includes a very comprehensive set of graph pattern features for prediction together with its systematic choice as inherent part. In addition to the improved prediction of affinity values for molecular compounds, MPK-MKL reduces the efforts for the most appropriate representation by a systematic analysis of graph pattern performances.

FIGURE 3.11: Average RMSEs of RLSR (left) and SVR (right) in preliminary experiments (part A) based on the intersection kernel



Chapter 4

Co-Regularisation

In the previous chapter we investigated a supervised inductive approach that used labelled training data and multiple views on data out of a big number of potential representations. We showed for the learning task of affinity prediction that we can indeed take profit from a model which is based on a linear combination of predictor functions related to different views. However, the described supervised approaches ignore the fact that the determination of ligand affinities is expensive as they have to be determined in a time- and cost-consuming procedure in laboratories.

In the present chapter, we take a step forward to a more realistic scenario concerning the considered learning task. More precisely, a typical ligand affinity prediction setting is characterised by only few ligands with affinity annotation as the source of labelled data. In contrast, a large amount of small molecular compounds is available, for which the affinity information towards a considered protein is unknown. However, these ligand candidates can be employed as unlabelled data easily, since no efforts have to be done for their labelling. With respect to the representation of data, molecular fingerprints are available which describe physico-chemical or structural information of the considered molecule in vectorial format. A variety of such publicly available or commercial molecular representations exist [Bender et al., 2009]. Each fingerprint captures a particular set of information and it is not clear a priori which fingerprint is the most appropriate for the learning task at hand. In Chapter 3, this problem is tackled by using multiple data representations simultaneously in an MKL approach. This approach utilises a kernel linear combination for the final predictor and outperformed single-view baselines in our practical experiments on affinity prediction. Different data representations can also be used to include unlabelled instances in the training process which we investigate in the present chapter. The intention behind the *co-regularisation* approach from semi-supervised learning is to compensate the lack of a satisfactory number of labelled examples by the usage of many unlabelled instances from the respective feature domain. Although we focus on affinity prediction, the presented approaches below are applicable for all learning tasks with

- real-valued label,
- few labelled examples, but
- many unlabelled instances,

- multiple data representations with appropriate measure to assess the similarity between instances (kernel function).

Semi-supervised learning has already been applied in the field of ligand prediction using labelled and unlabelled data. Ning et al. [2009] classified molecular compounds by taking into account additional information of related protein targets. Kondratovich et al. [2013] applied a *transductive support vector machine* model [Joachims, 1999]. However, the combination of multiple views and unlabelled data has not yet been used in the context of ligand affinity prediction. The following real-world examples, including affinity prediction (drug discovery) and another problem from the medical domain, show the practical relevance of the considered learning scenario and of the machine learning algorithms to solve them.

Example 4.1. ***(Drug discovery)*** *Nowadays, ten thousands of human proteins are already known, not to mention the number of all proteins in biological organisms. Therefore, it is not a contradiction that, given a particular protein, the number of labelled compounds for that protein is in general very small. In contrast, the few labelled compounds face a large amount of synthesizable small molecules without labels, the potential ligand candidates. More precisely, we know the structure of a lot of small molecular compounds and can represent them via different molecular fingerprint formats, but we do not have any binding information for them with respect to the considered protein. Affinity prediction for small molecules such that only few labelled training examples and many unlabelled instances are available is in the focus of the present chapter. The molecules with high predicted affinity values can be used as promising drug candidates in order to make drug discovery in pharmaceutical research more efficient.*

Example 4.2. ***(Body height prediction)*** *Several diseases, such as gigantism or microsomia, come along with an abnormal growth of the body and of extremities in particular. For the diagnosis of children it would be helpful to predict the final body height from the patient's related data, as the growth process can be influenced via hormones or other drugs. The diagnosis should occur as early as possible as the therapy becomes unfeasible once the epiphyseal plates are closed. Patient information records include, e.g., blood tests, radiographs, body height curves, or other indicators of the body's physical condition and development. Unlabelled medical data records of children exist in abundance. Labelled datasets for body height prediction are difficult to obtain as the final body height is actually only available in the future.*

Affinity prediction and comparable applications suffer from the problems arising from little label information and the need to choose the most appropriate view for learning. To overcome this difficulties, the semi-supervised and multi-view approach of co-regularisation matches the outcome of view predictors for unlabelled instances. This procedure leads to a regularisation of the view predictors as they are chosen out of the intersection set of predictors that coincide on unlabelled instances. More precisely, multiple predictor functions are learned such that each of them is related to a particular view on data. To this aim, both the regularised empirical risk of every single predictor and the pairwise distance between the outcomes of different view predictors for unlabelled instances are minimised. The final predictor is supposed to be the average of the simultaneously learned view predictor function.

In comparison to supervised approaches, semi-supervised algorithms are beneficial in the case of few labelled examples [Chapelle et al., 2006]. A semi-supervised SVR using only

a single view on data has been investigated by Wang et al. [2010a] and a co-regularised variant of RLSR named CoRLSR was presented by Brefeld et al. [2006]. We provide the SVR optimisation with a co-regularisation term and obtain *co-regularised support vector regression* (CoSVR) [Ullrich et al., 2016a, 2017]. For the co-regularisation term we investigate the properties and empirical performance of the squared loss function (ℓ_2-CoSVR) and the ε-insensitive loss function (ε-CoSVR). Because of the longer running time of the proposed base CoSVR algorithms compared to SVR, we define variants with a reduced number of variables. Based on a result of Sindhwani and Rosenberg [2008] we deduce a CoSVR transformation with single-view SVR properties in terms of optimisation variables and, thus, time complexity. Moreover, we prove upper bounds for the Rademacher complexity of co-regularised hypothesis spaces, which is useful to restrict the capacity of the considered function class to fit random data.

The present chapter is based on the publications [Ullrich et al., 2016a] and [Ullrich et al., 2017]. It is structured as follows. We start with the definition of a semi-supervised variant of the RRM principle which serves as initial point for the co-regularised algorithms. In Section 4.2 we present CoRLSR of Brefeld et al. [2006]. Subsequently, we introduce CoSVR and examine two loss functions for the actual co-regularisation term in Section 4.3. In addition to variants of base CoSVR with less optimisation variables, we also derive a transformation into the single-view method Σ-CoSVR in Section 4.3.3. In the following Section 4.3.5 we prove bounds for the Rademacher complexity. The practical benefit of the presented co-regularisation approaches for ligand affinity prediction will be shown in the concluding empirical analysis in Section 4.4.

4.1 Co-Regularisation for Regression

We consider a space of instances $\mathcal{X}$ and multiple views $v = 1, \ldots, M$ on data. We intend to learn different predictor functions $f_v : \Phi_v(\mathcal{X}) \to \mathcal{Y}$, each corresponding to a view v. Every view predictor f_v is intended to have a small training error with respect to n examples with known labels and a loss function ℓ^L. We introduced the concept of a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ in Definition 2.2 as a non-negative function with $\ell(y, y) = 0$ for all $y \in \mathcal{Y}$. Typically, a loss function is additionally required to be convex for the solution strategy of the resulting optimsation problem. The approach of co-regularisation is defined as a multi-view RRM problem such that additionally the difference between pairwise view predictions over m unlabelled examples measured with another loss function ℓ^U is minimal. In the following, an upper index L will refer to the empirical risk for labelled examples and the upper U refers to the error term with respect to unlabelled instances. The following definition generalises the concept of RRM and comprises the optimisation problems considered [Sindhwani and Rosenberg, 2008, Rosenberg and Bartlett, 2007, Brefeld et al., 2006].

Definition 4.1 (CoRRM)**.** Let ℓ^L and ℓ^U be loss functions for regression and $\mathcal{H}_v$ be appropriate function spaces. We consider labelled examples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ and unlabelled points $z_1, \ldots, z_m \in \mathcal{X}$. The *co-regularised risk minimisation* (CoRRM)

principle is to solve the optimisation

$$\min_{f_v \in \mathcal{H}_v} \sum_{v=1}^{M} \left(\nu_v \|f_v\|_{\mathcal{H}_v}^2 + \sum_{i=1}^{n} \ell^L \left(y_i, f_v(x_i) \right) \right) + \lambda \sum_{u,v=1}^{M} \sum_{j=1}^{m} \ell^U \left(f_u(z_j), f_v(z_j) \right), \tag{4.1}$$

where $\nu_v, \lambda > 0$ are the hyperparameters. The predictor $f : \mathcal{X} \to \mathcal{Y}$ for the regression problem at hand is supposed to be the average

$$f = \frac{1}{M} \sum_{v=1}^{M} f_v$$

of the view predictors $f_1, \ldots, f_M$.

The hyperparameters ν_v and λ play a slightly different role compared to ν and Λ in the MKL objectives in Chapter 3. However, all of them serve as trade-off parameters between parts of objective functions. The CoRRM approach in Equation 4.1 includes two types of jointly minimised error terms involving the view predictor functions $f_1, \ldots, f_M$. Firstly, all view predictors f_v are supposed to be good predictor functions themselves. More precisely, one aims at a small *labelled error* term

$$\sum_{v=1}^{M} \sum_{i=1}^{n} \ell^L(y_i, f_v(x_i)).$$

Due to the lack of labels for unlabelled instances, secondly, the CoRRM optimisation demands pairs of view predictors to coincide for unlabelled instances as good as possible. Although the true label of unlabelled instances is unknown, this assumption leads to an additional regularisation of the solution functions and implies a small *unlabelled error* term

$$\sum_{u,v=1}^{M} \sum_{j=1}^{m} \ell^U \left(f_u(z_j), f_v(z_j) \right).$$

The unlabelled error is equipped with hyperparameter $\lambda > 0$ to enable a trade-off between the different terms to minimise in Equation 4.1. Although there are no labels for unlabelled instances available, we use the name *unlabelled error* in order to express that differences between view predictions should measure the quality of the predictor functions. The norm terms $\|f_v\|_{\mathcal{H}_v}^2$, $v = 1, \ldots, M$, prevent overfitting. Analogous to the single-view case in Chapter 2 and the MKL scenario in Chapter 3, we prove a representation of the CoRRM solution functions in the following lemma.

Lemma 4.2. *Let $\mathcal{H}_1, \ldots, \mathcal{H}_M$ be RKHSs of the kernel functions $k_1, \ldots, k_M$. Furthermore, let $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ be labelled training examples and $z_1, \ldots, z_m \in \mathcal{X}$ be unlabelled instances. With ℓ^L and ℓ^U we denote two loss functions and $\nu_v, \lambda > 0$ are hyperparameters. The solutions $f_v \in \mathcal{H}_v$, $v = 1, \ldots, M$, of the CoRRM optimisation*

$$\min_{f_v \in \mathcal{H}_v} \sum_{v=1}^{M} \left(\sum_{i=1}^{n} \ell^L(y_i, f_v(x_i)) + \nu_v \|f_v\|_{\mathcal{H}_v}^2 \right) + \lambda \sum_{u,v=1}^{M} \sum_{j=1}^{m} \ell^U(f_u(z_j), f_v(z_j)) \tag{4.2}$$

from Definition 4.1 have a representation in form of

$$f_v(\cdot) = \sum_{i=1}^{n} \pi_{vi} k_v(x_i, \cdot) + \sum_{j=1}^{m} \pi_{v(n+j)} k_v(z_j, \cdot), \tag{4.3}$$

where $v = 1, \ldots, M$ *and* $\pi_{v1}, \ldots, \pi_{v(n+m)} \in \mathbb{R}$ *are real-valued coefficients.*

Proof. For every $v = 1, \ldots, M$ we consider the space

$$S_v = \operatorname{span}\{\Phi_v(x_i), i = 1, \ldots, n+m\}$$

and its orthogonal complement $S_v^\perp$. Analogous to the proof of the single-view case in Theorem 2.21, every view predictor $f_v \in \mathcal{H}_v$ can be written as $f_v = f_v^0 + f_v^1$, where $f_v^0 \in S_v$ and $f_v^1 \in S_v^\perp$. As shown in Equation 2.23, $f_v^1(x_i) = 0$ holds true for every view $v = 1, \ldots, M$ and every $i = 1, \ldots, n+m$. Consequently, both the empirical risk and the unlabelled loss term in Equation 4.2 do not depend on f_v^1. The norm terms in Equation 4.2 can be written as

$$\|f_v\|_{\mathcal{H}_v}^2 = \|f_v^0\|_{\mathcal{H}_v}^2 + \|f_v^1\|_{\mathcal{H}_v}^2$$

because of the orthogonality property of f_v^0 and f_v^1. The norm $\|f_v\|_{\mathcal{H}_v}^2$ is minimised, if f_v^1 is the zero function in $S_v^\perp$ which finishes the proof. □

Lemma 4.2 and its proof are analogues of Theorem 2.21 and the corresponding proof in a co-regularisation scenario. Similar to the proceeding in Chapter 3, we will consider co-regularisation in a least squares and support vector regression setting. Again, the choice of the loss functions ℓ^L and ℓ^U in Equation 4.1 specifies the actual optimisation problem to solve in the CoRRM optimisation. The case where $\ell^L = \ell^U$ equals the squared loss is already known as *co-regularised least squares regression* (CoRLSR) and was introduced by Brefeld et al. [2006]. It will be reviewed in Section 4.2. As a novel approach we will present *co-regularised support vector regression* (CoSVR) and its variants and properties in Section 4.3 below. In this context, we choose ℓ^L to be the ε-insensitive loss and ℓ^U to be an arbitrary loss function. However, we thoroughly investigate the cases squared loss function and ε-insensitive loss function for ℓ^U.

We will use the term *co-regularisation* both for the approach in CoRRM and the actual unlabelled error term. We point out that the view predictors f_v are simultaneously derived from the CoRRM minimisation in Equation 4.1. The view predictors are in general not equal to the single-view predictors f_v, that are calculated independently with single-view regression algorithms, for example, with RLSR or SVR according to Equations 2.25 or 2.27. Other than in Chapter 3, the Gram matrix K of a kernel function k in the present chapter comprises kernel values over labelled and unlabelled examples

$$K = (k(x_i, x_j))_{i,j=1}^{n+m} \in \mathbb{R}^{(n+m)\times(n+m)},$$

where the m unlabelled instances $x_{n+1}, \ldots, x_{n+m} \in \mathcal{X}$ are also denoted with $z_1, \ldots, z_m$ (compare Section 2.1). We will consider the decomposition of the Gram matrix

$$K = \begin{pmatrix} L \\ U \end{pmatrix} \tag{4.4}$$

into an upper submatrix $L \in \mathbb{R}^{n\times(n+m)}$ and a lower submatrix $U \in \mathbb{R}^{m\times(n+m)}$, respectively.

4.2 Co-Regularised Least Squares Regression

The implementation of CoRRM from Equation 4.1 with least squares labelled loss function *co-regularised least squares regression* (CoRLSR) [Brefeld et al., 2006] will be the first of the considered co-regularised algorithms. It is defined as follows.

Definition 4.3 (CoRLSR)**.** [Brefeld et al., 2006] Suppose we have kernel functions $k_1, \ldots, k_M$ and associated RKHSs $\mathcal{H}_1, \ldots, \mathcal{H}_M$. Furthermore, let $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ be labelled training examples, $z_1, \ldots, z_m \in \mathcal{X}$ be unlabelled instances, and $\nu_v, \lambda > 0$, $v = 1, \ldots, M$, be regularisation parameters. The minimisation problem

$$\min_{f_v \in \mathcal{H}_v} \sum_{v=1}^{M} \left(\nu_v \|f_v\|_{\mathcal{H}_v}^2 + \sum_{i=1}^{n} \|y_i - f_v(x_i)\|^2 \right) + \lambda \sum_{u,v=1}^{M} \sum_{j=1}^{m} \|f_u(z_j) - f_v(z_j)\|^2 \tag{4.5}$$

is called *co-regularised least squares regression* (CoRLSR).

The desired predictor $f : \mathcal{X} \to \mathcal{Y}$ for regression will again be the average $f = \frac{1}{M}\sum_{v=1}^{M} f_v$ of the optimisation results $f_1, \ldots, f_M$. Brefeld et al. [2006] derived a closed formula for the solution of CoRLSR shown in the subsequent lemma.

Lemma 4.4. *[Brefeld et al., 2006] Let the Gram matrices K_v for view $v \in \{1, \ldots, M\}$ have a decomposition into an upper part L_v and a lower part U_v according to Equation 4.4. We consider the matrix*

$$G_v = L_v^T L_v + \nu_v K_v + 2\lambda(M-1) U_v^T U_v$$

for $\nu_v, \lambda > 0$, and the vector $Y = (y_1, \ldots, y_n)^T$ of training labels. The CoRLSR problem from Equation 4.5 can be solved via

$$\pi = \begin{pmatrix} G_1 & -2\lambda U_1^T U_2 & \cdots & -2\lambda U_1^T U_M \\ -2\lambda U_2^T U_1 & G_2 & \cdots & -2\lambda U_2^T U_M \\ \vdots & \vdots & \ddots & \vdots \\ -2\lambda U_M^T U_1 & -2\lambda U_M^T U_2 & \cdots & G_M \end{pmatrix}^{-1} \begin{pmatrix} L_1^T Y \\ L_2^T Y \\ \vdots \\ L_M^T Y \end{pmatrix},$$

where

$$\pi = \begin{pmatrix} \pi_1 \\ \vdots \\ \pi_M \end{pmatrix} \in \mathbb{R}^{M(n+m)} \quad \textit{and} \quad \pi_v = \begin{pmatrix} \pi_{v1} \\ \vdots \\ \pi_{v(n+m)} \end{pmatrix} \in \mathbb{R}^{n+m} \tag{4.6}$$

are the kernel expansion coefficient vectors of the view predictors f_v corresponding to Equation 4.3.

We point to the fact that the indices $v1, \ldots, v(n+m)$ in Equation 4.6 are double indices and not products. The following proof of Brefeld et al. [2006] applies Lemma 4.2 which is basically a multi-view representer theorem. To be more precise, every view predictor f_v has a representation as kernel linear combination of k_v.

Proof. As a consequence of Lemma 4.2 the optimisation in Equation 4.5 admits a representation of the solution functions as kernel expansions centered at labelled and unlabelled examples. For this reason, we reformulate the problem in a parameterised way with kernel functions k_v and parameters π_v in place of functions f_v

$$\min_{\pi_v \in \mathbb{R}^{n+m}} \mathcal{Q}(\pi_1, \ldots, \pi_M) = \min_{\pi_v \in \mathbb{R}^{n+m}} \sum_{v=1}^{M} \left(\nu_v \pi_v K_v \pi_v + \|Y - L_v \pi_v\|^2 \right) + \lambda \sum_{u,v=1}^{M} \sum_{j=1}^{m} \|U_u \pi_u - U_v \pi_v\|^2.$$

The partial derivate of the objective $\mathcal{O}(\pi_1, \ldots, \pi_M)$ with respect to π_v is

$$\frac{\partial \mathcal{Q}(\pi_1, \ldots, \pi_M)}{\partial \pi_v} = 2G_v \pi_v - 2L_v^T Y - 4\lambda \sum_{u=1}^{M, u \neq v} U_v^T U_u \pi_u$$

with the definition of G_v from above. Setting all derivatives equal to $\mathbf{0}_{n+m}$ leads to the desired result. □

In the next section we introduce the support vector regression variant of CoRRM.

4.3 Co-Regularised Support Vector Regression

In the previous section for CoRLSR, the squared loss ℓ_2 was used for both training error and the co-regularisation term to obtain a multi-view analogue of RLSR. The SVR algorithm with ε-insensitive loss has a very good generalisation capability and at the same time shows very good prediction performance [Awad and Khanna, 2015]. Aside from that, SVR is the state-of-the-art method applied in ligand affinity prediction (compare Section 1.3.5). Therefore, we define co-regularised support vector regression (CoSVR) as the CoRRM optimisation from Equation 4.1 such that the ε-insensitive loss function is used for the empirical risk. In contrast to CoRLSR, the labelled loss function for CoSVR in its base version remains arbitrary according to Definition 2.2. We will investigate the cases of squared and ε-insensitive labelled loss function extensively below.

4.3.1 Base Algorithm

We start with the definition of the base algorithm as the CoRRM problem from Equation 4.1 with ε-insensitive loss function in the labelled error term and present two special cases.

Definition 4.5 (CoSVR, ℓ_2-CoSVR, ε-CoSVR)**.** [Ullrich et al., 2017] For $v = 1, \ldots, M$ let $\mathcal{H}_v$ be an RKHS, ℓ^U be an arbitrary loss function, and $\varepsilon^L, \nu_v, \lambda > 0$ be hyperparameters. The optimisation problem in Equation 4.1 such that ℓ^L is the ε-insensitive loss with $\varepsilon = \varepsilon^L$ is called *co-regularised support vector regression* (CoSVR).
(i) Co-regularised support vector regression with $\ell^U = \ell_2$

$$\begin{aligned}\min_{f_v \in \mathcal{H}_v} \sum_{v=1}^{M} \left(\frac{\nu_v}{2} \|f_v\|_{\mathcal{H}_v}^2 + \sum_{i=1}^{n} \max\{|y_i - f_v(x_i)| - \varepsilon^L, 0\} \right) \\ + \lambda \sum_{u,v=1}^{M} \sum_{j=1}^{m} \|f_u(z_j) - f_v(z_j)\|^2,\end{aligned} \tag{4.7}$$

is denoted *ℓ_2-co-regularised support vector regression* (ℓ_2-CoSVR).
(ii) Co-regularised support vector regression where ℓ^U is the ε-insensitive loss

$$\begin{aligned}\min_{f_v \in \mathcal{H}_v} \sum_{v=1}^{M} \left(\frac{\nu_v}{2} \|f_v\|_{\mathcal{H}_v}^2 + \sum_{i=1}^{n} \max\{|y_i - f_v(x_i)| - \varepsilon^L, 0\} \right) \\ + \lambda \sum_{u,v=1}^{M} \sum_{j=1}^{m} \max\{|f_u(z_j) - f_v(z_j)| - \varepsilon^U, 0\}\end{aligned} \tag{4.8}$$

is called *ε-co-regularised support vector regression* (ε-CoSVR).

In comparison to CoRRM from Equation 4.1, we introduced a factor of $1/2$ in the norm term for arithmetical reasons. The sums $\sum_{u,v=1}^{M}$ are actually always of the kind $\sum_{u,v=1}^{M,u\neq v}$, as the respective summands for $u = v$ are equal to zero. In the following, we present solutions for ℓ_2-CoSVR and ε-CoSVR.

Lemma 4.6. *[Ullrich et al., 2017] Let $\nu_v, \lambda, \varepsilon^L, \varepsilon^U > 0$. We use the notation introduced above. In particular, $\pi_v \in \mathbb{R}^{n+m}$ denote the kernel expansion coefficients of the view predictors f_v from Equation 4.3, whereas $\alpha_v, \hat{\alpha}_v \in \mathbb{R}^n$ and $\gamma_{uv} \in \mathbb{R}^m$ are dual variables.*

(i) *The dual optimisation problem of ℓ_2-CoSVR is*

$$\begin{aligned}\max_{\alpha_v, \hat{\alpha}_v \in \mathbb{R}^n, \gamma_{uv} \in \mathbb{R}^m} \sum_{v=1}^{M} \Bigg(-\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha_v - \hat{\alpha}_v)^T Y \\ -\varepsilon^L (\alpha_v + \hat{\alpha}_v)^T \mathbf{1}_n - \frac{1}{4\lambda} \sum_{u=1}^{M} \gamma_{uv}^T \gamma_{uv} \Bigg) \\ \text{s.t.} \left\{ \begin{array}{l} \mathbf{0}_n \leq \alpha_v, \hat{\alpha}_v \leq \mathbf{1}_n \\ \gamma_{uv} = \frac{2\lambda}{\nu_u} U_u \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_u - \frac{2\lambda}{\nu_v} U_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v \end{array} \right\}_{(u,v) \in [\![M]\!]^2},\end{aligned}$$

where

$$\begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v = \begin{pmatrix} \alpha_v - \hat{\alpha}_v \\ \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu}) \end{pmatrix}$$

and $\pi_v = \frac{1}{\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$ holds true.

(ii) The dual optimisation problem of ε-CoSVR equals

$$\max_{\alpha_v,\hat{\alpha}_v\in\mathbb{R}^n,\gamma_{uv}\in\mathbb{R}^m} \sum_{v=1}^{M} \left(-\frac{1}{2\nu_v} \begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v^T K_v \begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v + (\alpha_v - \hat{\alpha}_v)^T Y \right.$$
$$\left. -(\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n - \sum_{u=1}^{m} \gamma_{uv}^T \varepsilon^U \mathbf{1}_m \right)$$
$$\text{s. t.} \left\{ \begin{array}{l} \mathbf{0}_n \leq \alpha_v, \hat{\alpha}_v \leq \mathbf{1}_n \\ \mathbf{0}_m \leq \gamma_{uv} \leq \lambda \mathbf{1}_m \end{array} \right\}_{(u,v)\in[\![M]\!]^2},$$

where

$$\begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v = \begin{pmatrix} \alpha_v - \hat{\alpha}_v \\ \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu}) \end{pmatrix}$$

and $\pi_v = \frac{1}{\nu_v}\begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v$ are the kernel expansion coefficients from Equation 4.3.

Proof. The proof can be found in Appendix A. □

We will refer to ℓ_2-CoSVR and ε-CoSVR as the base CoSVR versions. An overview of the used variable identifiers and their purpose within the optimisation formulation and solution can be found in Table 4.1. If variables are used for the same purpose in different results or proofs they will get the same identifier if possible. The variable name $x_{(v)}$ means that the variable x exists with or without index v. The symbol

$$\begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v \in \mathbb{R}^{n+m}$$

represents a view-dependent vector composed from α- and γ-variables in different ways. The precise formulas can be found in Lemmas 4.6, 4.8, and 4.10. The symbol reveals the analogies between the result from Lemma 4.6 and the related ones in Lemmas 4.8 and 4.10. In the discussed problems above also

$$\alpha_v \cdot \hat{\alpha}_v = 0 \quad \text{and} \quad \gamma_{uv} = -\gamma_{vu} \quad \text{(for } \ell_2\text{-CoSVR)} \tag{4.9}$$

and

$$\alpha_v \cdot \hat{\alpha}_v = 0 \quad \text{and} \quad \gamma_{uv} \cdot \gamma_{vu} = 0 \quad \text{(for } \varepsilon\text{-CoSVR)} \tag{4.10}$$

holds true for the respective dual variables according to the KKT conditions in Theorem 2.14. Smola and Schölkopf [2004] commented on how SVR solvers incorporate comparable conditions in the single-view case. By definition, also $\zeta_{vv} = \mathbf{0}_m$ is valid for all $v = 1, \ldots, M$.

4.3.2 Reduction of Variable Numbers

According to Lemma 4.6 the base CoSVR algorithm can be solved as a QP with linear equality and inequality constraints. A solver for this convex optimisation problem has to handle $\mathcal{O}(Mn + M^2m)$ variables and corresponding constraints. In real-world applications, the number m of available unlabelled instances is by magnitudes greater than the

TABLE 4.1: Overview of variable notation in semi-supervised approaches

Variable Type	**Variable Identifier**	**Risk Term**
Slack variables	$\xi_{(v)} = (\xi_{(v)1}, \cdots, \xi_{(v)n})^T$	labelled error
	$\hat{\xi}_{(v)} = (\hat{\xi}_{(v)1}, \cdots, \hat{\xi}_{(v)n})^T$	
	$\zeta_{(u)v} = (\zeta_{(u)v1}, \cdots, \zeta_{(u)vm})^T$	unlabelled error
Kernel expansion variables	$\pi_{(v)} = (\pi_{(v)1}, \cdots, \pi_{(v)(n+m)})^T$	labelled/unlabelled error
Dual variables	$\alpha_{(v)} = (\alpha_{(v)1}, \cdots, \alpha_{(v)n})^T$	labelled error
	$\hat{\alpha}_{(v)} = (\hat{\alpha}_{(v)1}, \cdots, \hat{\alpha}_{(v)n})^T$	
	$\beta_{(v)} = (\beta_{(v)1}, \cdots, \beta_{(v)n})^T$	
	$\hat{\beta}_{(v)} = (\hat{\beta}_{(v)1}, \cdots, \hat{\beta}_{(v)n})^T$	
	$\gamma_{(u)v} = (\gamma_{(u)v1}, \cdots, \gamma_{(u)vm})^T$	unlabelled error
	$\delta_{(u)v} = (\delta_{(u)v1}, \cdots, \delta_{(u)vm})^T$	

number n of labelled instances. For example, assume we only had 2 views, 5 labelled, and 50 unlabelled examples. This would already result in 220 variables (see Table 4.2 in Section 4.3.4 for the precise number of variables). In comparison, a single-view SVR solver would have to take only $2n = 10$ variables into account. In addition, $M+3$ hyperparameters ν_v, λ, ε^L, and ε^U have to be tuned during the training phase for ε-CoSVR ($M + 2$ hyperparameters ν_v, λ, ε^L for ℓ_2-CoSVR). Whereas, only 2 hyperparameters ν and ε have to be optimised for single-view SVR.

In order to lower the negative effect on the running time, we present CoSVR variants with a reduced number of variables. To this aim, we decreased the number of variables by weaker demands on the view predictors in the error terms. We denote the variant with modification in the labelled error with CoSVR$^{\text{mod}}$ and in the unlabelled error with CoSVR$_{\text{mod}}$.

Modification of the Empirical Risk

The objective of base CoSVR becomes smaller if the empirical risk decreases with respect to labelled examples for each view predictor individually. The CoSVR$^{\text{mod}}$ approach applies the average prediction

$$f^{\text{avg}} = \frac{1}{M} \sum_{v=1}^{M} f_v, \tag{4.11}$$

to define the labelled error term. The function f^{avg} equals the final predictor f by definition.

Definition 4.7 (CoSVR$^{\text{mod}}$)**.** [Ullrich et al., 2017] For loss functions ℓ^L and ℓ^U as well as hyperparameters $\nu_v, \lambda, \varepsilon^L > 0$, the co-regularised support vector regression problem

with modified constraints for the labelled examples (CoSVR$^{\text{mod}}$) is defined as

$$\min_{f_v \in \mathcal{H}_v} \sum_{v=1}^{M} \frac{\nu_v}{2} \|f_v\|^2 + \sum_{i=1}^{n} \max\{|y_i - f^{\text{avg}}(x_i)| - \varepsilon^L, 0\} \\ + \lambda \sum_{u,v=1}^{M} \sum_{j=1}^{m} \ell^U \left(f_u(z_j), f_v(z_j)\right), \tag{4.12}$$

where $f^{\text{avg}} = 1/M \sum_{v=1}^{M} f_v$ is the view predictor average from Equation 4.11. If ℓ^U is the ε-insensitive loss with $\varepsilon^U > 0$, the problem in Equation 4.12 is called ε-CoSVR$^{\text{mod}}$. The case $\ell^U = \ell_2$ is denoted with ℓ_2-CoSVR$^{\text{mod}}$.

In the following lemma we present solutions for ε-CoSVR$^{\text{mod}}$ and ℓ_2-CoSVR$^{\text{mod}}$.

Lemma 4.8. *[Ullrich et al., 2017] Let $\nu_v, \lambda, \varepsilon^L, \varepsilon^U > 0$ be hyperparameters. We utilise dual variables $\alpha, \hat{\alpha} \in \mathbb{R}^n$ and $\gamma_{uv} \in \mathbb{R}^m$ (compare Table 4.1).*

(i) The ℓ_2-CoSVR$^{\text{mod}}$ dual optimisation problem equals

$$\max_{\alpha,\hat{\alpha}\in\mathbb{R}^n,\ \gamma_{uv}\in\mathbb{R}^m} \sum_{v=1}^{M} \Bigg(-\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha - \hat{\alpha})^T Y \\ -(\alpha + \hat{\alpha})^T \varepsilon^L \mathbf{1}_n - \frac{1}{4\lambda} \sum_{u=1}^{M} \gamma_{uv}^T \gamma_{uv} \Bigg) \\ \text{s. t.} \left\{ \begin{array}{l} \mathbf{0}_n \le \alpha, \hat{\alpha} \le \mathbf{1}_n \\ \gamma_{uv} = \frac{2\lambda}{\nu_u} U_u \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_u - \frac{2\lambda}{\nu_v} U_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v \end{array} \right\}_{v \in [\![M]\!]},$$

where

$$\begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v = \begin{pmatrix} \frac{1}{M}(\alpha - \hat{\alpha}) \\ \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu}) \end{pmatrix}$$

and $\pi_v = \frac{1}{\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$.

(ii) The ε-CoSVR$^{\text{mod}}$ dual optimisation problem can be written as

$$\max_{\alpha,\hat{\alpha}\in\mathbb{R}^n,\ \gamma_{uv}\in\mathbb{R}^m} \sum_{v=1}^{M} \Bigg(-\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha - \hat{\alpha})^T Y \\ -(\alpha + \hat{\alpha})^T \varepsilon^L \mathbf{1}_n - \sum_{u=1}^{M} \gamma_{uv}^T \varepsilon^U \mathbf{1}_m \Bigg) \\ \text{s. t.} \left\{ \begin{array}{l} \mathbf{0}_n \le \alpha, \hat{\alpha} \le \mathbf{1}_n \\ \mathbf{0}_m \le \gamma_{uv} \le \lambda \mathbf{1}_m \end{array} \right\}_{v \in [\![M]\!]},$$

where

$$\begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v = \begin{pmatrix} \frac{1}{M}(\alpha - \hat{\alpha}) \\ \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu}) \end{pmatrix},$$

and $\pi_v = \frac{1}{\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$.

Proof. The proof can be found in Appendix A. □

It is possible to reduce the number of variables even more using modified constraints for the co-regularisation term. Because of the typically small number n of labelled examples, the CoSVR$^{\text{mod}}$ algorithm is rather important from a theoretical perspective as the basis of a further CoSVR variant introduced in Section 4.3.3. The CoSVR variant presented in the following section is beneficial from a practical perspective if the number of views M and the number of unlabelled instances m is large.

Modification of the Co-Regularisation

The unlabelled error term of base CoSVR bounds the pairwise distances of view predictions, whereas now in CoSVR$_{\text{mod}}$ only the disagreement between predictions of each view and the average prediction of the residual views will be taken into account. We use the view-dependent average

$$f_v^{\text{avg}} = \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} f_u$$

of view predictors in the subsequent definition.

Definition 4.9 (CoSVR$_{\text{mod}}$)**.** [Ullrich et al., 2017] We consider RKHSs $\mathcal{H}_1, \ldots, \mathcal{H}_M$, loss functions ℓ^L and ℓ^U, as well as hyperparameters $\varepsilon^L, \nu_v, \lambda > 0$. The co-regularised support vector regression problem with modified constraints for the unlabelled examples (CoSVR$_{\text{mod}}$) is defined as

$$\begin{aligned} \min_{f_v \in \mathcal{H}_v} \sum_{v=1}^{M} \left(\frac{\nu_v}{2} \|f_v\|^2 + \sum_{i=1}^{n} \max\{|y_i - f_v(x_i)| - \varepsilon^L, 0\} \right) \\ + \lambda \sum_{v=1}^{M} \sum_{j=1}^{m} \ell^U \left(f_v^{\text{avg}}(z_j),\ f_v(z_j) \right), \end{aligned} \tag{4.13}$$

where $f_v^{\text{avg}} = 1/(M-1) \sum_{u=1}^{M,u\neq v} f_u$. If ℓ^U is the ε-insensitive loss with $\varepsilon^U > 0$ then the optimisation problem in Equation 4.13 is denoted with ε-CoSVR$_{\text{mod}}$ and the case $\ell^U = \ell_2$ with ℓ_2-CoSVR$_{\text{mod}}$.

Again we present solutions for ℓ_2-CoSVR$_{\text{mod}}$ and ε-CoSVR$_{\text{mod}}$.

Lemma 4.10. *[Ullrich et al., 2017] Let $\nu_v, \lambda, \varepsilon^L, \varepsilon^U > 0$ be hyperparameters. We utilise dual variables $\alpha_v, \hat{\alpha}_v \in \mathbb{R}^n$ and $\gamma_v, \hat{\gamma}_v \in \mathbb{R}^m$, as well as $\gamma_v^{\text{avg}} = \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} \gamma_u$ and $\hat{\gamma}_v^{\text{avg}} = \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} \hat{\gamma}_u$ analogous to the residual view predictor average.*

(i) The ℓ_2-CoSVR$_{\text{mod}}$ dual optimisation problem equals

$$\begin{aligned} \max_{\alpha_v, \hat{\alpha}_v \in \mathbb{R}^n,\ \gamma_v \in \mathbb{R}^m} \sum_{v=1}^{M} \Bigg(& -\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha_v - \hat{\alpha}_v)^T Y \\ & -(\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n - \frac{1}{4\lambda} \sum_{u=1}^{M} \gamma_v^T \gamma_v \Bigg) \\ \text{s. t.} \left\{ \begin{array}{l} \mathbf{0}_n \leq \alpha_v, \hat{\alpha}_v \leq \mathbf{1}_n \\ \gamma_v = \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} \frac{2\lambda}{\nu_u} U_u \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_u - \frac{2\lambda}{\nu_v} U_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v \end{array} \right\}_{v \in [\![M]\!]}, \end{aligned}$$

where

$$\begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v = \begin{pmatrix}\alpha_v - \hat{\alpha}_v\\ \gamma_v - \gamma_v^{\text{avg}}\end{pmatrix}$$

and $\pi_v = \frac{1}{\nu_v}\begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v$.

(*ii*) *The* ε*-CoSVR*$_{\text{mod}}$ *dual optimisation problem can be written as*

$$\begin{aligned}\max_{\alpha_v,\hat{\alpha}_v\in\mathbb{R}^n,\ \gamma_v,\hat{\gamma}_v\in\mathbb{R}^m} \sum_{v=1}^{M} \Bigg(&-\frac{1}{2\nu_v}\begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v^T K_v \begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v + (\alpha-\hat{\alpha})^T Y \\ &-(\alpha_v+\hat{\alpha}_v)^T\varepsilon^L\mathbf{1}_n - (\gamma_v+\hat{\gamma}_v)\varepsilon^U\mathbf{1}_m\Bigg)\\ \text{s.t.} \left\{ \begin{array}{l} \mathbf{0}_n \le \alpha_v, \hat{\alpha}_v \le \mathbf{1}_n \\ \mathbf{0}_m \le \gamma_v, \hat{\gamma}_v \le \lambda\mathbf{1}_m \end{array} \right\}_{v\in[\![M]\!]}&,\end{aligned}$$

where

$$\begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v = \begin{pmatrix}\alpha_v - \hat{\alpha}_v\\ (\gamma_v - \gamma_v^{\text{avg}}) - (\hat{\gamma}_v - \hat{\gamma}_v^{\text{avg}})\end{pmatrix},$$

and $\pi_v = \frac{1}{\nu_v}\begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v$.

Proof. The proof can be found in Appendix A. □

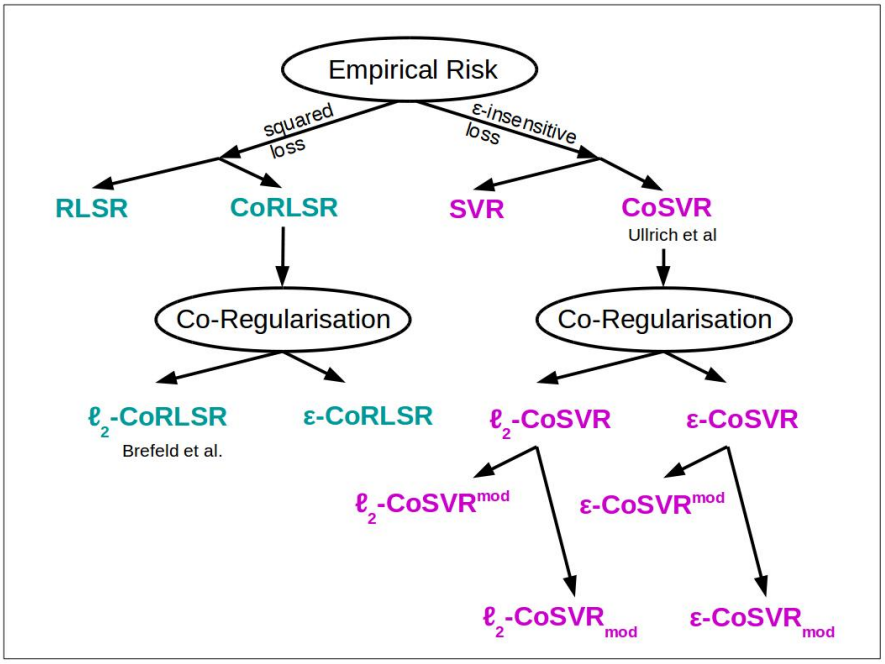
If we combine the modifications in the labelled and unlabelled error term we canonically obtain the variants ℓ_2-CoSVR$_{\text{mod}}^{\text{mod}}$ and ε-CoSVR$_{\text{mod}}^{\text{mod}}$. Moreover, it is possible to analogously define and solve ε-CoRLSR, the variant of CoRLSR with ε-insensitive loss in the unlabelled error. We omitted that for two reasons. Firstly, there are already plenty of co-regularised algorithms to investigate and compare, such that a greater variety of algorithms would probably not be beneficial. Secondly, in comparison to base CoRLSR the ε-CoRLSR modification cannot be solved as closed formula for the same reason this is not possible for single-view SVR (see Section 2.6.2). For an overview of the considered co-regularised approaches and their variants see Figure 4.1. In the taxonomy of Figure 4.1, CoRLSR is used as superordinate concept of ℓ_2-CoRLSR. For the sake of simplicity and in concordance with the work of Brefeld et al. [2006], in the following we denote ℓ_2-CoRLSR with CoRLSR (compare its introduction in Section 4.2).

In the base CoSVR versions the semi-supervision is realised with proximity constraints on pairs of view predictions. We show in the following lemma that the weaker constraints of the closeness of one view prediction to the average of the residual predictions implies a closeness of every pair of predictions too.

Lemma 4.11. *[Ullrich et al., 2017] Up to constants, the unlabelled error bound of CoSVR*$_{\text{mod}}$ *is also an upper bound of the unlabelled error of base CoSVR.*

Proof. We consider the settings of Lemma 4.6 (*i*) and Lemma 4.10 (*i*). For part (*ii*) the proof is equivalent taking $\varepsilon^U = 0$. To start with, in the case of $M = 2$, modified and base algorithm fall together which shows the claim. We continue with $M > 2$. Because of the definition of the ε-insensitive loss we know that $|f_v(z_j) - f_v^{\text{avg}}(z_j)| \le \varepsilon^U + c_{vj}$,

FIGURE 4.1: Overview of single-view and co-regularised approaches



where $c_{vj} \geq 0$ is the unlabelled error value

$$c_{vj} = \max\{|f_v^{\text{avg}}(z_j) - f_v(z_j)| - \varepsilon^U, 0\}$$

for fixed view v and fixed unlabelled instance z_j. We denote the maximum of c_{vj} with respect to all views $v = 1, \ldots, M$ with c_j

$$c_j = \max_{v \in \{1,\ldots,M\}} c_{vj}.$$

Hence, $|f_v(z_j) - f_v^{\text{avg}}(z_j)| \leq \varepsilon^U + c_j$ for all $v = 1, \ldots, M$. Now we conclude for all $j \in \{1, \ldots, m\}$ and $(u, v) \in \{1, \ldots, M\}^2$

$$\begin{aligned}
&|f_u(z_j) - f_v(z_j)| \\
&\leq |f_u(z_j) - f_u^{\text{avg}}(z_j)| + |f_u^{\text{avg}}(z_j) - f_v^{\text{avg}}(z_j)| + |f_v^{\text{avg}}(z_j) - f_v(z_j)| \\
&\leq \varepsilon^U + c_j + \tfrac{1}{M-1}|f_v(z_j) - f_u(z_j)| + \varepsilon^U + c_j,
\end{aligned}$$

and, therefore,

$$|f_u(z_j) - f_v(z_j)| \leq \tfrac{2(M-1)}{M-2}(\varepsilon^U + c_j). \tag{4.14}$$

We consider the upper bound B

$$\sum_{v=1}^{M} \sum_{j=1}^{m} \ell_{\varepsilon^U}(f_v^{\text{avg}}(z_j), f_v(z_j)) \leq M \sum_{j=1}^{m} c_j = B$$

of the unlabelled error of $CoSVR_{\text{mod}}$. From Equation 4.14 we conclude that the unlabelled error of CoSVR is bounded by $\tilde{B}$

$$\sum_{u,v=1}^{M} \sum_{j=1}^{m} \ell_{\tilde{\varepsilon}}(f_u(z_j), f_v(z_j)) \leq \tilde{B}$$

for $\tilde{\varepsilon} = \frac{2(M-1)}{M-2}\varepsilon^U$ and $\tilde{B} = \frac{2M(M-1)}{(M-2)}B$, which finishes the proof. □

The complexity class of the base CoSVR variants ℓ_2-CoSVR and ε-CoSVR is $\mathcal{O}(Mn + M^2m)$, where M is the number of views, n is the number of labelled examples and m the number of unlabelled instances. Via the presented modifications of CoSVR in the labelled and unlabelled error terms, the number of variables was reduced significantly (the precise numbers can be found in Table 4.2). In the case of $\text{CoSVR}^{\text{mod}}$ a reduction to $\mathcal{O}(n + M^2m)$ could be derived. The variable number reduction of $\text{CoSVR}_{\text{mod}}$ to $\mathcal{O}(Mn + Mm)$ is even more effective, as the number of labelled instances m is typically greater than the number of labelled examples n, which implies a complexity class of $\mathcal{O}(Mm)$. We point to the fact that the variable number of CoRLSR that can be solved analytically is also $\mathcal{O}(Mn + Mm)$. More details on computational aspects can be found in Section 4.3.4. A number of optimisation variables in complexity class $\mathcal{O}(Mm)$ is still a lot if one intends to solve a QP problem with an appropriate solver (compare Section 4.3.4). It is therefore even more valuable that there is a single-view reformulation of ℓ_2-$\text{CoSVR}^{\text{mod}}$.

4.3.3 Σ-CoSVR

Sindhwani and Rosenberg [2008] showed that a subset of co-regularisation approaches can be reformulated as single-view approach with fused kernel and sum space $\mathcal{H}_\Sigma$

$$\mathcal{H}_\Sigma = \{f : f = f_1 + f_2, f_1 \in \mathcal{H}_1, f_2 \in \mathcal{H}_2\}, \tag{4.15}$$

where $\mathcal{H}_1$ and $\mathcal{H}_2$ are RKHSs with reproducing kernels k_1 and k_2, respectively. The Σ in $\mathcal{H}_\Sigma$ symbolises the sum of functions or kernels (in contrast to Chapter 3, where Σ denotes an alphabet). The precise co-regularisation subset considered by Sindhwani and Rosenberg [2008] is characterised by a two-view setting, i.e., $M = 2$, a co-regularisation term with squared loss function $\ell^U = \ell_2$, and an empirical risk calculated with the average predictor f^{avg} according to Equation 4.11. The corresponding optimisation was called *co-regularised least squares* (CoRLS) algorithm by Rosenberg and Bartlett [2007] and can be formulated as

$$\min_{f \in \mathcal{H}_\Sigma} \sum_{v=1,2} \nu_v \|f_v\|^2_{\mathcal{H}_v} + \sum_{i=1}^{n} \ell^L \left(y_i, \frac{f_1(x_i) + f_2(x_i)}{2} \right) + \lambda \sum_{j=1}^{m} |f_1(z_j) - f_2(z_j)|^2 \tag{4.16}$$

using the notation from above. The minimisation in Equation 4.16 is a generalisation of ℓ_2-CoSVR. Note that CoRLS is related to but not equal with CoRLSR, which was introduced in Section 4.2. An overview of algorithms including CoRLS can be found in Figure 4.2 below.

Let k be a kernel with Gram matrix K and $Z = \{z_1, \ldots, z_m\} \subseteq \mathcal{X}$ be the set of unlabelled instances. With $k(Z, x)$ and $k(Z, Z)$ we refer to the submatrices

$$k(Z, x) = (k(z_1, x), \ldots, k(z_m, x))^T$$

and

$$k(Z, Z) = \big(k(z_j, z_{j'})\big)_{j,j'=1}^{m}$$

of the Gram matrix K. Furthermore, for hyperparameters $\nu_1, \nu_2 > 0$ we fix the kernel linear combinations $k^{\oplus}$ and $k^{\ominus}$ via

$$k^{\oplus} = \frac{1}{\nu_1} k_1 + \frac{1}{\nu_2} k_2 \quad \text{and} \quad k^{\ominus} = \frac{1}{\nu_1} k_1 - \frac{1}{\nu_2} k_2$$

for two kernel functions k_1 and k_2. In the subsequent theorem, we present the result of Sindhwani and Rosenberg [2008].

Theorem 4.12. *We consider two RKHSs $\mathcal{H}_1$ and $\mathcal{H}_2$ with reproducing kernels $k_v : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $v = 1, 2$, together with the scenario and symbols from above. Let $\nu_1, \nu_2, \lambda > 0$ be hyperparameters. The optimisation*

$$\min_{f \in \mathcal{H}} \|f\|_{\mathcal{H}_\Sigma}^2 + \sum_{i=1}^{n} \ell^L \left(y_i, \frac{1}{2} f(x_i) \right) \tag{4.17}$$

with norm

$$\|f\|_{\mathcal{H}_\Sigma}^2 = \min_{f = f_1 + f_2,\ f_v \in \mathcal{H}_v} \sum_{v=1,2} \nu_v \|f_v\|_{\mathcal{H}_v}^2 + \lambda \sum_{j=1}^{m} |f_1(z_j) - f_2(z_j)|^2$$

is a reformulation of the CoRLS problem in Equation 4.16 and $\mathcal{H}_\Sigma$ is an RKHS with kernel k_Σ

$$k_\Sigma(x, x') = k^{\oplus}(x, x') - k^{\ominus}(Z, x)^T \left(\frac{1}{\lambda} \mathbf{I}_m + k^{\oplus}(Z, Z) \right)^{-1} k^{\ominus}(Z, x') \tag{4.18}$$

for $x, x' \in \mathcal{X}$ and $Z \subset \mathcal{X}$.

The following definition picks up the single-view CoRLS reformulation of Sindhwani and Rosenberg [2008] with an ε-insensitive loss function (compare also Figure 4.2).

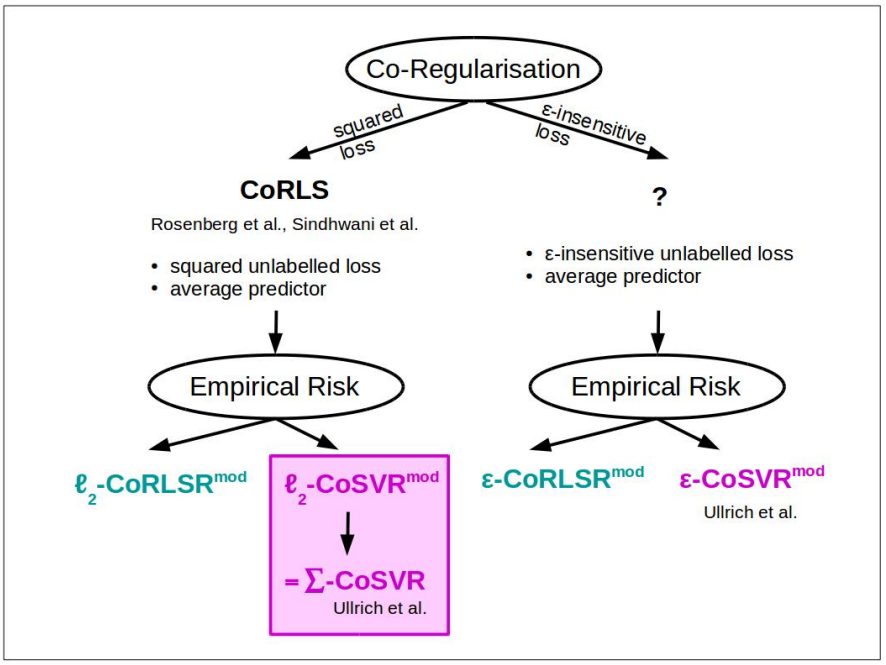
Definition 4.13 (Σ-CoSVR)**.** [Ullrich et al., 2017] Let $\mathcal{H}_\Sigma$ be the RKHS from Equation 4.15 with fused kernel function k_Σ from Equation 4.18. For $\varepsilon^L > 0$, we denote the SVR optimisation

$$\min_{f \in \mathcal{H}_\Sigma} \|f\|_{\mathcal{H}_\Sigma}^2 + \sum_{i=1}^{n} \max\{|y_i - \frac{1}{2} f(x_i)| - \varepsilon^L, 0\}, \tag{4.19}$$

Σ-co-regularised support vector regression (Σ-CoSVR).

We point out that the two RKSHs $\mathcal{H}_1$ and $\mathcal{H}_2$, their corresponding kernel functions k_1 and k_2, and the hyperparameters ν_1, ν_2, and λ appear in the definitions of k_Σ and $\|\cdot\|_{\mathcal{H}_\Sigma}$. For each pair x and x' of instances, the value of $k_\Sigma(x, x')$ in Equation 4.18 is

FIGURE 4.2: Overview of co-regularised approaches with two views and average predictor



calculated via kernel values of k_1 and k_2 including not only x and x' themselves but also the unlabelled examples in Z. The optimisation problem in Equation 4.19 is a standard SVR minimisation with additional information about unlabelled examples incorporated in the RKHS $\mathcal{H}_\Sigma$.

Lemma 4.14. *[Ullrich et al., 2017] The algorithms ℓ_2-CoSVR$^{\text{mod}}$ and Σ-CoSVR are equivalent.*

Proof. The proof directly follows from Theorem 2.2 of Sindhwani and Rosenberg [2008] for ε-insensitive loss function, where $\varepsilon^L, \nu_1, \nu_2, \lambda > 0$ are the hyperparameters of the ℓ_2-CoSVR$^{\text{mod}}$ optimisation. □

As Σ-CoSVR can be solved as a standard SVR algorithm with $\mathcal{O}(n)$ variables, we obtained a multi-view approach with single-view time complexity. The information of the two views and the unlabelled examples are included in the candidate space $\mathcal{H}_\Sigma$ and associated reproducing kernel k_Σ. More details on computational aspects of the presented co-regularisation algorithms for regression can be found in the following section.

4.3.4 Computational Aspects

The CoRLSR and CoSVR optimisation approaches mainly differ in the kind of loss functions and whether these are applied to every view predictor or the average of the view predictors. The precise settings result in different numbers of variables and constraints in total, as well as potentially non-zero variables (compare Table 4.2 and the proof of Lemma 4.19). The numbers of variables, constraints, and the number of non-zero variables determine storage space of the machine learning model and the running time

TABLE 4.2: Overview of variables and constraints for different CoSVR versions and CoRLSR

Algorithm	**M**	**# Variables**	**Variable Identifiers**
ε-CoSVR	≥ 2	$2[M]n + M^2m$	$\alpha_{[v]}, \hat{\alpha}_{[v]}, \gamma_{uv}$
ℓ_2-CoSVR	≥ 2	$2[M]n + \frac{1}{2}(M^2 - M)m$	$\alpha_{[v]}, \hat{\alpha}_{[v]}, \gamma_{uv}$
ε-CoSVR$_{\text{mod}}$	> 2	$2[M]n + 2Mm$	$\alpha_{[v]}, \hat{\alpha}_{[v]}, \gamma_v, \hat{\gamma}_v$
ε-CoSVR$_{\text{mod}}$	$= 2$	$2[M]n + 2m$	$\alpha_{[v]}, \hat{\alpha}_{[v]}, \gamma_v, \hat{\gamma}_v, \gamma_1 = \hat{\gamma}_2, \gamma_2 = \hat{\gamma}_1$
ℓ_2-CoSVR$_{\text{mod}}$	> 2	$2[M]n + Mm$	$\alpha_{[v]}, \hat{\alpha}_{[v]}, \gamma_v$
ℓ_2-CoSVR$_{\text{mod}}$	$= 2$	$2[M]n + m$	$\alpha_{[v]}, \hat{\alpha}_{[v]}, \gamma_v, \gamma_1 = -\gamma_2$
Σ-CoSVR	$= 2$	$2n$	$\alpha, \hat{\alpha}$
CoRLSR	≥ 2	$Mn + Mm$	π_v

Algorithm	**M**	**# Constraints**	**# Non-Zero Variables**
ε-CoSVR	≥ 2	$4[M]n + 2M^2m$	$[M]n + \frac{1}{2}(M^2 - M)m$
ℓ_2-CoSVR	≥ 2	$4[M]n + \frac{1}{2}(M^2 - M)m$	$[M]n + \frac{1}{2}(M^2 - M)m$
ε-CoSVR$_{\text{mod}}$	> 2	$4[M]n + 4Mm$	$[M]n + Mm$
ε-CoSVR$_{\text{mod}}$	$= 2$	$4[M]n + 4m$	$[M]n + m$
ℓ_2-CoSVR$_{\text{mod}}$	> 2	$4[M]n + Mm$	$[M]n + Mm$
ℓ_2-CoSVR$_{\text{mod}}$	$= 2$	$4[M]n + m$	$[M]n + m$
Σ-CoSVR	$= 2$	$4n$	n
CoRLSR	≥ 2	0	$Mn + Mm$

of the corresponding algorithm. We use the formulation of a *sparse* vector (or matrix) to indicate that the considered vector (or matrix) predominantly contains zeros. This property of vectors (or matrices) is also referred to as *sparsity*.

From the Lagrangian theory (see Section 2.3) we know that at least half of the dual α- and γ-variables (see Table 4.1) must be zero in the case of ε-insensitive loss function. According to the *Karush-Kuhn-Tucker conditions* [Boyd and Vandenberghe, 2004], only for active inequality constraints the corresponding dual α- and γ-variables can be non-zero at all. Because of the precise optimisation problem setting with ε-insensitive loss for the CoSVR variants, the number of non-zero variables in the learned model will be even smaller than the numbers reported in the non-zero variables column of Table 4.2. The actual number of non-zero variables in the final solution depends on the choice of ε. Labelled and unlabelled instances $x_i \in X$ and $z_j \in Z$ play the role of (labelled and unlabelled) support vectors if the corresponding dual variables are different from zero. Hence, the CoSVR versions admit a final model representation of only few support vectors relative to the overall number of labelled and unlabelled training examples. The property of the solution vector in the CoSVR variants to be sparse allows for a more efficient model storage compared to CoRLSR [Brefeld et al., 2006].

We summarised the numerical information on variables and constraints for the different algorithms in Table 4.2. The respective CoSVR$^{\text{mod}}$ variants are included in the table by cancelling the factor $[M]$ and the index $[v]$. In the special case of $M = 2$, the base and the respective CoSVR$_{\text{mod}}$ version fall together. In practical scenarios, the number m of unlabelled instances is greater than the number n of labelled examples. Furthermore,

the summands linear in n are at most linear in the number of views M. For this reason, the summands linear in m will mostly influence running time and memory requirements (see Table 4.2). To be more precise, the number of variables of ε-CoSVR is $2Mn+M^2m$. This can be reduced to $\mathcal{O}(Mm)$, which is the complexity class of the variable number of CoRLSR, via the loss function approach of ε-CoSVR$_{\text{mod}}$. The variable number reduction of ε-CoSVR$^{\text{mod}}$ is rather from a theoretical value. As mentioned already above, in the case $M = 2$ the variants ε-CoSVR and ε-CoSVR$_{\text{mod}}$ fall together. Because of the symmetry of variables for $M = 2$ the actual number of variables in Table 4.2 is even smaller than in the case of $M > 2$. Analogous considerations hold true for the ℓ_2-CoSVR variants. A substantial reduction of variable numbers can be achieved via the Σ-CoSVR approach where the number of variables $2n$ is the number of variables of a single-view SVR.

All presented problems are convex QPs with positive semi-definite matrices in the quadratic terms. According to Kozlov et al. [1980] QPs can be solved in polynomial time complexity in the number of optimisation variables, if the Hessian matrix in the squared term of the objective is positive definite. As the number m in real-world problems is greater than n, the running time of a QP-solver will be dominated by the respective second summand in the constraints column of Table 4.2 (except for Σ-CoSVR). Consequently the variable reduction of CoSVR$_{\text{mod}}$ compared to base CoSVR from $\mathcal{O}(Mn+M^2m)$ to $\mathcal{O}(Mm)$ reduces the running time significantly. However, a clear advantage of CoRLSR is that it can be solved as a system of $Mn+Mm$ equations which has cubic time complexity in the number of variables or equations, respectively.

4.3.5 A Rademacher Bound for CoSVR

The empirical Rademacher complexity $\hat{\mathcal{R}}_n$ from Definition 2.6 is a data-dependent measure for the capacity of a function class $\mathcal{H}$ to fit random data [Shawe-Taylor and Cristianini, 2004]. Rosenberg and Bartlett [2007] presented a bound on the empirical Rademacher complexity of CoRLS for the case $M = 2$ (compare Section 4.3.3). Inspired by the result of Rosenberg and Bartlett [2007], we prove empirical Rademacher complexity bounds for the ε-CoSVR$^{\text{mod}}$ and ℓ_2-CoSVR$^{\text{mod}}$ function classes. To this aim we fix the following notation. If a function $f_v \in \mathcal{H}_v$ has got a representation $f(\cdot) = \sum_{i=1}^{n+m} \pi_i k(x_i, \cdot)$ with coefficients $\pi \in \mathbb{R}^{n+m}$, we denote the kernel linear coefficients with $\pi(f)$ in order to indicate their relation to the function f.

Definition 4.15 (ℓ_2-CoSVR$^{\text{mod}}$ and ε-CoSVR$^{\text{mod}}$ function class)**.** [Ullrich et al., 2017] Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two RKHSs with kernels k_1 and k_2. Let K_1 and K_2 be the corresponding Gram matrices with lower submatrices U_1 and U_2, $\lambda, \nu_1, \nu_1, \mu > 0$ be hyperparameters, as well as $\mathcal{H}_\Sigma$ be the sum space from Equation 4.15. With

$$\begin{aligned}\mathcal{H}_\Sigma^{\ell_2} = \{f = f_1 + f_2 :\ & f_1 \in \mathcal{H}_1, f_2 \in \mathcal{H}_2, \\ & \nu_1 \pi_1^T K_1 \pi_1 + \nu_2 \pi_2^T K_2 \pi_2 + \lambda (U_1\pi_1 - U_2\pi_2)^T (U_1\pi_1 - U_2\pi_2) \leq 1\}\end{aligned} \tag{4.20}$$

and

$$\begin{aligned}\mathcal{H}_\Sigma^{\varepsilon} = \{f = f_1 + f_2 : & f_1 \in \mathcal{H}_1, f_2 \in \mathcal{H}_2, \\ & -\mu \mathbf{1}_{n+m} \leq \pi_1, \pi_2 \leq \mu \mathbf{1}_{n+m}\}\end{aligned} \tag{4.21}$$

we define two bounded versions of $\mathcal{H}_\Sigma$, where $\pi_1 = \pi_1(f_1)$ and $\pi_2 = \pi_2(f_2)$.

The meaning of parameter μ will become obvious in Lemma 4.19 (*ii*) below. The subsequent theorem of Rosenberg and Bartlett [2007] delivered a *generalisation bound*, which is a bound on the difference between the expected risk and the empirical risk (compare Equations 2.5 and 2.6 above).

Theorem 4.16. *[Rosenberg and Bartlett, 2007] Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ be a loss function according to Definition 2.2 and $\mathcal{J}$ be a RKHS of functions. Without loss of generality, assume*

$$\ell(f(x), y) \to [0, 1] \tag{4.22}$$

for all $f \in \mathcal{J}$. Moreover, let ℓ satisfy a uniform Lipschitz condition, i.e., there is a constant $B > 0$ such that

$$\frac{|\ell(f(x_1), \hat{y}) - \ell(f(x_2), \hat{y})|}{|f(x_1) - f(x_2)|} \leq B \tag{4.23}$$

holds true for all $\hat{y} \in \mathcal{Y}$, $x_1, x_2 \in \mathcal{X}$ with $x_1 \neq x_2$, and $f \in \mathcal{J}$. For labelled training data $(x_1, y_1), \ldots, (x_n, y_n)$ drawn i.i.d. from distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ and $\delta \in (0, 1)$

$$\mathbb{E}_{\mathcal{D}}(\ell(y, f(x)) \leq \hat{\mathbb{E}}(\ell(y, f(x)) + \hat{\mathcal{R}}_n(\mathcal{J}) + \frac{1}{\sqrt{n}} \left(2 + 3\sqrt{\frac{\ln(2/\delta)}{2}}\right) \tag{4.24}$$

is valid for every $f \in \mathcal{J}$ with probability at least $(1 - \delta)$.

Proof. The proof is a consequence of Theorem 2.7 [Shawe-Taylor and Cristianini, 2004] above as well as the additional precondition of the boundedness in Equation 4.22 and the Lipschitz-condition in Equation 4.23. It was presented by Rosenberg and Bartlett [2007]. □

The following lemma is an elaborate presentation of the result in [Ullrich et al., 2017].

Lemma 4.17. *Let $\mathcal{H}_\Sigma^{\ell_2}$ and $\mathcal{H}_\Sigma^{\varepsilon}$ be the function class subsets of $\mathcal{H}_\Sigma$ from Equations 4.20 and 4.21. The generalisation bound in Equation 4.24 holds true for $\mathcal{H}_\Sigma^{\ell_2}$ and $\mathcal{H}_\Sigma^{\varepsilon}$, if Equation 4.22 is fulfilled for all $f \in \mathcal{H}_\Sigma^{\ell_2}$ and all $f \in \mathcal{H}_\Sigma^{\varepsilon}$, respectively.*

Rosenberg and Bartlett [2007] demanded a boundedness property of the function class $\mathcal{J}$ in Equation 4.22. A weaker condition is the so-called *M-boundedness* [Cucker and Zhou, 2007] of a function class $\mathcal{J}$, when there is a constant M such that $\ell(f(x), y) \leq M$ for all $f \in \mathcal{J}$. If $\mathcal{J}$ was M-bounded, the constant M (which is different from the number M of views in the general multi-view case) would appear in Equation 4.24 as well. The M-boundedness is closely related to the fact that the kernel function k is bounded [Cucker and Zhou, 2007], i.e., $\sup_{x \in \mathcal{X}} k(x, x) \leq M$ for a constant M. This is a realistic and accomplishable assumption in our ligand affinity prediction scenario. More precisely, the linear kernel applied to binary fingerprint vectors Φ of dimension d fulfills $\langle \Phi(x), \Phi(x) \rangle \leq d$ for every molecular compound x.

Proof. It remains to show that the squared loss ℓ_2 and the ε-insensitive loss ℓ_ε satisfy a Lipschitz condition. Firstly, regarding the squared loss ℓ_2 we conclude

$$\begin{aligned}\frac{\big||f(x_1)-\hat{y}|^2-|f(x_2)-\hat{y}|^2\big|}{\big|f(x_1)-f(x_2)\big|} &\leq \frac{\big(|f(x_1)-\hat{y}|+|f(x_2)-\hat{y}|\big)\big(|f(x_1)-\hat{y}|-|f(x_2)-\hat{y}|\big)\big|}{\big|f(x_1)-f(x_2)\big|}\\ &\leq \frac{|f(x_1)+f(x_2)-2\hat{y}||f(x_1)-f(x_2)|}{\big|f(x_1)-f(x_2)\big|}\\ &\leq |f(x_1)-\hat{y}|+|f(x_2)-\hat{y}| \leq 1+1=2\end{aligned}$$

from Equation 4.22 for all $x_1, x_2 \in \mathcal{X}$, $\hat{y} \in \mathcal{Y}$, and all $f \in \mathcal{H}_\Sigma^{\ell_2}$. Hence, the Lipschitz condition is valid for ℓ_2 with $B = 2$. The Lipschitz condition of the ε-insensitive loss ℓ_ε will be proven by distinction of cases:
Case 1: $|f(x_1) - y| < \varepsilon$ and $|f(x_1) - y| < \varepsilon$ implies that

$$\big|\max\{0, |f(x_1)-y|-\varepsilon\} - \max\{0, |f(x_2)-y|-\varepsilon\}\big| = 0.$$

Case 2: $|f(x_1) - y| \geq \varepsilon$ and $|f(x_1) - y| \geq \varepsilon$ implies that

$$\begin{aligned}&\big|\max\{0, |f(x_1)-y|-\varepsilon\} - \max\{0, |f(x_2)-y|-\varepsilon\}\big|\\ &= \big||f(x_1)-y|-\varepsilon-(|f(x_2)-y|-\varepsilon)\big|\\ &= \big||f(x_1)-y|-|f(x_2)-y|\big| \leq \big|f(x_1)-y-(f(x_2)-y)\big| = \big|f(x_1)-f(x_2)\big|.\end{aligned}$$

Case 3: Without loss of generality, $|f(x_1) - y| \geq \varepsilon$ and $|f(x_1) - y| < \varepsilon$ implies that there is an $x_3 \in \mathcal{X}$ such that $|f(x_3) - y| = \varepsilon$ and $|f(x_1) - f(x_3)| < |f(x_1) - f(x_2)|$. Consequently, according to case 2

$$\begin{aligned}&\big|\max\{0, |f(x_1)-y|-\varepsilon\} - \max\{0, |f(x_2)-y|-\varepsilon\}\big|\\ &= \big|\max\{0, |f(x_1)-y|-\varepsilon\} - \max\{0, |f(x_3)-y|-\varepsilon\}\big|\\ &\leq |f(x_1)-f(x_3)| \leq |f(x_1)-f(x_2)|.\end{aligned}$$

Case 4: Without loss of generality, $|f(x_1)-y| \geq \varepsilon$ and $|f(x_1)-y| \leq \varepsilon$ implies that there are $x_3, x_4 \in \mathcal{X}$ such that $|f(x_3)-y| = \varepsilon$ and $|f(x_1)-f(x_3)| < |f(x_1)-f(x_2)|$ and such that $|f(x_4)-y| = \varepsilon$ and $|f(x_2)-f(x_4)| < |f(x_1)-f(x_2)|$. Consequently, according to case 2

$$\begin{aligned}&\big|\max\{0, |f(x_1)-y|-\varepsilon\} - \max\{0, |f(x_2)-y|-\varepsilon\}\big|\\ &= \big|\max\{0, |f(x_1)-y|-\varepsilon\} - \max\{0, |f(x_3)-y|-\varepsilon\}\\ &\quad + \max\{0, |f(x_4)-y|-\varepsilon\} - \max\{0, |f(x_2)-y|-\varepsilon\}\big|\\ &= \big|\max\{0, |f(x_1)-y|-\varepsilon\} - \max\{0, |f(x_3)-y|-\varepsilon\}\big|\\ &\quad + \big|\max\{0, |f(x_4)-y|-\varepsilon\} - \max\{0, |f(x_2)-y|-\varepsilon\}\big|\\ &\leq |f(x_1)-f(x_3)| + |f(x_4)-f(x_2)| \leq 2|f(x_1)-f(x_2)|.\end{aligned}$$

Hence, the Lipschitz condition is satisfied for ℓ_ε with $B = 2$ as well. □

In the following theorem, Rosenberg and Bartlett [2007] considered the precise function class $\mathcal{J}$

$$\mathcal{J} = \Bigg\{ f = \frac{f_1 + f_2}{2} : f_1 \in \mathcal{H}_1, f_2 \in \mathcal{H}_2, \sum_{v=1,2} \nu_v \|f_v\|^2_{\mathcal{H}_v} + \lambda \sum_{j=1}^{m} |f_1(z_j) - f_2(z_j)|^2 \leq 1 \Bigg\}. \tag{4.25}$$

on the basis of the CoRLS problem formulation from Equation 4.16. We will make use of two decompositions of the Gram matrices K_1 and K_2 of the reproducing kernels k_1 and k_2 into submatrices

$$K_1 = \begin{pmatrix} A & C \\ C^T & B \end{pmatrix} \quad \text{and} \quad K_2 = \begin{pmatrix} D & F \\ F^T & E \end{pmatrix},$$

where $A, D \in \mathbb{R}^{n \times n}$, $C, F \in \mathbb{R}^{n \times m}$, and $B, E \in \mathbb{R}^{m \times m}$.

Theorem 4.18. *[Rosenberg and Bartlett, 2007] For the function class $\mathcal{J}$ from Equation 4.25*

$$\frac{1}{\sqrt[4]{2}} \frac{M_\Sigma}{n} \leq \hat{\mathcal{R}}_n(\mathcal{J}) \leq \frac{M_\Sigma}{n} \tag{4.26}$$

holds true, where

$$M_\Sigma^2 = \frac{1}{\nu_1} \text{tr}(A) + \frac{1}{\nu_2} \text{tr}(D) - \lambda \text{tr}(J^T (\mathbf{I}_m + \lambda H)^{-1} J)$$

as well as $J = \frac{1}{\nu_1} C^T - \frac{1}{\nu_2} F^T$ and $H = \frac{1}{\nu_1} B + \frac{1}{\nu_2} E$.

Proof. The proof was presented by Rosenberg and Bartlett [2007]. □

Now we finally prove a bound on the empirical Rademacher complexities of $\mathcal{H}_\Sigma^{\ell_2}$ and $\mathcal{H}_\Sigma^{\varepsilon}$, respectively.

Lemma 4.19. *[Ullrich et al., 2017] Let $\mathcal{H}_\Sigma^{\ell_2}$ and $\mathcal{H}_\Sigma^{\varepsilon}$ be the function spaces presented in Equations 4.20 and 4.21 above based on two RKHS $\mathcal{H}_1$ and $\mathcal{H}_2$ with reproducing kernels k_1 and k_2. Let $\|L_1\|_\infty$ and $\|L_2\|_\infty$ be matrix norms of the upper submatrices of the Gram matrices K_1 and K_2 according to the decomposition in Equation 4.4. Furthermore, with s we denote the sparsity of the kernel expansion vectors $\pi_1(f_1)$ and $\pi_2(f_2)$, i.e., the maximal number of their components not equal to zero.*

(i) The empirical Rademacher complexity of the ℓ_2-CoSVR$^{\text{mod}}$ function class $\mathcal{H}_\Sigma^{\ell_2}$ can be bounded via

$$\frac{\sqrt[4]{2}^3}{n} \sqrt{tr_n(K_\Sigma)} \leq \hat{\mathcal{R}}_n(\mathcal{H}_\Sigma^{\ell_2}) \leq \frac{2}{n} \sqrt{tr_n(K_\Sigma)}, \tag{4.27}$$

where $tr_n(K_\Sigma) = \sum_{i=1}^n k_\Sigma(x_i, x_i)$ and k_Σ is the sum kernel of Equation 4.18.

(ii) The empirical Rademacher complexity of the ε-CoSVR$^{\text{mod}}$ function class $\mathcal{H}_\Sigma^\varepsilon$ can be bounded via

$$\hat{\mathcal{R}}_n(\mathcal{H}_\Sigma^\varepsilon) \leq \frac{2s}{n}\mu(\|L_1\|_\infty + \|L_2\|_\infty), \tag{4.28}$$

where $\mu = \max\{\frac{1}{2\nu_1}, \frac{1}{2\nu_2}, \frac{2\lambda}{\nu_1}, \frac{2\lambda}{\nu_2}\}$ and $\lambda, \nu_1, \nu_2 > 0$ are the hyperparameters of ε-CoSVR$^{\text{mod}}$ in Equation 4.12.

Proof. *(i)* By definition of the sum kernel k_Σ from Equation 4.18 it holds true that

$$M_\Sigma = \operatorname{tr}_n\left(\sqrt{\frac{1}{\nu_1}\operatorname{tr}(A) + \frac{1}{\nu_2}\operatorname{tr}(D) - \lambda\operatorname{tr}(J^T(\mathbf{I}_m + \lambda H)^{-1}}\right) = \operatorname{tr}_n(K_\Sigma),$$

where K_Σ is the Gram matrix of k_Σ. From the definition of $\mathcal{J}$ in Equation 4.25 and of $\mathcal{H}_\Sigma^{\ell_2}$ in Equation 4.20 as well as Definition 2.6 of the empirical Rademacher complexity we conclude

$$\hat{\mathcal{R}}_n(\mathcal{H}_\Sigma^{\ell_2}) = 2\hat{\mathcal{R}}_n(\mathcal{J}),$$

which combined with Equation 4.26 proves the claim.

(ii) From the box constraints of the dual optimisation problem of ε-CoSVR$^{\text{mod}}$ we know that for the kernel expansion coefficients

$$|\pi_{vi}| \leq \frac{1}{M\nu_v} = \frac{1}{2\nu_v} \quad \text{for} \quad i = 1, \ldots, n, v = 1, 2$$

and

$$|\pi_{vi}| \leq \frac{M\lambda}{\nu_v} = \frac{2\lambda}{\nu_v} \quad \text{for} \quad i = n+1, \ldots, n+m, v = 1, 2$$

holds true. Hence,

$$|\pi_{vi}| \leq \mu = \max\{\tfrac{1}{2\nu_1}, \tfrac{1}{2\nu_2}, \tfrac{2\lambda}{\nu_1}, \tfrac{2\lambda}{\nu_2}\} \tag{4.29}$$

for $i = 1, \ldots, n+m$ and $v = 1, 2$ is valid. We continue to consider the concatenated vector

$$\pi = \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix} \in \mathbb{R}^{2(n+m)}$$

in the sequel of the proof. Additionally, we fix the concatenated matrix $L \in \mathbb{R}^{n \times 2(n+m)}$ with

$$L = (L_1 \; L_2),$$

where L_1 and L_2 are the upper parts of the Gram matrices K_1 and K_2. Let s be the sparsity of the vector π, i.e., the number of its components different from zero (compare Section 4.3.4). From the dual optimisation problem of ε-CoSVR$^{\text{mod}}$, we know that

$s \ll 2(n+m)$. Moreover, for a constant $c > 0$ let the scaled ball $c \cdot B_1$ be defined as

$$c \cdot B_1 = \left\{ x \in \mathbb{R}^{2(n+m)} : \sum_{i=1}^{2(n+m)} |x_i| \leq c \right\}.$$

From the sparsity property and the definition of μ in Equation 4.29 we conclude

$$\sum_{i=1}^{2(n+m)} |x_i| \leq s\mu.$$

Hence, π lies in the scaled ball $s\mu \cdot B_1$. With instances $x_1, \ldots, x_n$ drawn i.i.d. from $\mathcal{X}$ and Bernoulli random variables $\sigma = (\sigma_1, \ldots, \sigma_n)^T$, we reformulate the empirical Rademacher complexity from Definition 2.6 as follows

$$\hat{\mathcal{R}}_n(\mathcal{H}_\Sigma^\varepsilon) \;=\; \mathbb{E}^\sigma \left[\sup_{f \in \mathcal{H}_\Sigma^\varepsilon} \left| \frac{2}{n} \sum_{i=1}^{n} \sigma_i f(x_i) \right| \right] \leq \frac{2}{n} \mathbb{E}^\sigma \left[\sup_{\pi \in s\mu B_1} |\sigma^T L \pi| \right]. \tag{4.30}$$

From Theorems II.2.3 and II.2.4 in [Werner, 1995] we know that

$$\sup_{\pi \in s\mu \cdot B_1} |\langle x, \pi \rangle| = s\mu \|x\|_\infty \tag{4.31}$$

is valid for all $v \in \mathbb{R}^{2(n+m)}$. From Equations 4.30 and 4.31 we conclude

$$\hat{\mathcal{R}}_n(\mathcal{H}_\Sigma^\varepsilon) \leq \frac{2}{n} \mathbb{E}^\sigma s\mu \|\sigma^T L\|_\infty = \frac{2}{n} \mathbb{E}^\sigma s\mu \|\sigma\|_\infty \|L\|_\infty = \frac{2}{n} \mathbb{E}^\sigma s\mu \|L\|_\infty$$

As $\|L\|_\infty$ is the row sum norm of L with

$$\|L\|_\infty = \max_{i \in \{1,\ldots,n\}} \sum_{j=1}^{n+m} \sum_{v=1,2} |k_v(x_i, x_j)|.$$

we finally obtain

$$\hat{\mathcal{R}}_n(\mathcal{H}_\Sigma^\varepsilon) \leq \frac{2s}{n} \mu (\|L_1\|_\infty + \|L_2\|_\infty),$$

which finishes the proof. □

4.4 Empirical Evaluation

In this section we investigate the performance of the co-regularisation algorithms introduced above and corresponding baselines at the prediction of small compound's affinity values with respect to considered target proteins.

4.4.1 Datasets, Implementation, and Experimental Setting

We performed our empirical analysis with 22 datasets which are described in detail in Appendix B. Each dataset contains between 21 and 600 ligands and their positive real-valued affinity towards a given human protein. For the representation of ligands we utilised the fingerprints ECFP4, GpiDAPH3, and Maccs. All included fingerprint types are binary, high-dimensional, and sparse. The performance analysis of co-regularised algorithms is based on the empirical section in [Ullrich et al., 2016a] and [Ullrich et al., 2017]. For the above-mentioned dataset we comment on the elaborate presentation in detail below.

The practical experiments were implemented with *Python* 2.7[1]. We developed an experimental framework which is available as open source[2]. A variety of co-regularised algorithms and baselines are available within the framework. An overview of the implemented algorithms can be found in Table 4.3. We used the *CVXOPT optimisation package*[3] for the solution of all single- and multi-view optimisation problems with ε-insensitive loss function. In order to solve the respective QPs with positive semi-definite matrices in the quadratic terms as well as equality and inequality constraints we applied the CVXOPT function *cvxopt.solvers.qp*. The experimental framework and all figures were generated with *Python* 2.7[4], *Jupyter Notebook* [Kluyver et al., 2016] and *Matplotlib* [Hunter, 2007].

We considered affinity prediction as co-regularised learning with few labelled and many unlabelled data instances. We randomly split each dataset of annotated ligands into a labelled and an unlabelled fraction. The co-regularised algorithms employ both the labelled and the unlabelled part for training. More precisely, in addition to labelled examples the co-regularised algorithms have access to the entire set of unlabelled instances without labels. The considered baselines without co-regularisation term are only aware of the labelled examples for training. For all algorithms the unlabelled data fraction is used for testing in the sense of a transductive scenario (see Section 2.2). We decided against a completely independent test sample because of the small fraction of labelled examples and the fact that technically all possible molecular compounds are known in advance in real-world applications as well.

Due to the small number of labelled examples n and large number of unlabelled instances m, we had to modify the grid search scheme for the hyperparameter tuning procedure as well. The standard k-fold CV scheme was introduced in Section 2.3.3 above. Assume N is the number of instances of a dataset. Firstly, for training and testing we performed a k-fold *inverse cross-validation* similar to the performance evaluation of CoRLSR by Brefeld et al. [2006]. We chose a fraction p of randomly drawn examples as labelled examples ($n = p \cdot N$) and the remaining examples as unlabelled instances for training and testing ($m = (1-p) \cdot N$). Secondly, for the hyperparameter search we performed another k'-fold inverse CV with a randomly drawn fraction p' of the labelled examples and the unlabelled instances ($n' = p' \cdot n$, $m' = p' \cdot m$). In contrast to the procedure in the empirical section of Chapter 3, the m' unlabelled instances for the hyperparameter tuning were drawn from the n unlabelled instances of the respective fold as we assumed them to be known (without labels). The reasoning behind was to provide sufficient labelled

[1] `https://www.python.org/`
[2] `https://bitbucket.org/Michael_Kamp/cosvr`
[3] `http://cvxopt.org/`
[4] `https://www.python.org/`

training examples n' by choosing p' close to 1.0. The performances of the different algorithms were calculated using 5-fold inverse CV ($p = 0.3$). The hyperparameters for each approach and for each dataset were optimised using grid search with 3-fold CV. The utilised hyperparameter grid was $\nu_v \in \{10^{-4}, 10^{-3} \ldots, 10^{3}\}$, $\lambda \in \{10^{-2}, 10^{-1} \ldots, 10^{4}\}$, and $\varepsilon^L, \varepsilon^U \in \{2^{-3}, 2^{-2} \ldots, 2^{0}\}$.

As we dealt with a regression task we used the RMSE from Section 2.2 as evaluation measure. In order to assess whether the RMSEs of method A were significantly greater or smaller than the RMSEs of another method B, we applied *Wilcoxon signed-rank tests*. The null hypothesis H_0 of this non-parametric test was that the median difference between pairs of measurements be zero[5]. The alternative hypothesis H_1 stated the median difference to be either greater or smaller than zero, i.e., either method A or method B had significantly smaller RMSEs than the respective other method.

We compared the CoSVR variants ε-CoSVR, ℓ_2-CoSVR, and Σ-CoSVR with CoRLSR and other single- and multi-view algorithms for regression utilising a linear kernel for the proposed kernel methods. A list of the reported algorithms in our empirical analysis can be found in Table 4.3. We point out that the baseline CoRLSR is a multi-view co-regularised method just like the CoSVR variants. For the practical experiments we applied the modified versions ε-CoSVR$_{\text{mod}}$ and ℓ_2-CoSVR$_{\text{mod}}$ due to their shorter running time in comparison to base CoSVR which has a greater number of variables (see Table 4.2). Pairwise Wilcoxon signed-rank tests showed that the modified versions do not have greater RMSEs than the base versions, although the unlabelled error term of the modified algorithms is formulated differently (compare the definitions in Sections 4.3.2). In addition to the co-regularised algorithms we considered the corresponding single-view *SVR (v)* and *RLSR (v)* methods (where v refers to the molecular fingerprint or view) as well as the canonical multi-view baselines *SVR (concat)* and *RLSR (concat)* (where *concat* refers to the concatenated features of the involved molecular fingerprints or views). Finally, for reasons of completeness we also report RMSEs for *SVR (avg)* and *SVR (best)*. The first of which outputs the average RMSE of the independently trained single-view SVR (v) predictors. The latter is an oracle that reports the best result of all single-view SVR (v) predictors for each dataset.

4.4.2 Results

The presentation of practical results is divided into two parts. In part A) we evaluate co-regularisation variants using combinations of standard molecular fingerprints. In part B) we analyse the CoSVR performance with toy data that is supposed to imitate real-world data and systematically varies structural properties of it.

A) Co-Regularisation Experiment with Combinations of Standard Molecular Fingerprints

In Figure 4.3 we present the results of the CoSVR variants ε-CoSVR (a)-(j), ℓ_2-CoSVR (b)-(k), and Σ-CoSVR (c)-(l). We compare them to the results of CoRLSR (a)-(c), SVR (concat) (d)-(f), SVR (best) (g)-(i), and SVR (avg) (j)-(l). The scatter plots of Figure 4.3 show the RMSE values of the respective opposed algorithms. In each scatter plot we aggregated the RMSE results for all combinations of two fingerprints GpiDAPH3/ECFP4, Maccs/GpiDAPH3, and Maccs/ECFP4, as well as the combination

[5] `http://www.biostathandbook.com`

FIGURE 4.3: Performance comparison of CoSVR variants and baselines

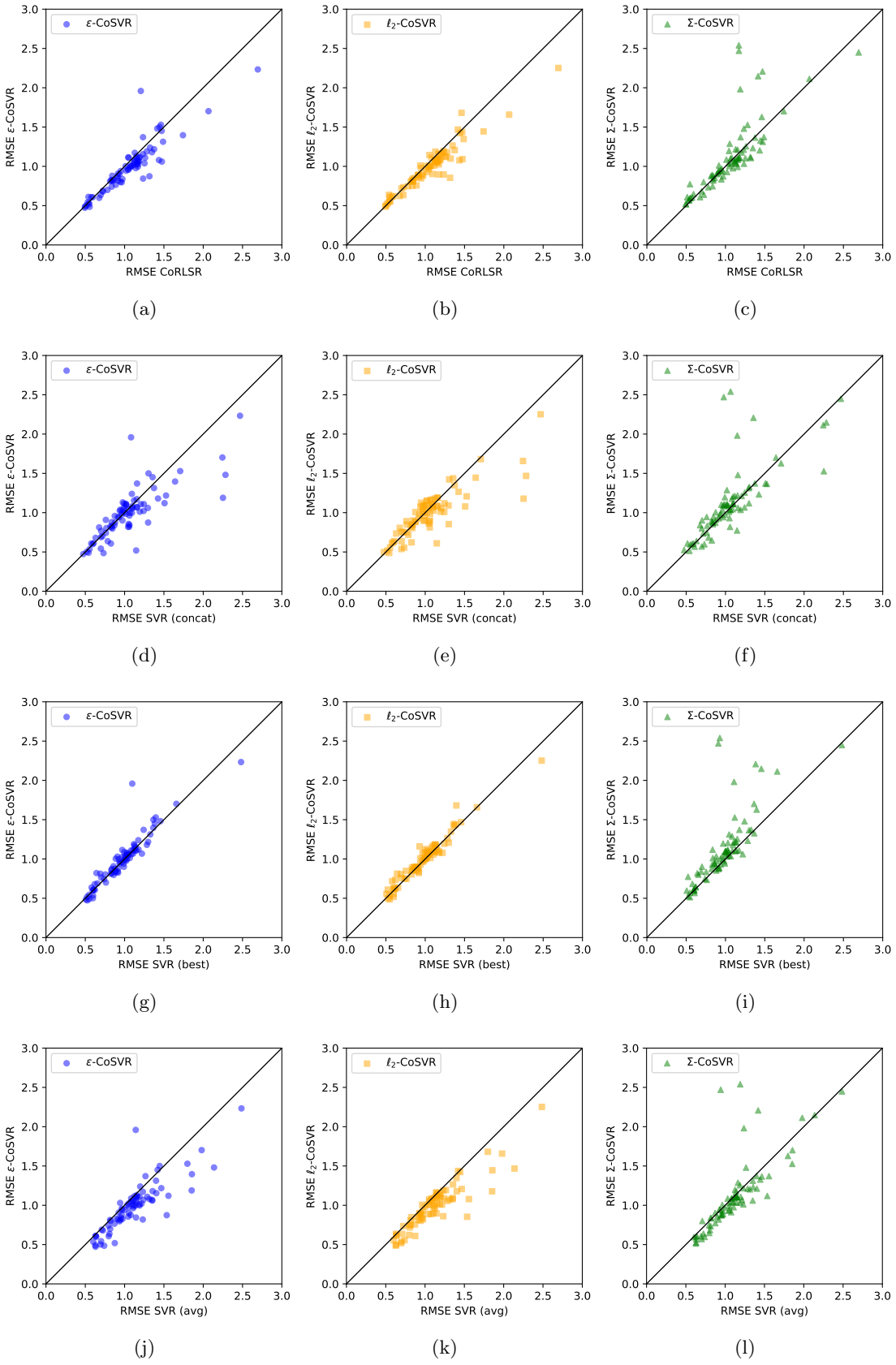
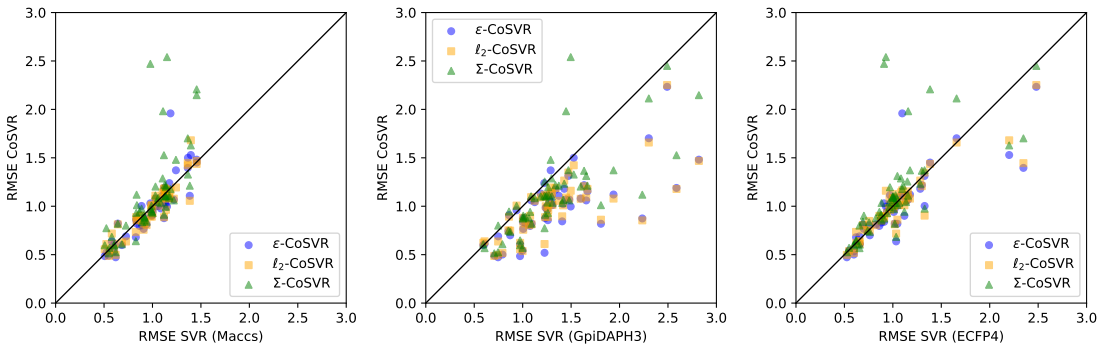


TABLE 4.3: List of single-view and multi-view methods

CoSVR Variants	**Description**
ε-CoSVR	multi-view modified ε-CoSVR$_{\text{mod}}$, see Section 4.3.2
ℓ_2-CoSVR	multi-view modified ℓ_2-CoSVR$_{\text{mod}}$, see Section 4.3.2
Σ-CoSVR	2-view CoSVR with fused kernel, see Section 4.3.3
Baselines	**Description**
CoRLSR	multi-view RLSR, see Section 4.2
RLSR (v)	single-view RLSR for view v, see Section 2.6.1
SVR (v)	single-view SVR for view v, see Section 2.6.2
RLSR (concat)	RLSR with multiple concatenated views
SVR (concat)	SVR with multiple concatenated views
Others	**Description**
SVR (best)	multiple single-view SVR, best prediction
SVR (avg)	multiple single-view SVR, average of predictions

of all three fingerprints Maccs/GpiDAPH3/ECFP4 to make the performance comparison more comprehensive with respect to views. Each point represents the RMSEs of two algorithms for one of the 22 dataset (for the three-view combination Maccs/GpiDAPH3/ECFP4 we only calculated the results for the 14 smallest datasets). The figures indicate that all CoSVR variants outperform CoRLSR, SVR (concat), and SVR (avg) for the majority of datasets. The performance advantage of CoSVR methods in comparison to the baselines grows from Σ-CoSVR over ℓ_2-CoSVR up to ε-CoSVR. Summarised over all fingerprint combinations and datasets, SVR (best) performs nearly equal to the variants ε-CoSVR and ℓ_2-CoSVR and slightly better than Σ-CoSVR. The results will be discussed in Section 4.4.3 below.

Additionally, we illustrate the benefit of using CoSVR variants compared to each of the considered single-view SVR (v) predictors. To this aim, the composition of the scatter plots in Figure 4.4 is different from the one of Figure 4.3. The circles, squares, and



(a) (b) (c)

FIGURE 4.4: Comparison of CoSVR variants with single-view SVR (v)

triangles show the RMSEs of ε-CoSVR, ℓ_2-CoSVR, and Σ-CoSVR on the y-axis compared to single-view SVR (v) on the x-axis (SVR (Maccs) in (a), SVR (GpiDAPH3) in (b), SVR (ECFP4) in (c)). In each diagram we unified different fingerprint combinations for the comparison with the single-view method, namely the ones that included the respective view. For example, the combinations Maccs/GpiDAPH3, Maccs/ECFP4, and Maccs/GpiDAPH3/ECFP4 were utilised for the comparison with SVR (Maccs) in Figure 4.4 (a). The plots show that CoSVR outperforms SVR (GpiDAPH3) and SVR (Maccs) for the majority of comparisons. The performance comparison of CoSVR with SVR (ECFP4) becomes obvious from the Tables 4.4 and 4.5 below.

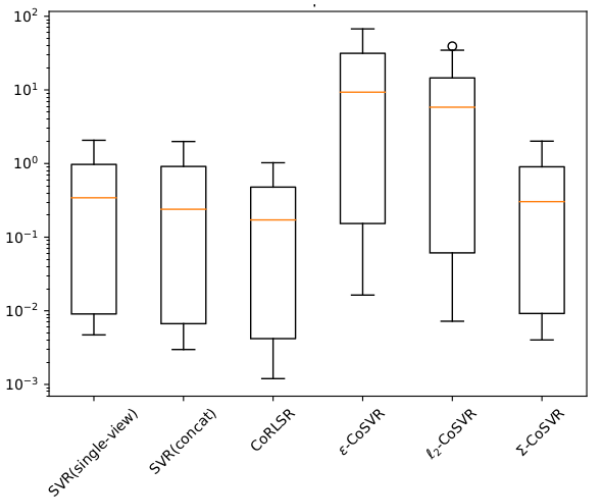
The indications with respect to ε-CoSVR and the single-view baselines SVR (v) as well as the multi-view baselines CoRLSR and SVR (concat) in Figures 4.3 and 4.4 are substantiated by a Wilcoxon signed-rank test on the results which we present in Table 4.4). In this table, we report the test statistics (Z and p-value) itemised with respect to the combinations of views. The results for combination Maccs/GpiDAPH3/ECFP4 should be considered in view of the fact that the sample size for a Wilcoxon signed-rank test should be greater than 20. Results in which ε-CoSVR statistically significantly outperforms the baselines with a significance level $p < 0.05$ are marked in with (+). Results for which we should remain with the null hypothesis (see above) are marked with ($\pm$). The test confirms that ε-CoSVR performs statistically significantly better than all single- and multi-view baselines for the fingerprint combinations GpiDAPH3/ECFP4 and better than CoRLSR for all view combinations. The test outcomes vary between (+) and ($\pm$) for the baseline SVR (concat) which will be discussed below. ε-CoSVR outperforms SVR (GpiDAPH3) independent of the fingerprint combination used for it. The advantage of ε-CoSVR against SVR (Maccs) and SVR (ECFP4) is significant for the comparison with fingerprint combination GpiDAPH3/ECFP4 and Maccs/GpiDAPH3/ECFP4, respectively.

In Table 4.5 we report the average RMSEs of all CoSVR variants and all methods listed in Table 4.3 for all two- and three-view combinations of the fingerprints Maccs, GpiDAPH3, and ECFP4. In terms of average RMSE, ε-CoSVR and ℓ_2-CoSVR outperform all other approaches for the view combination GpiDAPH3/ECFP4. For the fingerprint combinations Maccs/GpiDAPH3 and Maccs/GpiDAPH3/ECFP4 ε-CoSVR and ℓ_2-CoSVR outperform all other algorithms but the oracle SVR (best). With the exception of the comparison of SVR (Maccs) and ℓ_2-CoSVR, the methods ε-CoSVR and ℓ_2-CoSVR always have lower RMSEs than all other single-view SVR (v) approaches. We observe that the combination Maccs/ECFP4 results in the highest average RMSEs for CoSVR approaches. The performance of CoSVR approaches increases from Σ-CoSVR over ℓ_2-CoSVR to ε-CoSVR. The absolute RMSE values are smaller if three views instead of two views are used for learning. Apparently, the RMSE results for ℓ_2-CoSVR and ε-CoSVR are better if three views are used instead of two views. Note that SVR (best) is only a hypothetical baseline, since the best view varies between datasets and is thus unknown in advance. The algorithm Σ-CoSVR performs on average similar to the CoRLSR and SVR (concat) baseline and slightly worse than SVR (best). To avoid confusion about the different performances of Σ-CoSVR and ℓ_2-CoSVR, we point out that Σ-CoSVR equals ℓ_2-CoSVR$^{\text{mod}}$ (see Lemma 4.14) and not ℓ_2-CoSVR (equivalent with ℓ_2-CoSVR$_{\text{mod}}$ for $M = 2$) which we use for our experiments. The advantage in learning performance of ε-CoSVR and ℓ_2-CoSVR is accompanied with a longer running time as shown in Figure 4.5, where we compared the running time of the CoSVR variants with the ones of CoRLSR, SVR (concat), and single-view SVR for all combinations of two molecular fingerprints [Ullrich et al., 2017]. We averaged the results for SVR (ECFP4),

TABLE 4.4: Wilcoxon signed-rank test comparison of ε-CoSVR with baselines

Baseline	**Z**	**p-Value**	**Z**	**p-Value**
View Combination	GpiDAPH3/ ECFP4		Maccs/ GpiDAPH3	
CoRLSR	9.0	< 0.00014 (+)	54.0	< 0.01858 (+)
SVR (Maccs)	-	-	118.0	< 0.78260 ($\pm$)
SVR (GpiDAPH3)	1.0	< 0.00005 (+)	20.0	< 0.00055 (+)
SVR (ECFP4)	39.5	< 0.00473 (+)	-	-
SVR (concat)	3.0	< 0.00006 (+)	107.0	< 0.52660 ($\pm$)
Baseline	**Z**	**p-Value**	**Z**	**p-Value**
View Combination	Maccs/ ECFP4		Maccs/ GpiDAPH3/ECFP4	
CoRLSR	40.0	< 0.00498 (+)	16.0	< 0.02194 (+)
SVR (Maccs)	85.0	< 0.18310 ($\pm$)	13.0	< 0.01315 (+)
SVR (GpiDAPH3)	-	-	0.0	< 0.00098 (+)
SVR (ECFP4)	101.0	< 0.40770 ($\pm$)	48.0	< 0.77760 ($\pm$)
SVR (concat)	91.0	< 0.24910 ($\pm$)	33.0	< 0.22090 ($\pm$)

FIGURE 4.5: Average running times (logarithmic scale) of the CoSVR variants, CoRLSR, SVR (concat) and SVR



SVR (GpiDAPH3), and SVR (Maccs) and summarised it with *SVR (single-view)*. For a fair comparison with respect to running times, we point to the fact that we used our own implementation for all presented approaches. In accordance with the theory presented in Section 4.3.3, Σ-CoSVR exhibits a running very close to the single-view SVR approaches.

B) Co-Regularisation Experiments with Synthetic Data

The figures and tables above show that the CoSVR variants often outperform baselines on real-world ligand prediction datasets. In order to understand how CoSVR succeeds to achieve better prediction results than single-view baselines and, whether there are structural properties or barriers for the beneficial application of co-regularised algorithms, we add experiments with synthetic data. This synthetic data was generated with the aim to imitate structural properties of the real-world datasets. More precisely, we systematically varied structural parameters in order to expose the resulting consequences of the data structure for the ligand prediction performance of co-regularised algorithms.

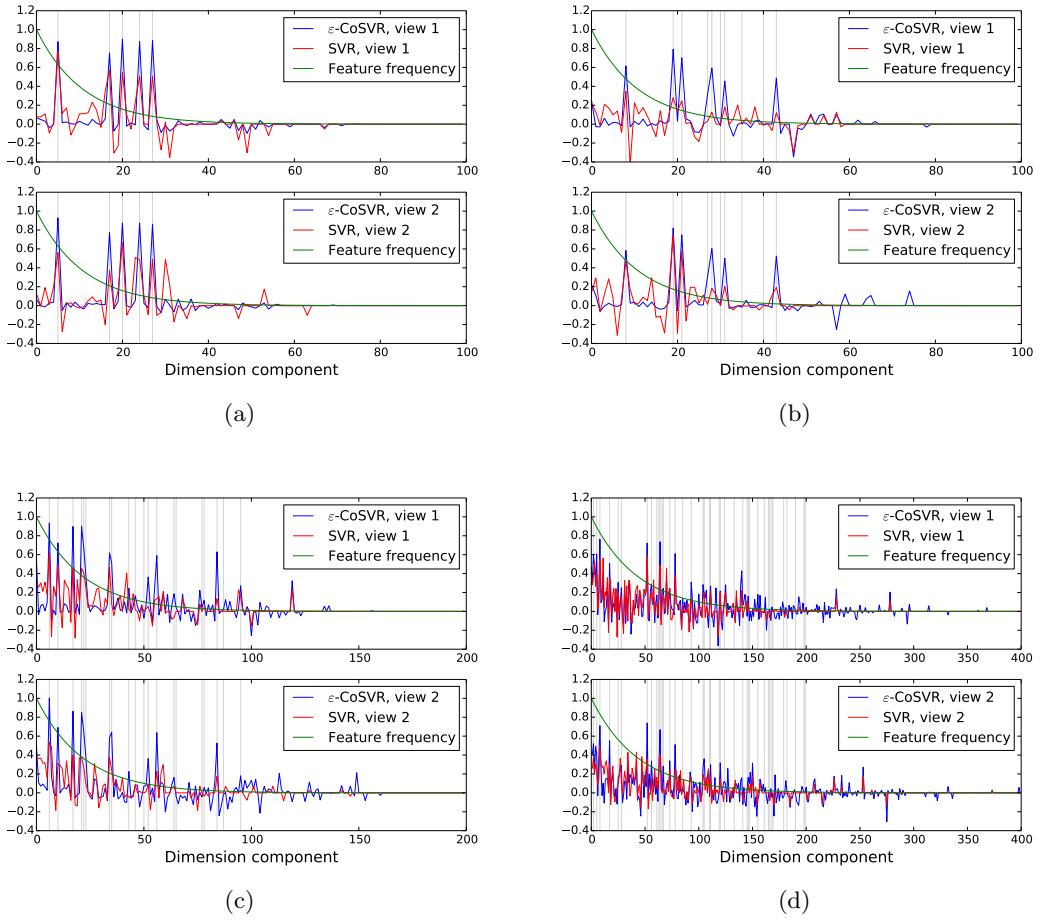
All data representations are binary, high-dimensional, and sparse. With *sparse* we refer to the fact that most of the features are equal to zero (compare also Section 4.3.4). We consider the frequency of a feature as the number of 1's in the respective component column divided by the total number of ligands in the dataset. A typical trend of feature frequencies is shown in Figure 4.8 for dataset P07858, where the components of the feature vector dimension on the x-axis are ordered by decreasing frequency and the dimension d is different for the considered fingerprints (see also Table B.2 in the appendix). The features seem to follow a characteristic exponential decay, which is less distinct for fingerprint Maccs because of its much smaller dimension compared to ECFP4 and GpiDAPH3.

We randomly generated binary datasets of 200 examples and different feature dimensions d (see Table 4.6) and a very small fraction of 0.1 of labelled examples. For the generation of two views, denoted with *view 1* and *view 2*, we independently drew *Bernoulli random variables* with an exponentially decreasing probability of success in order to mimic the feature frequency trend in the ligand datasets (see Figure 4.8). These frequencies are represented by the green lines in Figure 4.6 (a), (b), (c), and (d). From the rather frequent components $1, 2, \ldots, d/2$ we randomly chose a number of f relevant feature

Method	**View Combinations**			
	GpiDAPH3/ ECFP4	Maccs/ GpiDAPH3	Maccs/ ECFP4	Maccs/ GpiDAPH3/ECFP4
ε-CoSVR	1.031	0.993	1.014	0.873
ℓ_2-CoSVR	1.044	0.989	1.000	0.888
Σ-CoSVR	1.136	1.084	1.109	-
CoRLSR	1.174	1.052	1.045	0.938
SVR (Maccs)	-	0.996	1.011	0.950
SVR (GpiDAPH3)	1.333	1.358	-	1.206
SVR (ECFP4)	1.069	-	1.084	0.892
SVR (concat)	1.179	1.106	0.973	0.913
SVR (best)	1.067	0.981	0.948	0.846
SVR (avg)	1.201	1.177	1.048	1.016
RLSR (Maccs)	-	1.002	1.013	0.932
RLSR (GpiDAPH3)	1.327	1.347	-	1.202
RLSR (ECFP4)	1.066	-	1.096	0.898
RLSR (concat)	1.159	1.105	0.975	0.904

Table 4.5: Average RMSEs for all methods and fingerprints

FIGURE 4.6: Comparison of feature weights for toy experiment



columns (compare Table 4.6) and changed these features in view 2 by transferring them from view 1. Consequently, both views exhibited the same relevant feature columns (grey vertical lines 4.6). Finally, the label was calculated as sum of the relevant feature columns. We calculated the RMSEs of ε-CoSVR, single-view SVR (view 1), and single-view SVR (view 2) with default parameters and plotted the feature weights of the two view-predictors corresponding to ε-CoSVR (blue graphs) as well as the single-view SVR predictors SVR (view 1) and SVR (view 2) (red graphs) against the respective dimension component on the x-axis. We joined the data points with a continuous graph in order to emphasise the highly weighted relevant features at the grey lines (big amplitudes of blue peaks, smaller amplitudes of red peaks). On the majority, we observe non-negative

Subfigure	**Dimension**	**Features**	**RMSE**		
	d	f	SVR (view 1)	SVR (view 2)	ε-CoSVR
4.6 (a)	100	5	0.467	0.491	0.128
4.6 (b)	100	10	0.841	0.751	0.461
4.6 (c)	200	20	1.160	1.205	0.756
4.6 (d)	400	40	1.961	2.035	1.669

TABLE 4.6: Input parameters for synthetic datasets and RMSE results

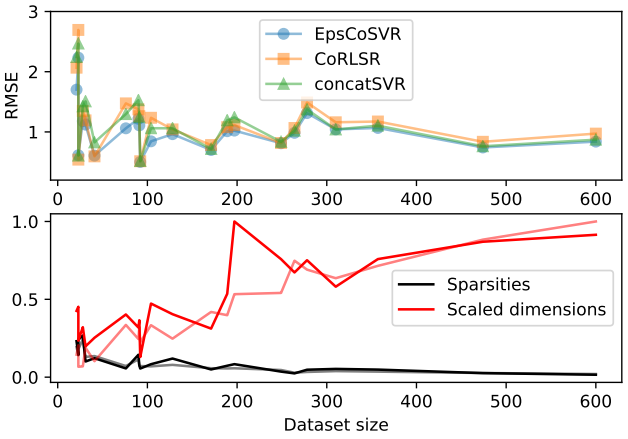
weights as the algorithm approximates the actual functional relation between input and output which is a sum of 1's. Figure 4.6 (a) shows that the view-predictors of ε-CoSVR are able to extract the relevant features and weight them with a big feature coefficient value, whereas the single-view SVR predictors weight every feature with a comparatively small coefficient value. This effect diminishes with increasing dimension d and fraction of relevant features f which we infer from the trend in Figures 4.6 (a) to Figure 4.6 (d). According to the results in Table 4.6, ε-CoSVR always performs much better than single-view SVR in terms of RMSEs.

4.4.3 Discussion

CoSVR regression techniques are able to take profit from the inherent information of unlabelled instances delivered in form of multiple sparse data representations. In view of the results presented above, they generally perform better than single-view SVR (v), and RLSR (v) approaches, which can be assumed the state-of-the-art method in affinity prediction and strongly related tasks [Bock and Gough, 2002, Liu et al., 2006, Maunz and Helma, 2008, Ding et al., 2013, Sugaya, 2014, Balfer and Bajorath, 2015, Abbasi et al., 2017]. More precisely, according to Table 4.5, ε-CoSVR and ℓ_2-CoSVR outperform the multi-view approach CoRLSR [Brefeld et al., 2006] on all view combinations and SVR (concat) on 3 out of 4 view combinations. ℓ_2-CoSVR outperforms all single-view SVR approaches on all view combinations and, ε-CoSVR on 3 out of 4 view combinations. These conclusions are refined via Wilcoxon signed-rank tests (see Table 4.4). Differences in the pairwise comparison of precise fingerprint combinations here arise from the fact that Table 4.5 contains absolute values and Table 4.4 delivers information on the difference in performance of two algorithms. Furthermore, ε-CoSVR and ℓ_2-CoSVR outperform SVR (best) for the view combination GpiDAPH3/ECFP4 and perform comparably for all other combinations. The performance of SVR (best) in comparison with CoSVR and the baselines is not surprising as the algorithm picks the best single-view SVR (v) predictor for each dataset individually. We did not expect CoSVR to beat the oracle SVR (best) as it represents an unrealistic baseline. Instead, we showed that the multi-view algorithms ε-CoSVR and ℓ_2-CoSVR perform automatically as good as the best single-view predictor. Because of the weaker performance of Σ-CoSVR, this variant is rather from a theoretical importance (see Sections 4.3.3, 4.3.4, and 4.3.5). To avoid confusion about the different performances of Σ-CoSVR and ℓ_2-CoSVR, we point out that Σ-CoSVR equals ℓ_2-CoSVR$^{\text{mod}}$ according to Lemma 4.14. It is not equal to ℓ_2-CoSVR which we use for our experiments and which is equivalent with ℓ_2-CoSVR$_{\text{mod}}$ for $M = 2$. Generally, the absolute RMSEs of the CoSVR variants decrease if 3 molecular fingerprints are utilised compared to a combination of 2 fingerprints.

A typical ligand affinity dataset using a binary molecular fingerprint is sparse. This is a consequence of the fact that fingerprints usually gather a large number of molecular features, of which finally a potential ligand only exhibits a few. From Figure 4.6 we conclude that ε-CoSVR detects the impact of the relevant features better than single-view SVR. Figures 4.6 (a) and (b) show that the view-predictors of ε-CoSVR provide the relevant features (grey line) with a particular large coefficient, whereas the SVR (view 1) and SVR (view 2) predictors assign a large weight to all of the frequent features (dimension between 1 and $d/2$). This effect can be explained with the feature information contained in unlabelled data in combination with the sparsity and the high dimensionality of the data in general. As the fraction of labelled examples in comparison to the one for unlabelled examples is small and because all features have a low frequency, ε-CoSVR has a

FIGURE 4.7: RMSE performance (top), scaled true dataset dimensions, and sparsities (bottom) for the fingerprint combination GpiDAPH3/ECFP4



much better chance to detect relevant features for the prediction in the unlabelled data that do not appear in the labelled data. In general, the RMSEs of all algorithms rise for increasing true dataset dimension (for the term *true dimension* compare Appendix B). In Figure 4.7 we plotted the scaled true dimensions, the sparsities, and the RMSE for ε-CoSVR, CoRLSR, and SVR (concat) against the number of ligands in the datasets. We observe that the results become worse if the dimension grows disproportional with respect to the dataset size and, hence, the number of features exceeds the number of training examples.

A general drawback of ε-CoSVR and ℓ_2-CoSVR is that high expenses are necessary to solve the corresponding optimisation problems which results in longer running times compared to single-view algorithms (see Table 4.2 and Figure 4.5). For this reason, CoSVR should not be preferred if predictions need to be delivered immediately. In contrast, Σ-CoSVR has the running time of a single-view algorithm. The choice of the algorithm's optimal hyperparameters has not only a strong impact on computing time but also for the respective predicition performance. To be more precise, for ε-CoSVR $M+3$ hyperparameters $\nu_1, \ldots, \nu_M, \lambda, \varepsilon^L$, and ε^U have to be tuned, which is costly if the parameter grid is close meshed. In contrast, for SVR only 2 optimal hyperparameters

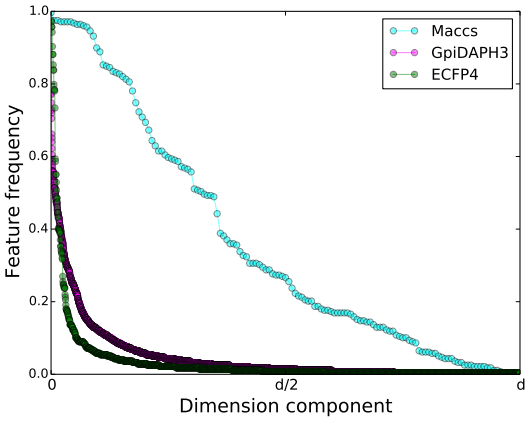


FIGURE 4.8: Feature frequency trend for the considered fingerprints

ν and ε have to be calculated. Therefore, a compromise between optimal parameter assignment and the resulting time complexity must be found in this context.

In summary, CoSVR and its variants are novel regression techniques using multiple views and unlabelled data with a high predictive performance. If computation capacity is not a limiting factor, the application of CoSVR supersedes the expensive choice of the most appropriate data representation for the learning task at hand.

Chapter 5

Projection-Based Learning

In the learning scenarios of the previous chapters labelled data was available for the generation of a regression model for ligand affinities. Multiple views on data were applied to find a supervised MKL model based on a linear combination of predictor functions and sufficient labelled training information in Chapter 3. The semi-supervised MVL approach of co-regularisation was used in Chapter 4 to obtain a good predictor for regression even if only little labelled training data was available in addition to a large number of unlabelled instances. The variety of vectorial fingerprint formats for small molecular compounds represent the multiple views on data in the previous chapters. Without further assumptions or practical experiments, it is not known which molecular fingerprint is optimal for the ligand affinity prediction problem at hand. In Chapter 5, we investigate the prediction of affinities provided the special case that no labelled examples are known for a considered target protein at all. The search for novel ligands of so-called *orphan proteins* is denoted with *orphan screening* and an instance of *unsupervised learning*. The regression case of orphan screening is the determination of ligand affinities for the orphan protein. We observed in practical experiments that the regression error of ligand affinity prediction with supervised kernel methods increased drastically for decreasing number of labelled examples. Therefore, we investigate alternative techniques for the marginal case of orphan screening in the present chapter.

The classification case of orphan screening has already been addressed [Geppert et al., 2009, Wassermann et al., 2009a, Geppert et al., 2010]. In this context, the *target-ligand kernel approach* (TLK) [Erhan et al., 2006, Jacob et al., 2008, Jacob and Vert, 2008] is an effective baseline that utilises kernel functions for both targets and ligand instances. Bock and Gough [2002, 2005] predicted the free energy of a protein-ligand-complex and performed a ranking approach. Our solution to the described unsupervised problem requires to firstly consider the actual *learning task* relative to its corresponding *target* and, secondly, the existence of different targets for which this learning tasks can be solved. In the practical problem of orphan screening the learning task is to predict ligand affinities for the orphan protein. Besides the orphan target, the affinity prediction task can be considered equivalently for different proteins. More precisely, to compensate for the lack of data, labelled information of other targets as well as relations between the targets will be used to infer an appropriate prediction model for the orphan target. We introduce and evaluate *corresponding projections* (CP) for orphan screening [Ullrich and Gärtner, 2014, Giesselbach et al., 2018]. Additionally, we suggest *orphan principal component analysis* (OPCA) as a variant of the interactive knowledge-based PCA of

[Oglic et al., 2014]. Although we focus on ligand affinity prediction, the presented algorithms in the present chapters are suitable for learning problems with the following properties

- a principal learning task and targets for which the learning task can be considered,
- an orphan target,
- further non-orphan targets,
- (at least one) representation of data instances for the principal learning task with appropriate similarity measure (kernel function),
- and a similarity measure (kernel function) for targets.

Two examples from real-world show that the listed preconditions indeed describe realistic and relevant scenarios in practice.

Example 5.1. ***(Orphan screening)*** *Orphan screening denotes the search for binding partners in compound databases for proteins without previously known information on ligands and their affinities [Bock and Gough, 2005, Wassermann et al., 2009a, Ullrich et al., 2010, Giesselbach et al., 2018]. The respective proteins are called orphan proteins. Prominent examples of orphan targets are the human G-protein coupled receptors (GPCRs) for which hardly any binding partners are known [Jacob et al., 2008, Zhou and Skolnick, 2012]. Because of their regulatory role in biochemical pathways binding partners of GPCRs are of great value in drug discovery research. Although millions of small molecular compounds are identified and protein-ligand information is described in molecular databases, orphan proteins still exist as the number of functional proteins in biological organisms is large and can be a result of newly-discovered proteins.*

Example 5.2. ***(Paper rating)*** *Another application from the biomedical domain is the suggestion or evaluation of medicinal articles, for example from PubMed*[1]*, to find the most promising treatment. Every patient in a hospital or medical practice is represented via a health record that comprises information on physical parameters, age, pre-existing condition, and prior examination and therapy efforts. Having these records and the article's text document both a patient and article similarity can be calculated. Medical experts are able to score the relevancy of a scientific article (here used synonymously with treatment) for documented patients. Such an evaluation would be desirable for completely new patients at the beginning of the therapeutic treatment.*

For both CP and OPCA, projections play an important role for the actual knowledge transfer from targets with labelled information to targets without labelled information (compare also Section 2.7 on dimensionality reduction). Projection-based methods have already been applied in chemoinformatics, for example, Vert and Kanehisa [2002] used kernel canonical correlation analysis (CCA) to extract the most relevant features for a gene classification problem. Because of the inclusion of labelled information from other related targets, we do not face a conventional unsupervised scenario. However, this classification is still correct for the principal learning task with respect to the orphan target. The learning scenario and the proposed algorithms better fit in the classes of *transfer learning* [Pan and Yang, 2010] or *multi-task learning* [Caruana, 1997]. Transfer

[1] `https://www.ncbi.nlm.nih.gov/pubmed/`

learning comprises approaches that use model information of already solved problems for the solution of an unsolved but related problem. Multi-task learning aims at a simultaneous solution of different problems without a focus on one particular target. In a transfer learning approach, Ning et al. [2009] enriched the training information for the protein target under consideration with labelled data from related proteins. Though, the focused target itself was not an orphan target.

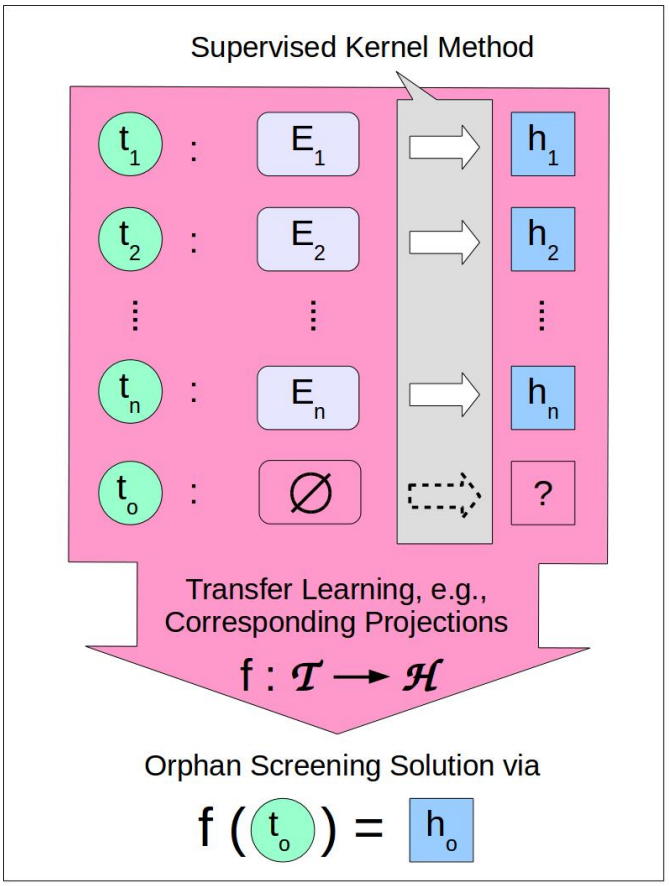
The classification of the approaches considered in the present chapter differs from the one in two previous chapters. Firstly, the two algorithms CP and OPCA are not restricted to an application for regression problems only. If the hypothesis model allows for an appropriate parameterisation, both algorithms can be applied to classification or structured output prediction as well. Furthermore, in contrast to the algorithms investigated in Chapters 3 and 4, the definitions of the algorithms in the present chapter do not essentially require a multi-view representation of data instances. Instead, we define CP and OPCA as single-view approaches and present additional multi-view variants. Mainly because of the knowledge transfer setting from targets with labelled training data to targets without labelled training data, a slightly modified notation will be used in Chapter 5 which will be introduced at the beginning of the following section.

The present chapter is based on our publication [Giesselbach et al., 2018]. It is structured as follows. Firstly, we introduce the learning scenario and the modified notation of the present chapter in Section 5.1. Secondly, in Section 5.2 we discuss the TLK baseline. The subsequent Section 5.3 is dedicated to the CP algorithms and its single- and multi-view variants. In the fourth empirical section, we evaluate the novel projection-based algorithm CP practically for the prediction problem of ligand affinity prediction and compare its performance with baseline methods. Finally, we introduce OPCA in an extended future work part in Section 5.5.

5.1 Orphan Screening Learning Scenario

As already mentioned above, the learning scenario investigated in the present chapter differs from the setting in Chapter 3 and 4. The modified scenario comes along with a slightly modified notation. In Chapter 5, with *principal learning task* we denote the search for a predictor function h from a hypothesis space $\mathcal{H}$ that maps instances from $\mathcal{X}$ to labels from $\mathcal{Y}$ analogous to the focused learning task in the previous main chapters. The principal learning task always corresponds to a particular target t from a target space $\mathcal{T}$ and, therefore, aims at a target-related hypothesis $h_t : \mathcal{X} \to \mathcal{Y}$. For ligand affinity prediction, the principal learning task is to find a binding affinity model h_t with respect to a protein target t. Superior to the principal learning task, we consider the learning problem to find a function $f : \mathcal{T} \to \mathcal{H}$ that assigns a binding model h_t to each target t. Although it is not a general requirement of transfer learning [Pan and Yang, 2010], we are interested in the case that both the instance space $\mathcal{X}$ and the label space $\mathcal{Y}$ as well as the principal learning task are the same for all targets. A target $t \in \mathcal{T}$ is called *supervised target* if there is labelled training data from $\mathcal{X} \times \mathcal{Y}$ to solve the principal learning task for t. The corresponding *supervised hypothesis* h_t can be found via an arbitrary supervised (kernel) method using the labelled examples for training. On the contrary, a target $t_o \in \mathcal{T}$ without labelled training information is called *orphan target* and, hence, to learn an *orphan hypothesis* $h_o \in \mathcal{H}$ for t_o is an unsupervised task.

FIGURE 5.1: Overview of the orphan screening's learning scenario



However, knowing f the orphan hypothesis h_o can be determined via

$$f(t_o) = h_o. \tag{5.1}$$

We will refer to the problem of finding a hypothesis for an orphan target as *orphan screening*, independent of whether we consider the practical problem of affinity prediction or another with the same preconditions on the learning scenario. An overview of the orphan screening learning scenario with principal and superordinate models can be found in Figure 5.1. In this figure, $t_1, \ldots, t_n \in \mathcal{T}$ denote the supervised targets whose so-called supervised hypotheses $h_1, \ldots, h_n \in \mathcal{H}$ can be learning with an arbitrary supervised kernel method. In contrast, t_o is the orphan target and h_o the orphan hypothesis that can be obtained via transfer learning. In Chapter 5, we will investigate kernelised algorithms again. Other than in Chapters 3 and 4, we will apply a kernel function $k_{\mathcal{T}}$ and a kernel function $k_{\mathcal{X}}$ as similarity measure for targets from $\mathcal{T}$ and instances from $\mathcal{X}$, respectively. Further changes with respect to notation will be explained below.

5.2 The Target-Ligand Kernel Approach

In chemoinformatics, orphan screening has been approached with the *target-ligand kernel* (TLK) algorithm which is not based on projections [Erhan et al., 2006, Jacob et al., 2008, Jacob and Vert, 2008, Wassermann et al., 2009a]. For the TLK algorithm, instances are

assumed to be pairs of targets $t \in \mathcal{T}$ and learning objects $x \in \mathcal{X}$ of the principal learning task according to

$$z = (t, x)$$

from the product space $\mathcal{Z} = \mathcal{T} \times \mathcal{X}$. We point to the fact that in the TLK scenario, z is not an unlabelled instance (as used in Chapter 4 or Section 5.3.4 below) but an element of $\mathcal{T} \times \mathcal{X}$. If kernel functions $k_\mathcal{T}$ and $k_\mathcal{X}$ for targets $t \in \mathcal{T}$ and instances $x \in \mathcal{X}$ exist, a product kernel for $z \in \mathcal{Z}$ can be defined as the product of target and object kernel

$$k_{TL}(z_1, z_2) = k_\mathcal{T}(t_1, t_2) \cdot k_\mathcal{X}(x_1, x_2) \tag{5.2}$$

for $z_1 = (t_1, x_1)$ and $z_2 = (t_2, x_2)$. According to the closure properties of kernels (compare Section 2.5) the *target-ligand kernel* k_{TL} from Equation 5.2 is a kernel function again. Training data for the TLK approach can be obtained by the original training examples combined with their respective supervised targets. Let each of the supervised targets $t_1, \ldots, t_n$ have example sets $E_1, \ldots, E_n \subseteq \mathcal{X} \times \mathcal{Y}$ of cardinality r which are supposed to be disjoint without loss of generality, i.e.,

$$\begin{aligned} E_1 &= \{(x_{11}, y_{11}), \ldots, (x_{1r}, y_{1r})\} \\ E_2 &= \{(x_{21}, y_{21}), \ldots, (x_{2r}, y_{2r})\} \\ &\vdots \qquad \vdots \qquad\qquad \vdots \\ E_n &= \{(x_{n1}, y_{n1}), \ldots, (x_{nr}, y_{nr})\}. \end{aligned}$$

For convenience, we assume the example sets to have equal size. With an appropriate renumbering of examples and corresponding supervised targets we obtain $N = n \cdot r$ labelled examples

$$(x_1, y_1), \ldots, (x_N, y_N) \in \mathcal{X} \times \mathcal{Y}. \tag{5.3}$$

in total for all supervised targets. They can be used to generate training examples from $\mathcal{Z} \times \mathcal{Y}$

$$(z_j, y_j) = ((t_i, x_{il}), y_{il}), \tag{5.4}$$

where $i = 1, \ldots, n$, $l = 1, \ldots, r$, and $j = 1, \ldots, N$. If we consider the renumbering of instances from $\mathcal{X}$ in Equation 5.3 the Gram matrix $K_\mathcal{X}$ of $k_\mathcal{X}$ is

$$K_\mathcal{X} = (k_\mathcal{X}(x_i, x_j))_{i,j=1}^{N}.$$

Finally, the TLK prediction model $f : \mathcal{Z} \to \mathcal{Y}$ can be calculated with any supervised kernel method, for example SVR or RLSR, utilising the described training data in Equation 5.4 and the kernel function k_{TL}. The Gram matrix $K_{TL} \in \mathbb{R}^{N \times N}$ of k_{TL} with respect to $z_1, \ldots, z_N$ can be written as

$$K_{TL} = K_\mathcal{X} \circ \begin{pmatrix} k_\mathcal{T}(t_1, t_1) \cdot \mathbf{1}_{r\times r} & k_\mathcal{T}(t_1, t_2) \cdot \mathbf{1}_{r\times r} & \ldots & k_\mathcal{T}(t_1, t_n) \cdot \mathbf{1}_{r\times r} \\ k_\mathcal{T}(t_2, t_1) \cdot \mathbf{1}_{r\times r} & k_\mathcal{T}(t_2, t_2) \cdot \mathbf{1}_{r\times r} & \ldots & k_\mathcal{T}(t_2, t_n) \cdot \mathbf{1}_{r\times r} \\ \vdots & \vdots & \ddots & \vdots \\ k_\mathcal{T}(t_n, t_1) \cdot \mathbf{1}_{r\times r} & k_\mathcal{T}(t_n, t_2) \cdot \mathbf{1}_{r\times r} & \ldots & k_\mathcal{T}(t_n, t_n) \cdot \mathbf{1}_{r\times r} \end{pmatrix},$$

where $N = n \cdot r$ and $\circ$ is the element-wise matrix product. A submatrix of $r \times r$ ones is represented by the symbol $\mathbf{1}_{r \times r}$. The orphan hypothesis $h_o : \mathcal{X} \to \mathcal{Y}$ can then be obtained via

$$h_o(x) = f((t_o, x)),$$

where $x \in \mathcal{X}$. In contrast to the CP and OPCA algorithms below, the training of the TLK model f does not require pre-trained supervised hypotheses $h_1, \ldots, h_n$. Furthermore, once the model f is available, it can be used to generate a binding model for arbitrary targets t via $h_t(\cdot) = f((t, \cdot))$ without a new optimisation step.

5.3 Corresponding Projections

The first novel projection-based approach to orphan screening reminds of an RRM framework according to Definition 2.5. Instead of an empirical risk objective as in Equation 2.8, we suggest to compare projections of targets and hypotheses in order to infer the best orphan hypothesis h_o. As explained in Section 5.1, the presented algorithm firstly lifts the initial prediction problem to a higher level and derives a model $f : \mathcal{T} \to \mathcal{H}$ that maps hypotheses to targets instead of labels to instances. Secondly, orphan screening is solved via CP and its variants by inserting t_o

$$h_o = f(t_o).$$

However, the model f can potentially output a hypothesis for any target $t \in \mathcal{T}$. We start with the definition and single-view variants of the algorithm and their solution. Subsequently, we suggest alternative algorithms that incorporate multiple views on data.

5.3.1 Similarity Transduction and Base Algorithm

Our aim is to learn a hypothesis $h_o : \mathcal{X} \to \mathcal{Y}$ for the orphan target $t_o \in \mathcal{T}$. For t_o we neither have training examples from $\mathcal{X} \times \mathcal{Y}$ nor other assumptions about an appropriate predictor. Nevertheless, we make use of information from the environment of the learning problem that will be helpful to solve orphan screening. Assume at first there are supervised targets $t_1, \ldots, t_n \in \mathcal{T}$ for which the corresponding hypotheses $h_1, \ldots h_n \in \mathcal{H}$ are already known. If labelled training pairs from $\mathcal{X} \times \mathcal{Y}$ are available for the supervised targets their hypotheses can be learned by any supervised (kernel) algorithm, for example SVR or RLSR. The respective training example sets $E_1, \ldots, E_n \subseteq \mathcal{X} \times \mathcal{Y}$ may or may not have identical instances. As already mentioned above, the idea is to find a model $f : \mathcal{T} \to \mathcal{H}$ that operates on a superior level and solves a supervised learning task with training examples $(t_1, h_1), \ldots, (t_n, h_n) \in \mathcal{T} \times \mathcal{H}$ such that the label space is the hypothesis space $\mathcal{H}$ of the principal learning task (see Section 5.1 and Figure 5.1). Let this function space $\mathcal{H}$ be a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ as introduced in Definition 2.15. Additionally suppose that there is a kernel function (or *similarity measure*) $k_{\mathcal{T}} : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ for targets. A good orphan hypothesis h_o can be achieved if we demand the corresponding projections to be approximately equal in the sense of

$$\frac{k_{\mathcal{T}}(t_i, t_o)}{\sqrt{k_{\mathcal{T}}(t_i, t_i)}} \approx \frac{\langle f(t_i), f(t_o) \rangle_{\mathcal{H}}}{\|f(t_i)\|_{\mathcal{H}}} = \frac{\langle h_i, h_o \rangle_{\mathcal{H}}}{\|h_i\|_{\mathcal{H}}} \tag{5.5}$$

for $i = 1, \ldots, n$. In terms of geometry we exploit the similarities in target and hypothesis space $\mathcal{T}$ and $\mathcal{H}$, respectively, to infer the orphan hypothesis. Consequently, we minimise a loss term for the left and the right hand side of Equation 5.5

$$\ell\left(k_{\mathcal{T}}(t_i, t_o)\|f(t_i)\|_{\mathcal{H}},\ \langle f(t_i), f(t_o)\rangle_{\mathcal{H}}\sqrt{k_{\mathcal{T}}(t_i, t_i)}\right) \tag{5.6}$$

for $i = 1, \ldots, n$, where $\ell : \mathbb{R}^2 \to \mathbb{R}^+$ is a loss function. We introduce the corresponding projections algorithm with ℓ being the squared loss ℓ_2 in the following definition.

Definition 5.1 (Corresponding projections)**.** [Giesselbach et al., 2018] We consider the principal learning task to find a hypothesis $h_t : \mathcal{X} \to \mathcal{Y}$ from a hypothesis space $\mathcal{H}$ with respect to a target $t \in \mathcal{T}$. Let $t_o \in \mathcal{T}$ be an orphan target and $(t_1, h_1), \ldots, (t_n, h_n) \in \mathcal{T} \times \mathcal{H}$ be examples of supervised targets and associated supervised hypotheses. Assume $k_{\mathcal{T}}$ is a kernel function for targets and $\mathcal{H}$ a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. The optimisation

$$f(t_o) = \underset{h \in \mathcal{H}}{\operatorname{argmin}}\ \nu\|h\|_{\mathcal{H}}^2 + \sum_{i=1}^{n}\left(\langle h, h_i\rangle_{\mathcal{H}}\sqrt{k_{\mathcal{T}}(t_i, t_i)} - k_{\mathcal{T}}(t_o, t_i)\|h_i\|_{\mathcal{H}}\right)^2, \tag{5.7}$$

where $\nu > 0$ is a hyperparameter, is called *corresponding projections* (CP) algorithm to solve orphan screening for the orphan target $t_o \in \mathcal{T}$.

The CP algorithm can be found as pseudocode in Appendix C. It is a step-wise description of all supervised and transfer learning tasks which are necessary to solve the orphan screening problem (compare Figure 5.1). Also other loss functions such as the ε-insensitive loss could be inserted for ℓ in Equation 5.6. We point out that the candidate space $\mathcal{H}$ can be chosen such that CP is applicable for other principal learning tasks like classification as well. Furthermore, there is no restriction on tasks $\mathcal{T}$ as well as the spaces $\mathcal{X}$ and $\mathcal{Y}$, apart from the existence of $k_{\mathcal{T}}$ and the Hilbert space property of $\mathcal{H}$. In the following two sections we discuss the cases of linear and non-linear hypothesis spaces $\mathcal{H}$ as well as a simplified CP algorithm [Giesselbach et al., 2018].

5.3.2 Linear and Simplified Algorithm

At first, we investigate the special case that $\mathcal{H}$ is the d-dimensional Euclidean space $\mathbb{R}^d$ with canonical inner product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\|$ and denote it *linear corresponding projections* (LCP). LCP is the starting point for further variants of the CP algorithm. Moreover, a linear hypothesis represents the baseline for ligand affinity prediction using molecular fingerprints (compare Section 1.3). For the sake of convenience we identify the orphan hypothesis $f(t_o) = \langle h_o, \cdot \rangle$ with its vector of coefficients $h_o \in \mathbb{R}^d$. We define the matrices $H \in \mathbb{R}^{d \times n}$ and $N \in \mathbb{R}^{n \times n}$ via

$$H = (h_1 \cdots h_n) \quad , \quad N = \operatorname{diag}(k_{\mathcal{T}}(t_i, t_i)), \tag{5.8}$$

as well as the vectors $\rho_o, \delta_o \in \mathbb{R}^n$ with

$$(\delta_o)_i = \delta_{oi} = k_{\mathcal{T}}(t_o, t_i)\|h_i\| \quad , \quad (\rho_o)_i = \sqrt{k_{\mathcal{T}}(t_i, t_i)}\delta_{oi}. \tag{5.9}$$

With $\operatorname{diag}(v)$ we name the diagonal matrix of a vector v. In contrast to its role in Equation 5.3 as example number, the symbol N in Equation 5.8 denotes a matrix. The following lemma represents a special case of Lemma 5.4 below.

Lemma 5.2. *Let $(t_1, h_1), \ldots, (t_n, h_n) \in \mathcal{T} \times \mathcal{H}$ be examples of targets and corresponding hypotheses, $k_\mathcal{H}$ be the linear kernel, $k_\mathcal{T}$ be an arbitrary kernel, and the hypothesis space $\mathcal{H} = \mathbb{R}^d$. Then LCP can be solved as*

$$f(t_o) = \left[\nu \mathbf{I}_d + \sum_{i=1}^{n} h_i k_\mathcal{T}(t_i, t_i) h_i^T\right]^\dagger \sum_{i=1}^{n} h_i \|h_i\| \sqrt{k_\mathcal{T}(t_i, t_i)} k_\mathcal{T}(t_o, t_i), \tag{5.10}$$

where $\nu > 0$ is a hyperparameter and † denotes the pseudoinverse of a matrix.

Proof. We reformulate the objective $\mathcal{Q}_o(h)$ in Equation 5.7 with the matrices and vectors defined in Equations 5.8 and 5.9 and receive

$$\mathcal{Q}_o(h) = \nu h^T h + h^T H N H^T h - 2h^T H \rho_o + {\delta_o}^T \delta_o.$$

The solution of LCP in Equation 5.10 can be derived by setting the gradient of $\mathcal{Q}_o(h)$

$$\frac{\partial \mathcal{Q}_o(h)}{\partial h} = 2\nu h + 2HNH^T h - 2H\rho_o$$

equal to zero. We obtain $h_o = [\nu \mathbf{I}_d + HNH^T]^\dagger H \rho_o$ which finishes the proof. □

As the matrix HNH from above is positive semi-definite, the inverse $[\nu \mathbf{I}_d + HNH]^{-1}$ always exists if ν is positive. Otherwise, the more general pseudoinverse can be applied. As a further variant we consider a simplified version of CP. For this simplified version we assume an arbitrary hypothesis space $\mathcal{H}$.

Definition 5.3. [Giesselbach et al., 2018] Let $(t_1, h_1), \ldots, (t_n, h_n) \in \mathcal{T} \times \mathcal{H}$ be supervised targets and corresponding supervised hypotheses. For an arbitrary kernel function $k_\mathcal{T}$ and hypothesis space $\mathcal{H}$ we define *simplified corresponding projections* (SCP)

$$f(t_o) = \sum_{i=1}^{n} h_i \frac{k_\mathcal{T}(t_o, t_i)}{\sqrt{k_\mathcal{T}(t_i, t_i)}} \tag{5.11}$$

for the orphan target $t_o \in \mathcal{T}$ as weighted sum of the supervised hypotheses.

Preliminary work on SCP classification has been published by Geppert et al. [2009] who applied a weighted sum of predictors denoted as *SVM linear combination* (SVM-LC). Using the SCP approach in Equation 5.11, the orphan hypothesis $h_o = f(t_o)$ can be determined simply as linear combination of the supervised hypotheses h_i. The corresponding linear coefficients can be calculated directly and have not to be optimised in form of model parameters. Therefore, the complexity of SCP is only $\mathcal{O}(nd\kappa)$ if the calculation cost for $k_\mathcal{T}$ can be bounded by κ. In contrast, the complexity for the LCP computation in Equation 5.10 is $\mathcal{O}(nd^3\kappa)$.

5.3.3 Non-Linear Corresponding Projections

In the previous section we considered a linear and a simplified version of CP. In the present section we exploit the Hilbert space property of the candidate space $\mathcal{H}$ with inner product $\langle \cdot, \cdot \rangle_\mathcal{H}$ and corresponding norm $\| \cdot \|_\mathcal{H}$ and derive two non-linear versions

of CP. Initially, we show that the orphan hypothesis h_o lies in the span of the supervised hypotheses $h_1, \ldots, h_n$, i.e., can be represented as linear combination

$$h_o = \sum_{i=1}^{n} \beta_{oi} h_i \quad , \quad \beta_o \in \mathbb{R}^n. \tag{5.12}$$

For the proof of Equation 5.12 we use a similar argument to the one applied in the proof of Theorem 2.21 [Schölkopf et al., 2001]. We consider the decomposition of $\mathcal{H}$ into $S = \text{span}\{h_1, \ldots, h_n\}$ and its orthogonal complement $S^\perp$. Consequently, the orphan hypothesis has got a representation $h_o = s + g$, where $s \in S$ and $g \in S^\perp$. We conclude for the objective of CP in Equation 5.7

$$\begin{aligned}
h_o &= \underset{s \in S, g \in S^\perp}{\text{argmin}} \ \nu \|s+g\|_{\mathcal{H}}^2 + \sum_{i=1}^{n} \left[\langle s+g, h_i \rangle_{\mathcal{H}} \sqrt{k_{\mathcal{T}}(t_i,t_i)} - k_{\mathcal{T}}(t_o,t_i) \|h_i\|_{\mathcal{H}} \right]^2 \\
h_o &= \underset{s \in S, g \in S^\perp}{\text{argmin}} \ \nu \|s\|_{\mathcal{H}}^2 + \|g\|_{\mathcal{H}}^2 + \sum_{i=1}^{n} \left[\langle s, h_i \rangle_{\mathcal{H}} \sqrt{k_{\mathcal{T}}(t_i,t_i)} - k_{\mathcal{T}}(t_o,t_i) \|h_i\|_{\mathcal{H}} \right]^2 \\
h_o &= \underset{s \in S}{\text{argmin}} \ \nu \|s\|_{\mathcal{H}}^2 + \sum_{i=1}^{n} \left[\langle s, h_i \rangle_{\mathcal{H}} \sqrt{k_{\mathcal{T}}(t_i,t_i)} - k_{\mathcal{T}}(t_o,t_i) \|h_i\|_{\mathcal{H}} \right]^2,
\end{aligned}$$

where we use that $g \perp \{s, h_1, \ldots, h_n\}$ and the objective is minimised if g is the zero element in $S^\perp$. Analogous to Equations 5.8 and 5.9, we define matrices $G, N \in \mathbb{R}^{n \times n}$

$$(G)_{i,j} = \langle h_i, h_j \rangle_{\mathcal{H}}, \quad N = \text{diag}(k_{\mathcal{T}}(t_i,t_i)) \tag{5.13}$$

and vectors $\rho_o, \delta_o \in \mathbb{R}^n$

$$(\delta_o)_i = k_{\mathcal{T}}(t_o,t_i)\|h_i\|_{\mathcal{H}}, \quad (\rho_o)_i = \sqrt{k_{\mathcal{T}}(t_i,t_i)}\ (\delta_o)_i. \tag{5.14}$$

For convenience, we identify h_o with its vector $\beta_o \in \mathbb{R}^n$ of coefficients in Equation 5.12.

Lemma 5.4. *[Giesselbach et al., 2018] Let $\mathcal{H}$ be a Hilbert space and $k_{\mathcal{T}} : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ be a kernel function for targets. Using the representation of h_o in Equation 5.12 the CP algorithm from Equation 5.7 can be solved as*

$$\beta_o = [\nu G + GNG]^\dagger G \rho_o, \tag{5.15}$$

where N, G, and ρ_o are the matrices and vectors from Equations 5.13 and 5.14. We call this version of CP non-linear corresponding projections (NLCP).

Proof. Using Equations 5.12, 5.13, and 5.14, the objective $\mathcal{Q}_o(h)$ in Equation 5.7 can be reformulated in terms of β

$$\mathcal{Q}_o(\beta) = \nu \beta^T G \beta + \beta^T GNG \beta - 2\beta^T G \rho_o + (\delta_o)^T \delta_o.$$

If we set the gradient $\partial \mathcal{Q}_o(\beta)/\partial\beta$ equal to zero, we obtain the desired result $\beta_o = [\nu G + GNG]^\dagger G \rho_o$. □

The computation of β_o in Equation 5.15 requires to invert an $n \times n$ matrix, which results in a computational complexity of $\mathcal{O}(n^3)$ for NLCP.

For NLCP we assumed that the supervised hypotheses $h_1, \ldots, h_n$ and the orphan hypothesis h_o are elements of the Hilbert space $\mathcal{H}$. Because of the existence of labelled training examples from $\mathcal{X} \times \mathcal{Y}$ for the supervised targets $t_1, \ldots, t_n$ the supervised hypotheses can be derived via arbitrary supervised kernel methods. However, we did not use further structural information about the hypotheses and the hypothesis space. In the following, we consider the renumbered union of training instances $x_1, \ldots, x_N \in \mathcal{X}$ contained in all training sets $E_1, \ldots, E_n$ presented in Equation 5.3. For the second non-linear CP variant we make two additional assumptions. Firstly, $\mathcal{H}$ is a RKHS with kernel k. Secondly, every supervised hypothesis h_i of target t_i, $i = 1, \ldots, n$, has got a representation

$$h_i(x) = \sum_{j=1}^{N} \pi_{ij} k(x_j, x). \tag{5.16}$$

If x_j was not in the training set E_i of t_i, the corresponding component π_{ij} of the vector of coefficients $\pi_i \in \mathbb{R}^N$ is equal to zero. In contrast to the notation in Chapter 4, the double index of π_{ij} refers to the index of the supervised target i and the numbering of instances j. A representation of hypotheses according to Equation 5.16 is a consequence of Theorem 2.21 for supervised kernel methods with well-posed optimisation problems, e.g., for SVR or RLSR. In real-world learning scenarios it is possible that the number of instances N is smaller than the number of supervised targets n. For example, this may happen in case every supervised target has got the same small labelled ligand training set and n is greater than the cardinality of this training set. For this case, the CP solution according to Equation 5.15 can be rewritten such that it can be solved in time $\mathcal{O}(N^3)$ where $N < n$. According to Equation 5.12, h_o can be represented as linear combination of supervised hypotheses

$$h_o(x) = \sum_{i=1}^{n} \beta_{oi} h_i(x).$$

Together with the representation of supervised hypotheses in Equation 5.16 we conclude

$$\begin{aligned} h_o(x) &= \sum_{i=1}^{n} \beta_{oi} \left(\sum_{j=1}^{N} \pi_{ij} k(x_j, x) \right) \\ &= \sum_{j=1}^{N} \left(\sum_{i=1}^{n} \beta_{oi} \pi_{ij} \right) k(x_j, x) = \sum_{j=1}^{N} \pi_{oj} k(x_j, x), \end{aligned} \tag{5.17}$$

where $\beta_o \in \mathbb{R}^n$ and $\pi_i, \pi_o \in \mathbb{R}^N$. Hence, with $\Pi = (\pi_1 \cdots \pi_n) \in \mathbb{R}^{N \times n}$ the coefficients of the orphan target are $\pi_o = \Pi \beta_o$. Let K be the Gram matrix of kernel k with respect to $x_1, \ldots, x_N$. Analogous to the derivation of the NLCP solution in Equation 5.15 we define the matrices $\tilde{G} \in \mathbb{R}^{N \times n}$ and $N \in \mathbb{R}^{n \times n}$

$$\tilde{G} = K\Pi, \quad \tilde{N} = \operatorname{diag}(k_{\mathcal{T}}(t_i, t_i)) \tag{5.18}$$

and vectors $\tilde{\rho}_o, \tilde{\delta}_o \in \mathbb{R}^n$

$$(\tilde{\delta_o})_i = k_{\mathcal{T}}(t_o, t_i) \sqrt{\pi_i^T K \pi_i}, \quad (\tilde{\rho}_o)_i = \sqrt{k_{\mathcal{T}}(t_i, t_i)}\ (\tilde{\delta}_o)_i. \tag{5.19}$$

Again, we identify h_o with its vector of coefficients π_o. We point to the fact that the subsequent Lemma 5.5 as well as all following lemmas and definitions are novel.

Lemma 5.5. *Let the supervised hypotheses $h_1, \ldots, h_n$, and the orphan hypothesis h_o have the representations from Equation 5.16 and 5.17, respectively, with $\pi_i, \pi_o \in \mathbb{R}^N$. With k we denote the reproducing kernel of $\mathcal{H}$ and with $k_\mathcal{T}$ a kernel function for targets from $\mathcal{T}$. If K is the Gram matrix of kernel k with respect to $x_1, \ldots, x_N$, the CP from Equation 5.7 can be solved as*

$$\pi_o = \left[\nu K + \sum_{i=1}^{n} K\pi_i k_\mathcal{T}(t_i, t_i)\pi_i^T K\right]^\dagger \cdot \sum_{i=1}^{n} \left(\sqrt{\pi_i^T K \pi_i}\sqrt{k_\mathcal{T}(t_i, t_i)} k_\mathcal{T}(t_o, t_i)\right) K\pi_i, \tag{5.20}$$

where $\nu > 0$ and $t_1, \ldots, t_n$ are the supervised targets. We call this approach kernel corresponding projections (KCP).

Proof. Given the preconditions of the lemma, the objective in Equation 5.7 can be parameterised as

$$\mathcal{Q}_o(\pi) = \nu\pi^T K\pi + \pi^T \tilde{G}\tilde{N}\tilde{G}^T\pi - 2\pi^T\tilde{G}\tilde{\rho}_o + \tilde{\delta}_o{}^T\tilde{\delta}_o.$$

We obtain the solution of KCP in Equation 5.20 if we put $\partial\mathcal{Q}_o/\partial\pi$ equal to zero. □

The computation of KCP in Equation 5.20 requires to invert a $N \times N$ matrix and has thus a complexity of $\mathcal{O}(N^3)$. For this reason, KCP should be preferred to NLCP if the hypothesis space is an RKHS, the number of targets is greater than the number of labelled training instances, and the supervised hypotheses have a representation as kernel linear combination as discussed above.

5.3.4 Multi-View Corresponding Projections

Analogous to SVR or RLSR, the definition of CP is not based on multiple representations of data. But equivalent to the co-regularisation approach in Chapter 4 for SVR or RLSR, the CP algorithm can be modified to a multi-view algorithm. We present two different multi-view versions of CP. Both versions make use of the supervised hypotheses for the optimisation step. For each supervised target t_i, $i = 1, \ldots, n$, and each view $v = 1, \ldots, M$ a supervised hypothesis $h_i^v \in \mathcal{H}_v$ can be generated via a supervised kernel method in a preliminary step of the CP optimisation from the labelled training data in E_i. In contrast to the single-view setting from above, the involved instances $x_1, \ldots, x_N$ are now available in M different data representations. The intuition behind the multi-view approach in Definition 5.6 is the following. Firstly, a term that reminds of the regularised empirical risk expands the idea of CP in Equation 5.7 to multiple views. Secondly, the inner products of supervised hypotheses and orphan hypothesis is supposed to be similar for pairs of views. For both multi-view algorithms in the subsequent Definition 5.6 and Definition 5.8 below, the multi-view version of the superordinate model f introduced in Equation 5.1 maps targets from $\mathcal{T}$ to M hypotheses from $\mathcal{H}_1 \times \cdots \times \mathcal{H}_M$. Every candidate space $\mathcal{H}_v$ is assumed to be a Hilbert space with inner product

$\langle \cdot, \cdot \rangle_{\mathcal{H}_v}$ and corresponding norm $\| \cdot \|_{\mathcal{H}_v}$. The multi-view solution of orphan screening can then be obtained via $f(t_o) = (h_o^1, \ldots, h_o^M)$.

Definition 5.6 (MVCP)**.** We consider the orphan target t_o and supervised targets $t_1, \ldots, t_n$. Suppose for each supervised target t_i a hypothesis h_i^v can be learned for every view v on data, i.e., there are pairs $(t_i, h_i^v) \in \mathcal{T} \times \mathcal{H}_v$ for $i = 1, \ldots, n$ and $v = 1, \ldots, M$. The optimisation $(h_o^1, \ldots, h_o^M) =$

$$\begin{aligned} \underset{h^v \in \mathcal{H}_v}{\operatorname{argmin}} \sum_{v=1}^{M} \left[\nu_v \|h^v\|_{\mathcal{H}_v}^2 + \sum_{i=1}^{n} \left(\langle h^v, h_i^v \rangle_{\mathcal{H}_v} \sqrt{k_{\mathcal{T}}(t_i, t_i)} - k_{\mathcal{T}}(t_o, t_i) \|h_i^v\|_{\mathcal{H}_v} \right)^2 \right] \\ + \lambda \sum_{u,v=1}^{M} \sum_{i=1}^{n} \left(\langle h^u, h_i^u \rangle_{\mathcal{H}_v} - \langle h^v, h_i^v \rangle_{\mathcal{H}_v} \right)^2 \end{aligned} \tag{5.21}$$

is called *multi-view corresponding projection* (MVCP), where $\lambda, \nu_v > 0$ are hyperparameters and the final predictor for the orphan target is the average $h_o = 1/M \sum_{v=1}^{M} h_o^v$.

The hyperparameters ν_v and λ play comparable roles for MVCP and the second multi-view CP algorithm below like in the definition of base CoSVR in Chapter 4. The proof for the solution of MVCP requires a number of further symbols analogous to the proofs of LCP, NLCP, and KCP above. We define the matrices $G_v, N, D_v \in \mathbb{R}^{n \times n}$

$$\begin{aligned} (G_v)_{i,j} &= \langle h_i^v, h_j^v \rangle_{\mathcal{H}_v}, \\ N &= \operatorname{diag}(k_{\mathcal{T}}(t_i, t_i)), \text{ and} \\ D_v &= \nu_v G_v + G_v N G_v + 2\lambda(M-1) G_v G_v, \end{aligned} \tag{5.22}$$

as well as the vector $\rho_o^v \in \mathbb{R}^n$

$$(\rho_o^v)_i = \sqrt{k_{\mathcal{T}}(t_i, t_i)} k_{\mathcal{T}}(t_o, t_i) \|h_i^v\|_{\mathcal{H}_v}. \tag{5.23}$$

Lemma 5.7. *Let G_v, N, D_v, and ρ_o^v be defined according to Equations 5.22 and 5.23. The solution $\beta_o^1, \ldots, \beta_o^M \in \mathbb{R}^n$ of MVCP can be obtained by the system of equations*

$$\begin{pmatrix} D_1 & -2\lambda G_1 G_2 & \ldots & -2\lambda G_1 G_M \\ -2\lambda G_2 G_1 & D_2 & \ldots & -2\lambda G_2 G_M \\ \vdots & \vdots & \ddots & \vdots \\ -2\lambda G_M G_1 & -2\lambda G_M G_2 & \ldots & D_M \end{pmatrix} \begin{pmatrix} \beta_o^1 \\ \beta_o^2 \\ \vdots \\ \beta_o^M \end{pmatrix} = \begin{pmatrix} G_1 \rho_o^1 \\ G_2 \rho_o^2 \\ \vdots \\ G_M \rho_o^M \end{pmatrix},$$

where β_o^v are the linear coefficients of the orphan hypothesis in view v

$$h_o^v = \sum_{i=1}^{n} (\beta_o^v)_i h_i^v$$

for $v = 1, \ldots, M$.

The proof for the solution of MVCP for orphan screening is analogous to the proof of the CoRLSR solution in Lemma 4.4.

Proof. For a fixed view on data v, a representation of h_o^v as linear combination of supervised hypotheses

$$h_o^v = \sum_{i=1}^{n} (\beta_o^v)_i h_i^v \tag{5.24}$$

can be proven analogous to the linear case in Equation 5.12, where $\beta_o^v \in \mathbb{R}^n$ are the linear coefficients. With the representation of h_o in Equation 5.24 the objective $\mathcal{Q}_o$ in Equation 5.21 can be parameterised with variables β_o^v for $v = 1, \ldots, M$. If we apply the definitions in Equations 5.22 and 5.23 the claim is a consequence of the gradient

$$\begin{aligned}\frac{\partial \mathcal{Q}_o(\beta_1, \ldots, \beta_M)}{\partial \beta_v} =& 2\nu G_v \beta_v + 2G_v N {G_v}^T \beta_v - 2G_v \rho_o^v \\ &+ 4\lambda(M-1) G_v {G_v}^T \beta_v - 4\lambda \sum_{u=1,\ldots,M}^{u \neq v} G_v {G_u}^T \beta_u\end{aligned}$$

put equal to zero. □

The second multi-view approach adopts the idea of co-regularisation from Chapter 4. For more details on co-regulariation compare Sections 4.2 and 4.3. Unlabelled data instances with respect to the principal learning task are used to compare the predictions of the orphan hypotheses in different views. Let $z_1, \ldots, z_m \in \mathcal{X}$ be unlabelled data instances. As we have different views on data, we consider M different kernel matrices K_v, $v = 1, \ldots, M$, with respect to the union of training instances $x_1, \ldots, x_N \in \mathcal{X}$, analogous to the Gram matrix definition in Lemma 5.5. In order to illustrate the analogies with Chapter 4 we will use the notation $L_v \in \mathbb{R}^{N \times N}$ for K_v. Equivalently,

$$U_v = \{k_v(x_i, x_j)\}_{i=1,\ j=N+1}^{N,\ N+m} \in \mathbb{R}^{N \times m}$$

denotes the view kernel matrix with respect to labelled and unlabelled instances, where $x_{N+1}, \ldots, x_{N+m} = z_1, \ldots, z_m$. As a difference to the approaches in Chapter 4, we consider the kernel expansion π_o^v and π_i^v, $v = 1, \ldots, M$, $i = 1, \ldots, n$, only with respect to the labelled instances of the principal learning task $x_1, \ldots, x_N$. Hence, $\pi_o^v, \pi_i^v \in \mathbb{R}^n$ holds true. We define the matrices $\tilde{G}_v \in \mathbb{R}^{N \times n}$, $\tilde{N} \in \mathbb{R}^{n \times n}$, and $D_v \in \mathbb{R}^{N \times N}$ as follows

$$\begin{aligned}\tilde{G}_v &= L_v \Pi_v, \\ \tilde{N} &= \operatorname{diag}(k_{\mathcal{T}}(t_i, t_i)), \text{ and} \\ D_v &= \nu_v L_v + \tilde{G}_v \tilde{N} \tilde{G}_v^T + 2\lambda(M-1) U_v^T U_v,\end{aligned} \tag{5.25}$$

as well as the vector $\tilde{\rho}_o^v \in \mathbb{R}^n$

$$\tilde{\rho}_o^v = \sqrt{k_{\mathcal{T}}(t_i, t_i)} k_{\mathcal{T}}(t_o, t_i) \sqrt{(\pi_i^v)^T L_v \pi_i^v}. \tag{5.26}$$

Definition 5.8 (CoCP)**.** Suppose for every view $v = 1, \ldots, M$ on data we have pairs $(t_i, h_i^v) \in \mathcal{T} \times \mathcal{H}_v$, $i = 1, \ldots, n$ of supervised targets and corresponding supervised hypotheses. Let $z_1, \ldots, z_m$ denote unlabelled instances with respect to the principal

learning task from $\mathcal{X}$. The optimisation $(h_o^1, \ldots, h_o^M) =$

$$\underset{h^v \in \mathcal{H}_v}{\operatorname{argmin}} \sum_{v=1}^{M} \left[\nu_v \|h^v\|_{\mathcal{H}_v}^2 + \sum_{i=1}^{n} \left(\langle h^v, h_i^v \rangle_{\mathcal{H}_v} \sqrt{k_{\mathcal{T}}(t_i, t_i)} - k_{\mathcal{T}}(t_o, t_i) \|h_i^v\|_{\mathcal{H}_v} \right)^2 \right] + \lambda \sum_{u,v=1}^{M} \sum_{j=1}^{m} \left(h^u(z_j) - h^v(z_j) \right)^2$$

is called *co-regularised corresponding projections* (CoCP), where $\lambda, \nu_v > 0$ are hyperparameters and the final predictor for the orphan target is the average $h_o = 1/M \sum_{v=1}^{M} h_o^v$.

The objective of CoCP is equipped with an additional regularisation term utilising unlabelled instances which hopefully results in an orphan hypothesis h_o with improved predictive performance.

Lemma 5.9. *With the preconditions of Definition 5.8 as well as Equations 5.25 and 5.26, the system of equations*

$$\begin{pmatrix} D_1 & -2\lambda {U_1}^T U_2 & \ldots & -2\lambda {U_1}^T U_M \\ -2\lambda {U_2}^T U_1 & D_2 & \ldots & -2\lambda {U_2}^T U_M \\ \vdots & \vdots & \ddots & \vdots \\ -2\lambda {U_M}^T U_1 & -2\lambda {U_M}^T U_2 & \ldots & D_M \end{pmatrix} \begin{pmatrix} \pi_o^1 \\ \pi_o^2 \\ \vdots \\ \pi_o^M \end{pmatrix} = \begin{pmatrix} \tilde{G}_1 \tilde{\rho}_o^1 \\ \tilde{G}_2 \tilde{\rho}_o^2 \\ \vdots \\ \tilde{G}_M \tilde{\rho}_o^M \end{pmatrix}.$$

delivers the solution of CoCP.

Proof. The reasoning of the proof is equivalent to the one of Lemma 5.7. □

Analogous to the single-view case, it depends on the prerequisites on the candidate spaces and the relation between the number of supervised targets n and the number of training instances N whether MVCP or CoCP is to be favoured. The computation complexity is $\mathcal{O}(Mn)$ for MVCP and $\mathcal{O}(MN)$ for CoCP as a consequence of the respective matrix inversion.

5.4 Empirical Evaluation

Our aim is to solve the learning problem of orphan screening in the regression scenario, i.e., to predict ligand affinities for orphan proteins with no labelled training data. For this purpose, we consider transfer learning approaches which take labelled training data of other proteins into account. In the present empirical section, we investigate the performance of the learning algorithms from above and evaluate their usefulness for orphan screening. The present section is based on the empirical section in [Giesselbach et al., 2018]. We comment on the the elaborate presentation of results in the following.

5.4.1 Datasets, Implementation, and Experimental Setting

For the practical experiments in Chapter 5, we use 9 protein ligand datasets. Each of the 9 sets relates to a human protein (the target) and comprises between 268 and 2648

ligands annotated with their binding affinity towards the protein as pK_i-value (compare Section 1.3.1 on the biochemical background). More details on the datasets can be found in Appendix B. Hence, we have a data matrix $\Phi(X) \in \mathbb{R}^{n \times D}$ of $n = 8928$ ligands in a feature space $\mathbb{R}^D$ induced by the feature map Φ. The experimental framework and all figures were generated with *Python* 2.7[2], *Jupyter Notebook* [Kluyver et al., 2016] and *Matplotlib* [Hunter, 2007].

For the real-world learning task of orphan screening the feature map Φ is a vectorial representation of small molecular compounds from $\mathcal{X}$. We apply the standard molecular fingerprints ECFP4 and GpiDAPH3 (compare also Section 1.3.2). Additionally, we consider 2 combined variants of the fingerprint formats ECFP4 and GPIDAPH3. Firstly, we use a concatenation of the respective ECFP4 and GpiDAPH3 fingerprint vectors to a final vectorial representation of length $D = 30812$ called *Concat*. The second combined fingerprint was obtained by a *Johnson-Lindenstrauss* (JL) projection according to Section 2.7.1 applied to the concatenated fingerprint Concat and will be denoted *JL-Concat*. In order to obtain the JL property from Equation 2.35, we chose an image dimension of $d = 1000$ and a number of instances $n = 8928$ such that with an error bound of $\varepsilon = 0.1$ the dimension d is approximately $(\ln n)\varepsilon^{-2}$ according to Section 2.7.1 above. Furthermore, we generated the random projection matrix $P \in \mathbb{R}^{d \times D}$ such that for $i = 1, \ldots, d$ and $j = 1, \ldots, D$

$$(P)_{i,j} = \begin{cases} -1 \cdot \frac{1}{\sqrt{1000}} & : \quad \text{with probability } p = 0.5 \\ 1 \cdot \frac{1}{\sqrt{1000}} & : \quad \text{with probability } (1-p) = 0.5 \end{cases}$$

to satisfy Equation 2.36. For more details on the choice of P consult Section 2.7.1 on JL random projections. The JL projection-based ligand representations JL-Concant pursues with the idea of information transfer based on projections. In contrast to the Concat representation, JL-Concat induces a baselines approach with a low-dimensional feature space.

We test and compare CP in its NLCP implementation from Section 5.3.3 with baseline approaches applied to the learning problem of orphan screening. For the sake of simplicity we will refer to the algorithm as CP. An overview of the considered baselines can be found in Table 5.1. With *SCP* we refer to the weighted sum of supervised hypotheses defined in Equation 5.11. The *TLK* predictor assigns an affinity value to pairs of targets and ligands and is described in detail in Section 5.2. For the experiments each of the 9 protein targets is assumed to be the orphan target and the respective other 8 targets serve as supervised targets. In contrast, the TLK variant *TLK-Clo-3* only incorporates the 3 closest targets of the orphan protein t_o as supervised targets. In this context, *closest* (*farthest*) refers to the protein with the biggest (smallest) similarity or kernel value compared to the orphan protein. Given a fixed orphan target, with *Avg* we refer to the average predictor of the respective other 8 supervised targets. Analogous to TLK-Clo-3, the *Avg-Clo-3* algorithm only incorporates the 3 respective closest proteins for the average predictor. The baselines *Closest Protein* and *Farthest Protein* use the supervised hypothesis of the closest and farthest supervised hypothesis, respectively. With *Supervised-l%* we refer to standard SVR with $l\%$, $l \in \{5, 10, 30, 50, 80\}$, labelled training data. SVR is in a sense an optimal but unfair baseline as for an orphan target there is actually no labelled data available. We oppose the performance of the described algorithms using the standard molecular representation formats ECFP4 and GpiDAPH3

[2] `https://www.python.org/`

to the performance results obtained with the fingerprints Concat and JL-Concat. These combined fingerprints add a canonical multi-view approach to our experiments.

In order to simulate the real-world scenario of orphan screening, each of the 9 proteins was assumed to be the orphan target once and the respective 8 others the supervised targets. For a fixed orphan target, we drew 240 ligands from each of the remaining sets and repeated this procedure 10 times. We report RMSE values (compare Section 2.2) to evaluate the regression performance of the different algorithms and averaged over the 10 folds for every orphan target. The choice of ν in Equation 5.15 posed a problem as a labelled training set of 8 supervised targets and corresponding supervised hypotheses for the general assignment of hypotheses from $\mathcal{H}$ to targets from $\mathcal{T}$ was not sufficient to perform a reasonable hyperparameter tuning procedure. We observed in preliminary experiments that the results were barely affected by the choice of ν. Therefore, we fixed $\nu = 5.0$ for all orphan targets. Furthermore, we introduced a small modification in the objective of NLCP in Equation 5.15 with $\beta_o = [\nu G + \lambda \mathbf{I}_n + GNG]^\dagger G\rho_o$ and $\lambda = 1.0$. The summand $\lambda \mathbf{I}_n$ is an additional regularisation term and ensures the existence of the inverse $[\nu G + \lambda \mathbf{I}_n + GNG]^{-1}$ if λ is large enough. For the initial training procedure of the supervised hypotheses for supervised targets we applied a 3-fold cross-validation. The hyperparameters ν and ε of the SVR algorithm were optimised within the ranges $\nu \in \{2^{-5}, 2^{-4}, \ldots, 2^5\}$ and $\varepsilon \in \{0.1, 0.01, 0.001\}$. For a fair comparison between CP and baseline results, we applied our own SVR implementation based on Definition 2.24 for the determination of supervised hypotheses and the Supervised-l% baseline. For the ligands, we used a linear kernel applied to the standard and combined fingerprint representations. The similarity (target kernel) values $k_\mathcal{T}$ for CP according to Section 5.3.3 were derived from a positive semi-definite similarity matrix for proteins. The contained similarity values were calculated based on amino acid sequence similarity measures and normalised.

5.4.2 Results

The experiments in the present empirical section augment the results presented in the work of Giesselbach et al. [2018]. Figure 5.2 shows two boxplots with the RMSE results for CP and all baselines from Table 5.1. The RMSEs are averaged over all 9 proteins and all 10 ligand draws for the supervised models. We report averaged RMSEs for the

Name	**Description**
Simplified	SCP approach from Equation 5.11
TLK	TLK approach from Section 5.2
TLK-Clo-3	TLK approach from Section 5.2
Avg	Average of supervised predictors
Avg-Clo-3	Average of supervised predictors
Closest Protein	Predictor of the closest protein
Farthest Protein	Predictor of the farthest protein
Supervised-l%	Standard SVR with l% of labelled data

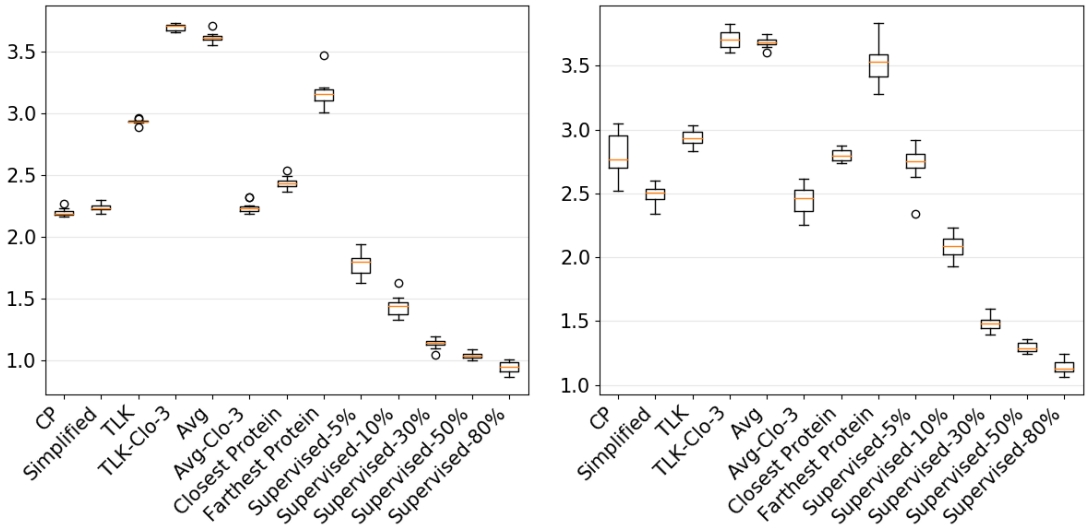
Table 5.1: Overview of baseline approaches

standard molecular fingerprints ECFP4 (a) and GpiDAPH3 (b). To start with, we realise a generally worse performance of all approaches with GpiDAPH3 in comparison to the application of ECFP4. In both cases, CP outperforms the Avg baseline which does not make use of inter-target similarities at all. For the ECFP4 fingerprint, CP also beats all other baselines that make use of these similarities exhibiting an average RMSE of 2.197. However, this is not the case with respect to the Simplified and Avg-Clo-3 approach if GpiDAPH3 was used. In order to understand how the similarity relation of proteins affects the prediction quality, we compare the CP performance with the performance of the Closest Protein and Farthest Protein model. The fact that Closest Protein performs much better than Farthest Protein supports the intuition that the molecular similarity principle [Bender and Glen, 2004] does not only hold for small compounds but also for proteins, in particular, for the orphan protein. The molecular similarity principle introduced in Section 1.3.3 states that similar molecules are supposed to have similar properties with respect to binding and vice versa. The modified average model Avg-Clo-3 of the 3 most similar targets compared to the orphan target yields a significant performance improvement both for ECFP4 and GpiDAPH3. Again, the orphan protein's binding model obviously profits from the focus on closer proteins. Additionally, we compare with the state-of-the-art approach TLK for orphan screening which incorporates both target and ligand similarities. It was introduced in detail in Section 5.2 and basically solves a supervised problem in terms of target-ligand pairs. CP outperforms TLK for both fingerprints. However, an advantage of the TLK approach is that no supervised hypotheses have to be learned for proteins with training information in a preliminary step. Regarding TLK-Clo-3 we could not show the positive effect of emphasising closer targets during model calculation which we have seen for Avg-Clo-3 versus Avg. The approach closest to CP in terms of RMSE is the Simplified approach. Therefore, depending on the precise learning task at hand, it might be a valuable alternative to CP because of its shorter running time. Supervised-l% denotes the standard supervised SVR algorithm which uses l% of the available data as labelled training examples. As CP operates in the learning scenario of no labelled training information for the orphan target, Supervised-l% outperforms CP as expected. We pursued the experiments with CP and baselines for orphan screening using 2 combined fingerprints as canonical multi-view representations of small molecular compounds. In Figures 5.3 (a) and 5.3 (b) we observe that the considered algorithms show a very similar performance in applying the combined variant Concat and JL-Concat compared to the ECFP4 fingerprint (see Figure 5.2 (a) above). This will be discussed in the following section.

5.4.3 Dicussion

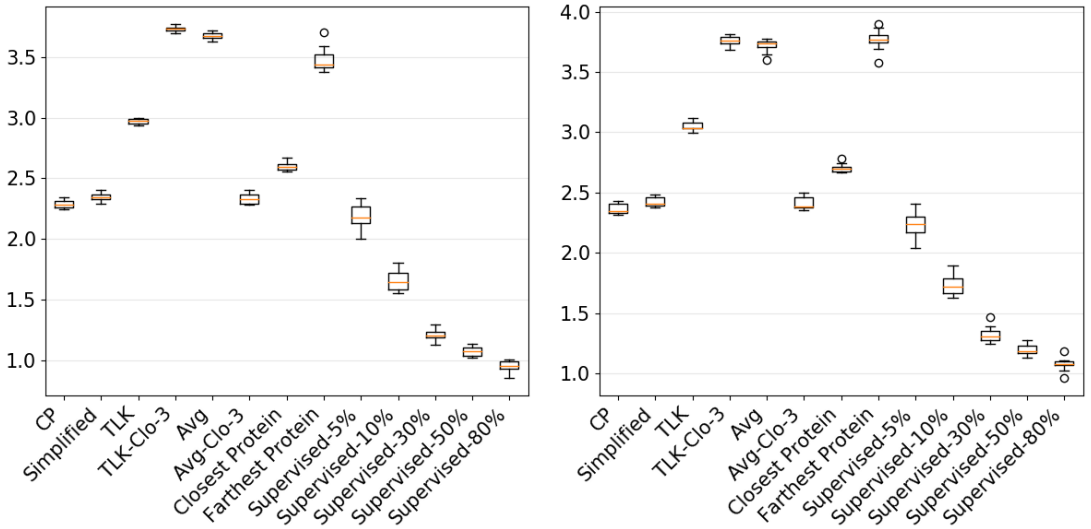
Orphan screening is a challenging and important real-world learning problem. More precisely, we investigated the task of ligand affinity prediction for a protein with no labelled training compounds. We defined CP and variants of it as a novel kernel method to master this unsupervised problem. The approach of CP is to firstly derive protein-ligand binding models for protein targets with labelled training data. Secondly, with further information about the relations between protein targets the knowledge about protein-ligand binding is transferred to the orphan target. For this reason, CP can be assigned to transfer learning or multi-task learning as well. Supervised learning algorithms are based on labelled data and its results degrade typically with a decreasing number of labelled examples. For both the standard molecular fingerprints ECFP4 and GpiDAPH3 and the combined representations Concat and JL-Concat we observed that

FIGURE 5.2: RMSEs of CP and baselines averaged over all proteins and draws using fingerprint ECFP4 (a) and GpiDAPH3 (b)



(a) (b)

FIGURE 5.3: RMSEs of CP and baselines averaged over all proteins and draws using the fingerprints Concat (a) and JL-Concat (b)



(a) (b)

the averaged RMSE values seem to increase exponentially from a training examples ratio from 80% to 5% (compare Figures 5.2 (a) and (b) as well as 5.3 (a) and (b)). In the performed experiments, CP shows comparable results to the Supervised-5% approach. This is an interesting result as CP operates without labelled training examples for the orphan target and, therefore, the Supervised-l% approaches are unrealistic baselines. A conclusion from this result can be that a training with only few potentially unfavourably examples is inferior to an approach without labelled training examples for the focused orphan target which includes plenty of information from related protein targets as well as well as the inter-protein relation. It is also remarkable that CP outperforms the orphan screening baseline TLK (see Section 5.2) for ligand prediction clearly for every applied representation of molecular compounds. However, although CP induces a general predictor function f from targets to hypotheses in Equation 5.7, the optimisation to obtain h_o has to be performed for every orphan protein target t_o again. In contrast, it is an advantage of the TLK algorithm, that it derives a general predictor function f in one optimisation without the need to generate supervised hypotheses for supervised targets in a preliminary step. However, the precise performance strongly depends on the utilised fingerprint representation. The application of the fingerprint ECFP4 yielded a better RMSE performance than the fingerprint GpiDAPH3 (see Figures 5.2 (a) and (b)). Additionally, we showed that CP and baselines using the concatenated fingerprint Concat exhibit approximately equal results compared to the application of the standard molecular fingerprint ECFP4 (see Figure 5.3 (a)). That means, the multi-view approach automatically performs nearly equal to the best of the single-view approaches, an effect that we also found in the experimental sections of Chapter 3 and Chapter 4. The effect remained for the JL-Concat representation which only uses a drastically reduced fingerprint dimension compared to the concatenation of standard molecular fingerprints.

In summary, the very good performance of CP in the practical experiments motivates future work in orphan screening. Including labelled training information for the orphan target and considering other GPCR datasets are interesting directions to consider. In the following section we define the versions MVCP and MVOPCA of CP putting CP in a multi-view setting which can be investigated in future work.

5.5 Future Work: Orphan Principal Component Analysis

In this section, we present an alternative algorithm with single- and multi-view variant to solve the orphan screening learning problem. It is not included in the empirical section of the present chapter, for which reason we present it in an extended future work section. The proposed algorithm is again based on projections and the transfer learning approach discussed in Section 5.1 (compare also Figure 5.1). As there is no labelled training data for the orphan target t_o, we make use of known hypotheses for supervised targets, i.e., pairs $(t_1, h_1), \ldots, (t_n, h_n) \subseteq \mathcal{T} \times \mathcal{H}$. In the considered scenario, $\mathcal{T}$ is a target space with a similarity measure (kernel function) $k_{\mathcal{T}} : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ and $\mathcal{H}$ is a hypothesis space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Other than CP and its variants from above, the novel approach introduced in the following applies *principal component analysis* (PCA) for dimensionality reduction (compare Section 2.7.2) to infer an appropriate hypothesis for the orphan target t_o. More precisely, *Interactive knowledge-based kernel principal component analysis* (IPCA) was developed by Oglic et al. [2014] to facilitate the integration of expertise on a focused domain into a visualisation process. The idea of IPCA is to directly include expert knowledge in form of *control points*, *classification*

constraints, *must-link constraints*, and *cannot-link constraints* [Oglic et al., 2014] into the PCA optimisation objective and to calculate the resulting projection accordingly. We will use the IPCA algorithm with must-link constraints below in order to solve the unsupervised learning problem of finding a hypothesis $h_o \in \mathcal{H}$ for the orphan target t_o. For more details on IPCA we refer to Oglic et al. [2014].

5.5.1 Base Algorithm

At first, we define the instance space $\mathcal{X}$ as a union of targets and hypotheses

$$\mathcal{X} = \mathcal{T} \cup \mathcal{H}. \tag{5.27}$$

which is crucially different to the instance space $\mathcal{X}$ being molecular compounds or a product of proteins and potential ligands as considered in Section 5.2.

very special approach. Additionally, we define a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ via

$$k(x, x') = \begin{cases} k_{\mathcal{T}}(x, x') & : \quad x, x' \in \mathcal{T} \\ \langle x, x' \rangle_{\mathcal{H}} & : \quad x, x' \in \mathcal{H} \\ 0 & : \quad \text{otherwise} \end{cases} .$$

The kernel k is positive semi-definite as $k_{\mathcal{T}}$, $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, and the constant 0 are positive semi-definite as well. We fix a set

$$X_{\cup} = \{x_1, \ldots, x_p, x_{p+1}, \ldots, x_{p+q}\} \subseteq \mathcal{X}, \tag{5.28}$$

5.2 where $\{x_1, \ldots, x_p\} \subseteq \mathcal{T}$ are targets and $\{x_{p+1}, \ldots, x_{p+q}\} \subseteq \mathcal{H}$ are hypotheses. The set $X_{\cup}$ must necessarily comprise the supervised targets t_i, the supervised hypotheses h_i, $i = 1, \ldots, n$, and the orphan target t_o. Optionally, further targets and hypotheses from $\mathcal{X}$ can be contained in $X_{\cup}$ as well. Let $K_{\mathcal{T}}$ be the Gram matrix of $k_{\mathcal{T}}$ with respect to $x_1, \ldots, x_p$ and $K_{\mathcal{H}}$ be the matrix

$$K_{\mathcal{H}} = (\langle x_i, x_j \rangle_{\mathcal{H}})_{i,j=p+1}^{p+q} .$$

Putting it together, the Gram matrix K of k with respect to $x_1, \ldots, x_p, x_{p+1}, \ldots, x_{p+q}$ equals

$$K = \begin{Bmatrix} K_{\mathcal{T}} & \mathbf{0}_{p \times q} \\ \mathbf{0}_{q \times p} & K_{\mathcal{H}} \end{Bmatrix} \subseteq \mathbb{R}^{D \times D}, \tag{5.29}$$

where $D = p + q$.

The idea of IPCA is to enrich the dimensionality reduction technique of PCA with additional expert knowledge on the considered domain. The so-called *must-link constraints* describe the request that similar instances are supposed to have a small distance in the image space of the optimised IPCA projection. In the orphan screening scenario the must-link constraints demand supervised targets t_i and their corresponding supervised hypotheses h_i to have a small distance after projection for all $i = 1, \ldots, n$. These n constraints are represented with the following set $\mathcal{C}$ of index pairs for instances

$$\mathcal{C} = \{(l, l') : x_l \text{ is a supervised target with supervised hypothesis } x_{l'}\}$$

with cardinality equal to n. Consequently, $l \in \{1, \ldots, p\}$ and $l' \in \{p+1, \ldots, p+q\}$ holds true. In particular, x_l is not the orphan target t_o. Oglic et al. [2014] incorporated the requirements expressed with must-link constraints in the determination of the optimal projection Π as follows

$$\begin{aligned} \max_{\Pi \in \mathbb{R}^{D \times d}} & \operatorname{tr}\left(\frac{1}{D}\Pi^T K H_D K \Pi\right) - \nu \operatorname{tr}\left(\frac{1}{|\mathcal{C}|}\Pi^T K L K \Pi\right), \\ \text{s.t. } & \Pi^T K \Pi = \mathbf{I}_d \end{aligned} \tag{5.30}$$

where $\nu > 0$ is a trade-off hyperparameter. The matrices H_D and L in Equation 5.30 are defined as

$$H_D = \mathbf{I}_D - \tfrac{1}{D}\mathbf{1}_D(\mathbf{1}_D)^T$$

as well as

$$L = \sum_{(l,l') \in \mathcal{C}} (e_l - e_{l'})(e_l - e_{l'})^T,$$

where e_l is the unit vector in $\mathbb{R}^D$ such that the l-th component is equal to 1. With $A = K^{1/2}\Pi$ and $M = K^{1/2}(\frac{1}{D}H_D - \frac{\nu}{n}L)K^{1/2}$, the optimisation in Equation 5.30 is equivalent with the standard PCA optimisation from Equation 2.39 in the introductory section on kernel methods

$$\begin{aligned} \max_{A \in \mathbb{R}^{D \times d}} & \operatorname{tr}\left(A^T M A\right) \\ \text{s.t. } & A^T A = \mathbf{I}_d. \end{aligned}$$

The modified PCA projection in Equation 5.30 demands supervised targets and corresponding hypotheses to have a small Euclidean distance in the image space of the projection. Consequently, the unsupervised task of orphan screening can be solved by searching for the hypothesis in $\mathcal{H}$ with smallest distance to the projection of the orphan target t_o. For a general Gram matrix $K \in \mathbb{R}^{n \times n}$, corresponding to instances $x_1, \ldots, x_n \in \mathcal{X}$, and some instance $x \in \mathcal{X}$ we fix the notation

$$K(x) = (k(x, x_1), \ldots, k(x, x_n))^T, \tag{5.31}$$

which we will use in the following.

Definition 5.10 (OPCA)**.** Let $\mathcal{X}$ and $K \in \mathbb{R}^{D \times D}$ be defined as in Equations 5.27 and 5.29. Let $\Pi \in \mathbb{R}^{D \times d}$ be the solution of the optimisation in Equation 5.30. The determination of the orphan hypothesis h_o for the orphan target t_o via

$$h_o = \operatorname*{argmin}_{h \in \mathcal{H}'} \|\Pi^T(K(h) - K(t_o))\|^2, \tag{5.32}$$

is called *orphan principal component analysis* (OPCA).

There are different options for the precise implementation of the OPCA algorithm. At the least, the set $X_\cup$ comprises the supervised targets and corresponding hypotheses as well as the orphan target. In order to find a better orphan hypothesis, further hypotheses can be added to $X_\cup$, for example, random hypotheses or linear combinations of $h_1, \ldots, h_n$. If $\mathcal{H}' = X_\cup$, the solution of Equation 5.32 reduces to finding the minimum

of a finite set of numbers, i.e., the values of the objective in Equation 5.32 for the elements of $X_\cup$. In the case of $\mathcal{H}' = \mathcal{H}$ the determination of the solution for h_o in Equation 5.32 depends on the properties of the hypothesis space $\mathcal{H}$ and the kernel function $k_\mathcal{H}$. We present a non-trivial solution for the case that $\mathcal{H}'$ equals the span of the supervised hypotheses in the following lemma. We use the notation from Section 5.5.1.

Lemma 5.11. *Let $x_{p+1}, \ldots, x_{p+q}$ be the supervised hypotheses and the candidate space $\mathcal{H}' \subseteq \mathbb{R}^{d'}$ from Equation 5.32 be defined as*

$$\mathcal{H}' = \operatorname{span}\{x_{p+1}, \ldots, x_{p+q}\}.$$

Furthermore, let $k_\mathcal{H}$ be the linear kernel and $K_\mathcal{H}$ as well as $K_\mathcal{T}$ be the Gram matrices applied in 5.29 such that $K_\mathcal{H}^{-1}$ exists. We consider a decomposition of the projection $\Pi \in \mathbb{R}^{D \times d}$ from Equation 5.30 into an upper submatrix $\Pi_\mathcal{T} \in \mathbb{R}^{p \times d}$ and a lower submatrix $\Pi_\mathcal{H} \in \mathbb{R}^{q \times d}$. Let $t_o \in \mathcal{T}$ be the orphan target. The solution of the OPCA optimisation in Equation 5.32 can be calculated as

$$\alpha_o = K_\mathcal{H}^{-1} \left(\Pi_\mathcal{H} \Pi_\mathcal{H}^T\right)^{-1} \Pi_\mathcal{H} \Pi_\mathcal{T}^T \; K_\mathcal{T}(t_o), \tag{5.33}$$

where $\alpha_o \in \mathbb{R}^q$ and $h_o = \sum_{i=1}^q (\alpha_o)_i x_{p+i}$.

Proof. We define the instance matrix $X \in \mathbb{R}^{q \times d'}$ via

$$X = \begin{pmatrix} x_{p+1} \\ \vdots \\ x_{p+q} \end{pmatrix}, \tag{5.34}$$

where $x_{p+1}, \ldots, x_{p+q} \in \mathcal{H}$ are the supervised hypotheses. By definition of $\mathcal{H}'$, every hypothesis $h \in \mathcal{H}'$ has got a representation $h = X^T \alpha$ for appropriate $\alpha \in \mathbb{R}^q$. For the objective in Equation 5.32 we conclude

$$\begin{aligned} \|\Pi^T (K(h) - K(t_o))\|^2 &= \|\Pi_\mathcal{H}^T \; K_\mathcal{H}(h) - \Pi_\mathcal{T}^T \; K_\mathcal{T}(t_o)\|^2 \\ &= \|\Pi_\mathcal{H}^T \; X X^T \alpha - \Pi_\mathcal{T}^T \; K_\mathcal{T}(t_o)\|^2, \end{aligned}$$

where the first equality follows from the zero matrices in the definition of K in Equation 5.29. We obtain the parameterised solution α_o of the OPCA minimisation in Equation 5.32 by putting the gradient with respect to α equal to zero

$$2 \left(\Pi_\mathcal{H}^T \; X X^T \alpha_o - \Pi_\mathcal{T}^T \; K_\mathcal{T}(t_o)\right)^T \cdot \left(\Pi_\mathcal{H}^T \; X X^T\right) = \mathbf{0}_q^T$$

As $K_\mathcal{H} = X X^T$ holds true because of the linear kernel, α_o can be calculated as

$$\begin{aligned} \alpha_o &= \left(K_\mathcal{H} \; \Pi_\mathcal{H} \Pi_\mathcal{H}^T \; K_\mathcal{H}\right)^{-1} K_\mathcal{H} \; \Pi_\mathcal{H} \Pi_\mathcal{T}^T \; K_\mathcal{T}(t_o) \\ &= K_\mathcal{H}^{-1} \left(\Pi_\mathcal{H} \Pi_\mathcal{H}^T\right)^{-1} \Pi_\mathcal{H} \Pi_\mathcal{T}^T \; K_\mathcal{T}(t_o) \end{aligned}$$

which finishes the proof. □

5.5.2 Multi-View OPCA

Analogous to the multi-view CP algorithm in Section 5.3.4 above, we present a multi-view variant of OPCA in the final section of Chapter 5. To this aim, the OPCA algorithm has to be generalised slightly. Assume there is a target space $\mathcal{T}$ with kernel function $k_{\mathcal{T}}$ for targets and, furthermore, there are hypothesis spaces H_v with reproducing kernels k_v, $v = 1, \ldots, M$, which represent M views on data instances. Again, we consider an instance space as union of target space and hypothesis spaces

$$\mathcal{X}^{\mathrm{MV}} = \mathcal{T} \cup \mathcal{H}_1 \cup \ldots \cup \mathcal{H}_M.$$

In contrast to single-view OPCA, the hypothesis spaces $\mathcal{H}_v$ correspond to M views on data. As the labelled training instances of the supervised targets t_i, $i = 1, \ldots, n$, have different view representations as well, M supervised hypotheses can be learned with arbitrary supervised kernel methods for each target t_i. Hence, we have n $(M+1)$-tuples $(t_i, h_i^v, \ldots, h_i^v)$ as subsets of $\mathcal{X}^{\mathrm{MV}}$. Analogous to $X_{\cup}$ in Equation 5.28 from the single-view OPCA version above we define

$$X_{\cup}^{\mathrm{MV}} = \{x_1, \ldots, x_p, x_{p+1}, \ldots, x_{p+q}, \ldots, x_{p+(M-1)q+1}, \ldots, x_{p+Mq}\}, \tag{5.35}$$

where $x_1, \ldots, x_p$ are targets and $x_{p+(v-1)q+1}, \ldots, x_{p+v \cdot q}$ are hypotheses from $\mathcal{H}_v$. We define a kernel function $k^{\mathrm{MV}} : \mathcal{X}^{\mathrm{MV}} \times \mathcal{X}^{\mathrm{MV}} \to \mathbb{R}$

$$k^{\mathrm{MV}}(x, x') = \begin{cases} k_{\mathcal{T}}(x, x') & : \quad x, x' \in \mathcal{T} \\ k_v(x, x') & : \quad x, x' \in \mathcal{H}_v \\ 0 & : \quad \text{otherwise} \end{cases},$$

Let $K_{\mathcal{T}}$ be the Gram matrix of $k_{\mathcal{T}}$ according to the targets $x_1, \ldots, x_p$ and K_v be the Gram matrices of k_v corresponding to the hypotheses $x_{p+(v-1)q+1}, \ldots, x_{p+v \cdot q}$. We obtain the Gram matrix with respect to instances from $X_{\cup}^{\mathrm{MV}}$ as

$$K^{\mathrm{MV}} = \begin{pmatrix} K_{\mathcal{T}} & \mathbf{0}_{p \times q} & \mathbf{0}_{p \times q} & \ldots & \mathbf{0}_{p \times q} \\ \mathbf{0}_{q \times p} & K_1 & \mathbf{0}_{q \times q} & \ldots & \mathbf{0}_{q \times q} \\ \mathbf{0}_{q \times p} & \mathbf{0}_{q \times q} & K_2 & \ldots & \mathbf{0}_{q \times q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{q \times p} & \mathbf{0}_{q \times q} & \mathbf{0}_{q \times q} & \ldots & K_M \end{pmatrix} \in \mathbb{R}^{D \times D}, \tag{5.36}$$

where the dimension D is equal to $p + Mq$. The supervised view hypotheses h_i^v, $v = 1, \ldots, M$, correspond to the supervised target t_i for $i = 1, \ldots, n$. For this reason, their pairwise PCA projections are supposed to have a small distance in the image space of the projection analogous to the single-view OPCA approach. More formally, we define the multi-view version of the index set with cardinality Mn

$$\mathcal{C}^{\mathrm{MV}} = \{(l, l') : \; x_l \text{ is a supervised target and} \\ x_{l'} \text{ is one of its supervised view hypotheses}\},$$

which is used to incorporate the must-link constraints in the standard PCA objective as proposed by Oglic et al. [2014] analogous to the single-view case in Equation 5.30.

Definition 5.12 (MVOPCA)**.** Let $t_o \in \mathcal{T}$ be an orphan target without labelled training information for the principal learning task. Furthermore, let $\mathcal{X}^{\mathrm{MV}}$ and $K^{\mathrm{MV}} \in \mathbb{R}^{D \times D}$

be defined as in Equations 5.35 and 5.36, respectively. We fix the vector

$$K^{\mathrm{MV}}(x) = (k^{\mathrm{MV}}(x, x_1), \ldots, k(x, x_D))^T,$$

where $x_1, \ldots, x_D \in \mathcal{X}^{\mathrm{MV}}$ are the element of $X_{\cup}^{\mathrm{MV}}$ from Equation 5.35. Let $\Pi^{\mathrm{MV}} \in \mathbb{R}^{D \times d}$ be the solution of the optimisation in Equation 5.30, such that $\mathcal{X} = \mathcal{X}^{\mathrm{MV}}$ and $K = K^{\mathrm{MV}}$ are applied. The determination of the hypotheses h_o^v via

$$h_o^v = \operatorname*{argmin}_{h \in \mathcal{H}'_v} \|(K^{\mathrm{MV}}(h) - K^{\mathrm{MV}}(t_o))\Pi^{\mathrm{MV}}\|^2, \tag{5.37}$$

where for $v = 1, \ldots, M$, the candidate space $\mathcal{H}'_v$ is an appropriate subset of $\mathcal{H}_v$, is called *multi-view orphan principal component analysis* (MVOPCA).

The final predictor h_o for the orphan target t_o is supposed to be the average of the orphan view hypotheses

$$h_o = \frac{1}{M} \sum_{v=1}^{M} h_o^v.$$

Once the PCA projection $\Pi^{\mathrm{MV}} \in \mathbb{R}^{D \times d}$ is calculated in the multi-view scenario of Equation 5.30, the optimisation problems for $v = 1, \ldots, M$ in Equation 5.37 are independent from each other. For this reason, orphan screening can be solved via MVOPCA according to Definition 5.12 analogous to the solution of OPCA above.

Chapter 6

Conclusion

6.1 Summary

In the present thesis, we considered the regression problem of ligand affinity prediction. Ligands are small molecular compounds that bind to proteins. The strength of the binding is a characteristic of the precise protein-ligand pair and it is expressed with a real-valued affinity. Protein-ligand complexes are involved in a multitude of biochemical pathways. For this reason, ligands are potential drug candidates and ligand affinity prediction has the potential to greatly accelerate the drug discovery process by making it more efficient. Ligand affinity prediction comes with a variety of descriptors for molecular compounds. Consequently, one is confronted with one of the inherent challenges to choose the optimal representation for instances to solve the considered learning problem. Additionally, molecular compounds can canonically be interpreted as graph data. Given a particular protein, often only few annotated compounds are available because of the time- and cost-intensive determination in practice. It is known that the generalisation performance of machine learning models fall with a decreasing number of labelled examples. In contrast, millions of unlabelled small molecules are gathered in molecular databases which are disposable to describe the search space. In the extreme case, there are no labelled molecular compounds for so-called orphan proteins at all. The obvious question here is how to overcome this lack of information. However, relations between the orphan protein and other reference proteins with respect to structure or function exist and can be expressed numerically. The majority of related work was concerned with ligand prediction as a classification problem, i.e., the simplified problem to decide whether a compound binds to a protein or not. In almost the same manner, only few approaches for orphan screening which is the unsupervised version of affinity prediction existed at all. We considered affinity prediction

(i) as supervised regression problem for graph instances with multiple data representations (Chapter 3),

(ii) as semi-supervised regression task with few labelled examples and multiple views on data (Chapter 4), and

(iii) as unsupervised or transfer learning task (orphan screening) in the single- and multi-view scenario (Chapter 5).

We applied multi-view kernel methods to solve affinity prediction under particular consideration of the three settings from above. Kernel functions are generalised similarity measures for the corresponding data instances and they are canonically related to the views on data. Kernels provide the kernel methods with useful properties. We developed novel schemes to select graph patterns as a basis for the multi-view learning (MVL) model (Chapter 3) as well as novel algorithms for both semi-supervised multi-view regression (Chapter 4) and transfer learning (Chapter 5). On the one hand, we managed to improve the ligand affinity prediction performance despite of the limitations

(i) multitude of data representations,

(ii) usually only few labelled ligands, and

(iii) no labelled ligands at all in the case of orphan proteins.

On the other hand, we contributed to the machine learning subareas of multi-view learning and kernel methods by presenting a novel selection scheme based on graph patterns and novel algorithms that can be applied in the described semi-supervised and unsupervised setting from above. More details on the results in the three scenarios will be discussed below. In summary, we achieved the main objectives of the thesis independent of the preconditions on data availability. We showed that the affinity prediction performance could be improved using MVL without the need to choose the optimal representation for molecular instances. In all three settings the performance of the multi-view approaches at least measured up to the performance of the best single-view baseline approach by including multiple data representations simultaneously. The novel machine learning techniques can be applied to general learning problems with the mentioned preconditions on data representation and structure.

In Chapter 3, we regarded ligand affinity prediction as a supervised regression task. That means, we assumed sufficient labelled training ligands were available from molecular databases for a considered protein. As molecules are atoms connected by different types of chemical bonds, the data instances for affinity prediction can be interpreted as undirected labelled graphs. Many different representations for small molecular compounds exist a priori from a variety of applications. Moreover, various graph patterns describe the properties of graphs and can therefore be adducted to represent the molecular instances as well. Supervised affinity prediction is a relevant real-world application, as the selection of molecular compounds with predicted high affinity values from a large database of molecules can be included as promising candidates in drug discovery experiments in practice. Not only in the context of ligand affinity prediction, the choice of the best data representation is a non-trivial problem. To test and oppose every single view on data would be computationally expensive. Even if a combination of multiple views can be incorporated, as done in MVL, a preselection of (graph) data representations must be performed in order to reduce the complexity of the machine learning modelling. Other than existing standard molecular fingerprints, which consider predefined structural units in the neighbourhood of atoms, we systematically collected cyclic, tree, and shortest path graph patterns based on WL labelling in increasing depths for the representation of small molecular compounds as potential ligands. We applied a least squares and an ε-insensitive loss variant of multiple kernel learning (ℓ_2-MKL and ε-MKL). We developed a preprocessing scheme to make a preselection out of the large number of available graph patterns for the representation of data. More precisely, for each graph pattern class we identified the WL depths with the best single-view regression

results for the prediction of affinities in preliminary experiments. Subsequently, we used the best combinations of graph pattern representations from preliminary experiments to perform MKL with multi-pattern kernels (MPK) following the consensus principle from above. We refer to the combination of promising graph patterns and MKL as MPK-MKL scheme to handle the multitude of graph pattern representations. In the supervised setting, we accomplished the objectives of this thesis. Firstly, we showed that ligand affinity prediction as a regression task can be improved via MKL. Secondly, we tackled the inherent challenge of the optimal fingerprint choice for the representation of compound instances. Particularly, we took into consideration the graph structure of the learning objects and performed a systematic selection of graph patterns into the modelling process. In the empirical evaluation, the MPK-MKL approaches outperformed the single-view baselines in average for both a binary and a counting feature representation of the graph patterns and, particularly, for standard molecular fingerprints. For the considered protein-ligand datasets we observed that WL label patterns showed the best prediction results in preliminary single-view approaches with respect to the root mean squared error (RMSE) of true label and predicted affinity. The MPK-MKL scheme for multi-view learning can be applied for any regression problem with graph instances and sufficient labelled training examples.

In Chapter 4, we investigated ligand affinity prediction in the semi-supervised setting. Semi-supervision in this context refers to the fact that in addition to a few ligands with known affinity with respect to a given protein also a lot of unlabelled molecular compounds are available for learning. This is a more realistic affinity prediction scenario (compared to Chapter 3) as the determination of affinities of small molecules in laboratories is expensive, whereas plenty of synthesizable compounds are gathered and enriched with additional information in molecular databases. These compounds serve as potential ligand candidates and representatives of the instance space of molecules. Both labelled and unlabelled molecular compounds can be represented with a variety of molecular fingerprints and an appropriate representation has to be chosen. To the best of our knowledge we are the first to combine both a semi-supervised approach and MVL for ligand affinity prediction. On the one hand, via the fusion of semi-supervision and MVL it is possible to omit the choice of the best data representation and, on the other hand, unlabelled instances can be utilised to compensate for the small number of labelled compounds. We applied the extended regularised risk minimisation (RRM) approach of co-regularisation, which in addition to the empirical risk minimisation for labelled examples aims at the reduction of an error term for unlabelled instances. More precisely, the unlabelled error term compares the predictions of different model functions which relate to particular views for unlabelled instances. We defined co-regularised support vector regression (CoSVR) as a novel kernel method. In particular, we solved and discussed the least squares and ε-insensitive loss variants with respect to the co-regularisation term (ℓ_2-CoSVR and ε-CoSVR). In a stepwise manner we modified these algorithms in order to reduce the number of optimisation variables and algorithm parameters. Finally, we presented Σ-CoSVR which exhibits complexity properties of a single-view algorithm. The empirical evaluation yielded that CoSVR achieves lower RMSE values for the predicted affinities in comparison to the majority of single- and multi-view baselines. Like in the supervised case, we achieved the objectives of the present thesis stated in the introduction. Firstly, the prediction error of ligand affinities could be reduced via the multi-view approach of co-regularisation. The multi-view approaches in the empirical analysis at least performed as good as the best single-view baseline by including all

molecular representations in one optimisation problem. By investigating the more realistic scenario of few labelled ligands and sufficient unlabelled database compounds, we addressed one of the limitations of affinity prediction from the introduction. Secondly, according to machine learning techniques, we presented the novel kernelised multi-view algorithm CoSVR and different variants of it with respect to the number of optimisation variables. For the variant of Σ-CoSVR we proved a Rademacher bound for the co-regularised candidate function class. The Rademacher bound can be used to control the expected error.

Thirdly, we considered affinity prediction in the most challenging situation where no labelled training compounds are available for the protein of interest. The described learning problem is called orphan screening. The complete absence of ligands with affinity label represents another limiting circumstance of affinity prediction tasks discussed in the introduction. However, there is labelled training information for other proteins which are related to the orphan protein to some extent. The inter-protein relation can be expressed by a similarity measure calculated, for example, from structural or taxonomy properties. Labelled and unlabelled compounds can be represented with a variety of molecular fingerprints in the unsupervised setting as well. Receptor proteins in central biochemical pathways are orphan proteins and the prediction of potential ligands would support the discovery of novel drugs. Orphan screening can be regarded a general learning problem for prediction tasks with the same preconditions on data. We achieved the objectives of the thesis in this unsupervised setting and compensated for the lack of labelled training data by two projection-based approaches from transfer learning which infer a binding model from binding information of other proteins. In addition to labelled training instances for related proteins and similarity information for small compounds, both approaches include inter-protein relations in order to enable the transfer of knowledge from one protein to another. The first approach of corresponding projections (CP) minimises an objective similar to regularised RRM. Due to the lack of labelled training examples, the empirical risk is replaced by a term which adjusts projections of targets and corresponding hypotheses. For this primarily single-view algorithm we define a linear, a simplified, non-linear (kernel) and a multi-view variant. The empirical evaluation showed that CP is able to outperform the orphan screening state-of-the-art approach of target-ligand kernels (TLK) as well as further baselines if the molecular fingerprint was chosen appropriately. CP experiments with combined (multi-view) representations of molecular compounds delivered promising results as well. Firstly, CP based on the combined fingerprints performed as well as the best single-view CP approach. Secondly, this was even the case if the dimensionality reduction technique of Johnson-Lindenstrauss (JL) projection was applied to the combined fingerprint. The second approach for the solution of orphan screening is a variation of principal component analysis called orphan principal component analysis (OPCA), which includes the connection between proteins and corresponding hypotheses via so-called must-link constraints in the optimisation step. OPCA is also a single-view kernel method in the first place, which can be transformed into a multi-view algorithm. We presented novel kernel methods for transfer learning. On the one hand, they can be applied to solve an unsupervised problem for an orphan target. On the other hand, the primal learning problem of the transfer task is not restricted to regression. CP, OPCA, and their variants can be utilised to solve classification or other learning tasks as well if the respective preconditions are met.

6.2 Future Directions

We considered affinity prediction in different settings of data availability and concurrently investigated multi-view kernel methods for supervised, semi-supervised, and unsupervised learning. The broad range of algorithms and potential applications of the considered techniques open a wide scope for future work. We list interesting topics in the following.

A starting point for future work is the continuation of the empirical analysis of MVL for ligand affinity prediction. The prediction of ligand affinities is prospectively useful to predict promising drug candidates in form of ligands with high predicted binding affinities. Hence, for practical purposes the order of molecular compounds with respect to their affinities is more important than the correctness of the affinity values. For this reason, other evaluation measures, such as the rank correlation coefficient Kendall's Tau should be considered as well similar to the analysis of Li et al. [2011]. Similarly, the utilisation of further molecular fingerprint formats and further kernel functions, e.g., the Tanimoto kernel [Geppert et al., 2009, 2008], would be useful to extend the existing empirical analysis. The empirical results in Chapter 4 indicate that the prediction performance goes up with increasing number of integrated views in a MVL approach. Therefore, the inclusion of three or even more views might be beneficial. However, the inclusion of more and more views leads to a longer running time which is a general issue of MVL methods that is worth investigating in the future as well. The results in Chapter 5 achieved in combination with the application of dimensionality reduction techniques, such as JL or PCA projections, suggest to examine the redundancy in the features of molecular fingerprints towards feature selection and finally model reduction. Although the three settings supervised learning, semi-supervised learning and unsupervised learning represent very different learning scenarios which do not appear simultaneously in practice, it would be very illustrative with respect to machine learning algorithms and protein-ligand data to compare the outcomes of MKL, co-regularisation, and projection-based approaches directly. In order to lift ligand affinity prediction from the so far rather hypothetical level, a next step towards a support of drug discover by machine learning would be validation of prediction results with real-world laboratory tests. The thesis was oriented towards the relevant real-world problem of ligand affinity prediction. Though, we pointed out that all proceedings and presented novel kernel algorithms are applicable to other tasks with the same preconditions on the learning scenario. Consequently, the usage and evaluation of the algorithms from all three main chapters to other learning problems than affinity prediction would be a great benefit. A list of alternative tasks can be found in the introduction.

In Chapter 3 we assumed a supervised learning scenario with sufficient labelled training examples and the availability of multiple representations for data instances. Future efforts concerning this learning problem should generally take artificial neural networks (ANN) [Speck-Planche and Cordeiro, 2014, Ferreira and Andricopulo, 2019, Tetko and Engkvist] into account as they come with a canonical multi-view variant. More precisely, data instances can be included in the input layer of the network in form of tensors which directly facilitate the inclusion of different data representations. The view choice is still not completely obsolete by the proposed MPK-MKL scheme for the preselection of promising view combinations. The resulting issues of running time complexity might also be solved via ANNs in combination with parallel computing. Chapter 4 was dedicated to the semi-supervised setting stated to be the most realistic real-world scenario

for ligand affinity prediction. The good performance of co-regularised algorithms from the respective empirical analysis led us to the conclusion that feature information was extracted from unlabelled data. In future attempts, we would like to understand better how this extraction occurs and, particularly, how the sparsity property of the molecular fingerprints relates to that process. The longer running time of co-regularised algorithms compared to single-view methods was already approached with variants of CoSVR, in particular, via Σ-CoSVR. Although Σ-CoSVR has the running time of a single-view algorithm, its prediction performance is worse than the one of base CoSVR. Future work should be directed towards a satisfactory trade-off between prediction performance and computation effort. Additionally, two-view semi-supervised algorithms like SVM-2K [Farquhar et al., 2005] or Σ-CoSVR point to further research topics. For example, SVM-2K is a semi-supervised support vector machine for classification based on two kernel functions and canonical correlation analysis (CCA). As CoSVR is restricted to regression tasks, the idea of co-regularisation could be transferred to classification or other learning tasks as well, e.g., via co-training [Blum and Mitchell, 1998]. Moreover, a generalisation of the two-view algorithms and their error bounds to arbitrary many data representations would be a beneficial contribution for MVL in the future. Finally, in Chapter 5 we considered orphan screening, the unsupervised case of ligand affinity prediction. We identified various future work topics on orphan screening both in the empirical and the theoretical direction. Firstly, the comparison of OPCA and the multi-view variants of CP and OPCA with the TLK approach and other baselines would be a canonical continuation of the present empirical results. Further approaches like structured output SVM or multi-output regression should be tested for orphan screening as well. An advantage of the transfer learning approaches CP and OPCA is their applicability to general primary learning tasks. Consequently, not only novel real-world applications from unsupervised regression, but also alternative unsupervised prediction tasks like classification could be tackled with the presented transfer learning approaches. The performance of single-view SVR approaches with very few labelled training ligands motivate to investigate the question, whether orphan screening with inclusion of related protein information should be favoured to supervised single-view approaches with too few training examples. Regarding algorithmic aspects, we suggest to additionally define and evaluate an ε-insensitive loss variant of CP analogous to the variants of MKL and CoSVR. In contrast to TLK, the CP and OPCA optmisation for the knowledge transfer from related proteins to the orphan protein has to be performed for every considered orphan target individually. A variant with universal orphan screening solution would be a useful completion in this research field.

Appendices

A Proofs

Appendix A contains long proofs of Chapters 3 and 4. The proof of Lemma 3.22 in Chapter 3 is a detailed version of the proof in [Cortes et al., 2009]. A proof for a classification scenario of Lemma 3.24 can be found in [Vishwanathan et al., 2010]. We transfer the proof of Vishwanathan et al. [2010] to the regression case. All three proofs for Chapter 4 presented in the appendix are original [Ullrich et al., 2016a, 2017]. They follow the same scheme shown for Lemma 4.6, but each with differences in the precise details. For the used variables also consult Table 4.1 above.

A.1 Proof of Lemma 3.22

Definition 3.21 (ℓ_2-MKL). [Cortes et al., 2009] Let $k_1, \ldots, k_M$ be kernel functions defined on an instance space $\mathcal{X}$ and k_b be the kernel linear combination according to Equation 3.21 with RKHS $\mathcal{H}_b$ and linear coefficients $b_1, \ldots, b_M \geq 0$. Let $(x_1, y_1), \ldots, (x_n, x_n)$ be labelled training examples from $\mathcal{X} \times \mathcal{Y}$. The optimisation

$$\min_{f \in \mathcal{H}_b,\ b \geq \mathbf{0}_M} \|f\|_{\mathcal{H}_b}^2 + \nu \sum_{i=1}^{n} |y_i - f(x_i)|^2, \qquad \text{s.t. } \|b - b^0\| \leq \Lambda \tag{3.25}$$

where $\nu, \Lambda > 0$ are hyperparameters and $b^0 \geq \mathbf{0}_M$ the initial linear coefficients, is called *ℓ_2-multiple kernel learning* (ℓ_2-MKL).

Lemma 3.22 *[Cortes et al., 2009] Let $K_b, K_1, \ldots, K_M \in \mathbb{R}^{n \times n}$ be the Gram matrices of the kernel functions $k, k_1, \ldots, k_M$ and Y be the vector of real-valued labels. For $\pi \in \mathbb{R}^n$ we put*

$$w = (\pi^T K_1 \pi, \ldots, \pi^T K_M \pi)^T.$$

The solution f of the minimisation in Equation 3.25 has got a representation in terms of b and π corresponding to Equation 3.23. For $b \geq \mathbf{0}_M$, $\pi \in \mathbb{R}^n$, and initial linear coefficients $b^0 \geq \mathbf{0}_M$

$$b = b^0 + \frac{w}{\|w\|}$$

holds true, where $\pi = (K_b + 1/\nu \cdot \mathbf{I}_n)^{-1} Y$.

Proof. According to Lemma 3.20, which is an MKL version of the representer theorem, the solution of Equation 3.25 has got a representation in form of

$$f(\cdot) = \sum_{i=1}^{n} \pi_i k_b(x_i, \cdot) = \sum_{i=1}^{n} \pi_i \sum_{v=1}^{M} b_v k_v(x_i, \cdot),$$

where $x_1, \ldots, x_n \in \mathcal{X}$ are the training instances. Analogous to the derivation of the solution of RLSR in Section 2.6.1, we can find a kernelised reformulation of the optimisation in Equation 3.25 with slack variables $\xi \in \mathbb{R}^n$

$$\begin{aligned} \min_{\pi, \xi \in \mathbb{R}^n, b \geq \mathbf{0}_M} \quad & \pi^T K_b \pi + \nu \xi^T \xi \\ \text{s.t. } & Y - K_b \pi = \xi \\ & \|b - b^0\| \leq \Lambda, \end{aligned}$$

where $K_b = \sum_{v=1}^{M} b_v K_v$, $b = (b_1, \ldots, b_M)^T$, and $\nu, \Lambda > 0$ and $b^0 \geq \mathbf{0}_M$ are the algorithm's hyperparameters and the initial linear coefficients. Firstly, we derive the dual problem with respect to the kernel expansion variables π by including the respective constraint into the objective with multipliers $\alpha \in \mathbb{R}^n$

$$\begin{aligned} \min_{\pi, \xi \in \mathbb{R}^n, b \geq \mathbf{0}_M} \max_{\alpha \in \mathbb{R}^n} \quad & \pi^T K_b \pi + \nu \xi^T \xi + \alpha^T (Y - K_b \pi - \xi) \\ \text{s.t. } & \|b - b^0\| \leq \Lambda. \end{aligned}$$

Secondly, if we put the Lagrangian's derivatives with respect to ξ and π to zero, we obtain

$$\xi = \frac{\alpha}{2\nu} \quad \text{and} \quad \alpha = 2\pi,$$

respectively. With $w(\pi) = (\pi^T K_1 \pi \ldots \pi^T K_M \pi)^T$ we obtain a min-max-problem via resubstitution

$$\min_{b \geq \mathbf{0}_M} \max_{\pi \in \mathbb{R}^n} \; - b^T w(\pi) - \frac{1}{\nu} \pi^T \pi + 2\pi^T Y \tag{6.1}$$

$$= \max_{\pi \in \mathbb{R}^n} \Big(\min_{b \geq \mathbf{0}_M} \; - b^T w(\pi) \Big) - \frac{1}{\nu} \pi^T \pi + 2\pi^T Y. \tag{6.2}$$

The equality of Equations 6.1 and 6.2 follows from *von Neumann's minimax theorem* [Kuhn and Tucker, 1958]. The application of von Neumann's minimax theorem allows for an initial consideration of the convex optimisation problem in b and to ignore that w is actually a function in π. The Lagrangian is

$$L = -w^T b - \beta^T b + \gamma(\|b - b^0\| - \Lambda),$$

where $\beta \geq \mathbf{0}_M$ and $\gamma \geq 0$ are the Lagrangian multipliers with respect to the inequality constraints for b. Thirdly, the derivative of $\partial L / \partial b$ put equal to zero together with the remaining KKT condition in Equation 2.13 yield

$$b = \frac{w + \beta}{2\gamma} + b^0, \tag{6.3}$$

$$0 = b^T \beta, \tag{6.4}$$

$$\text{and} \quad 0 = \gamma(\|b - b^0\| - \Lambda). \tag{6.5}$$

From the constraints of the original problem in Equation 3.25, we know that $\|b-b^0\| \leq \Lambda$ is valid. If $\|b-b^0\|$ was strictly smaller than Λ, we obtained $\gamma = 0$. This is impossible, because b must be as large as possible according to Equations 6.2 and 6.3. Hence, $\|b-b^0\| = \Lambda$ is valid. A combination of Equation 6.3 and Equation 6.4 shows that

$$-\|\beta\|^2 = (w + 2\gamma b_0)^T \beta \geq 0$$

and, consequently, $\beta = \mathbf{0}_M$. Finally, from $\|b-b^0\| = \Lambda$ and Equation 6.3 we obtain

$$b = b^0 + \Lambda \frac{w}{\|w\|}. \tag{6.6}$$

If we reclaim to the maximisation in Equation 6.1, we derive the RLSR solution in (2.26)

$$\pi = \left(K_b + \frac{1}{\nu}\mathbf{I}_n\right)^{-1} Y$$

with inverse regularisation parameter $\frac{1}{\nu}$ and the relation between b and w expressed in Equation 6.6. □

A.2 Proof of Lemma 3.24

Definition 3.23 (ε-MKL). [Vishwanathan et al., 2010] Let $\mathcal{H}_b$ be the RKHS of the kernel linear combination k_b from Equation 3.21, where $k_1, \ldots, k_M : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ are kernel functions. Let $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ be training examples. The optimisation

$$\min_{f \in \mathcal{H}_b} \frac{1}{2}\|f\|^2_{\mathcal{H}_b} + \nu \sum_{i=1}^{n} \max\{|y_i - f(x_i)| - \varepsilon, 0\} + \frac{\Lambda}{2}\|b\|_p^2 \tag{3.26}$$

is called *ε-multiple kernel learning* (ε-MKL), where $\varepsilon, \Lambda, \nu > 0$, $b \geq \mathbf{0}_M$ are hyperparameters.

Lemma 3.24 *[Vishwanathan et al., 2010] We consider the view-related kernel functions $k_1, \ldots, k_M$ and corresponding Gram matrices $K_1, \ldots, K_M \in \mathbb{R}^{n \times n}$. Additionally, let k_b be the reproducing kernel from Equation 3.21 with RKHS $\mathcal{H}_b$. Assume, for hyperparameters $p > 1$ and $q > 1$ the relation $\frac{1}{p} + \frac{1}{q} = 1$ holds true. The solution f of ε-MKL from Equation 3.26 has got a parameterisation in form of*

$$f(\cdot) = \sum_{i=1}^{n} \pi_i \sum_{v=1}^{M} b_v k_v(x_i, \cdot).$$

The parameters $b \geq \mathbf{0}_M$ and $\pi \in \mathbb{R}^n$ can be determined via the dual optimisation

$$\max_{\alpha, \hat{\alpha} \geq \mathbf{0}_n} \; -\frac{1}{8\Lambda}\left(\sum_{v=1}^{M} \left((\alpha - \hat{\alpha})^T K_v (\alpha - \hat{\alpha})\right)^q\right)^{\frac{2}{q}} + (\alpha - \hat{\alpha})^T Y - \varepsilon(\alpha + \hat{\alpha})\mathbf{1}_n,$$
$$\text{s. t. } \mathbf{0}_n \leq \alpha, \hat{\alpha} \leq \nu \mathbf{1}_n,$$

such that additionally

$$b_v = \frac{1}{2\Lambda}\left((\alpha-\hat{\alpha})^T K_v(\alpha-\hat{\alpha})\right)^{\frac{q}{p}} \left(\sum_{v=1}^{M}\left((\alpha-\hat{\alpha})^T K_v(\alpha-\hat{\alpha})\right)\right)^{\frac{1}{q}-\frac{1}{p}}, v = 1,\ldots,M,$$

and $\pi = \alpha - \hat{\alpha}$ *is valid.*

Proof. At first we reformulate the ε-MKL problem in the feature space of $\mathcal{H}_b$ with reproducing linear combination kernel k_b. The corresponding feature map Φ_b of k_b can be obtained with the concatenated and weighted features $\Phi_v : \mathcal{X} \to \mathbb{R}^{d_v}$ of the single kernels k_v

$$\Phi_b^T = (\sqrt{b_1}\Phi_1^T \cdots \sqrt{b_M}\Phi_M^T) : \mathcal{X} \to \mathbb{R}^d,$$

where $d = d_1 + \cdots + d_M$ (compare Definition 2.16). For the sake of simplicity, we omit an index b at dimension d. In the feature space of k_b, the minimisation problem we want to solve is to find a linear model $w_b^T = (w_1^T, \ldots, w_v^T) \in \mathbb{R}^d$ such that

$$\min_{w_b\in\mathbb{R}^d,\xi,\hat{\xi}\geq\mathbf{0}_n,b\geq\mathbf{0}_M} \frac{1}{2}w_b^T w_b + \nu\sum_{i=1}^{n}(\xi_i+\hat{\xi}_i) + \frac{\Lambda}{2}\|b\|_p^2$$
$$\text{s.t.} \left\{\begin{array}{rcl} y_i - w_b^T\Phi(x_i) & \leq & \varepsilon+\xi_i \\ w_b^T\Phi(x_i) - y_i & \leq & \varepsilon+\hat{\xi}_i \\ \xi_i,\hat{\xi}_i & \geq & 0 \end{array}\right\}_{i\in[\![n]\!]},$$

where $(x_1, y_1), \ldots, (x_n, x_n)$, are the labelled training examples and $\varepsilon, \nu, \Lambda > 0$ are hyper-parameters. We obtain a convex optimisation problem if we substitute w_v by $w_v/\sqrt{b_v}$

$$\min_{w_v\in\mathbb{R}^{d_v},\xi,\hat{\xi}\geq\mathbf{0}_n,b\geq\mathbf{0}_M} \frac{1}{2}\sum_{v=1}^{M}\frac{w_v^T w_v}{b_v} + \nu\sum_{i=1}^{n}(\xi_i+\hat{\xi}_i) + \frac{\Lambda}{2}(\sum_{v=1}^{M}b_v^p)^{\frac{2}{p}}$$
$$\text{s.t.} \left\{\begin{array}{rcl} y_i - \sum\limits_{v=1}^{M} w_v^T\Phi_v(x_i) & \leq & \varepsilon+\xi_i \\ \sum\limits_{v=1}^{M} w_v^T\Phi_v(x_i) - y_i & \leq & \varepsilon+\hat{\xi}_i \\ \xi_i,\hat{\xi}_i & \geq & 0 \end{array}\right\}_{i\in[\![n]\!]},$$

where we apply the definition of the ℓ_p-norm. Analogous to the proof for Lemma 3.22 above, we consider the minimisation problem with respect to w_v, ξ, and $\hat{\xi}$ first. The Lagrangian L with Lagrangian multipliers $\alpha, \hat{\alpha}, \beta, \hat{\beta} \geq \mathbf{0}_n$ is

$$\begin{aligned} L = {} & \frac{1}{2}\sum_{v=1}^{M}\frac{w_v^T w_v}{b_v} + \nu\sum_{i=1}^{n}(\xi_i+\hat{\xi}_i) + \frac{\Lambda}{2}\left(\sum_{v=1}^{M}b_v^p\right)^{\frac{2}{p}} - \sum_{i=1}^{n}(\beta_i\xi_i + \hat{\beta}_i\hat{\xi}_i) \\ & + \sum_{i=1}^{n}\alpha_i\left(y_i - \sum_{v=1}^{M}w_v^T\Phi_v(x_i) - \varepsilon - \xi_i\right) + \sum_{i=1}^{n}\hat{\alpha}_i\left(\sum_{v=1}^{M}w_v^T\Phi_v(x_i) - y_i - \varepsilon - \hat{\xi}_i\right). \end{aligned}$$

The derivatives of L with respect to w_v, ξ, and $\hat{\xi}$

$$\begin{aligned}\frac{\partial L}{\partial w_v} &= w_v/b_v - \sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i)\Phi_v(x_i) = 0\\ \frac{\partial L}{\partial \xi} &= \nu\mathbf{1}_n - \beta - \alpha = \mathbf{0}_n, \quad \frac{\partial L}{\partial \hat{\xi}} = \nu\mathbf{1}_n - \hat{\beta} - \hat{\alpha} = \mathbf{0}_n\end{aligned}$$

lead us to

$$w_v = b_v \sum_{i=1}^{n}(\alpha_i - \hat{\alpha}_i)\Phi_v(x_i) \quad \text{and} \quad \alpha, \hat{\alpha} \leq \nu\mathbf{1}_n. \tag{6.7}$$

Via resubstitution and with the Gram matrix

$$K_v = \left(\Phi_v^T(x_i)\Phi_v(x_j)\right)_{i,j=1}^{n}$$

of kernel k_v we obtain the following min-max-problem

$$\begin{aligned}\min_{b_v \geq 0} \max_{\alpha,\hat{\alpha} \geq \mathbf{0}_n} \; & -\frac{1}{2}\sum_{v=1}^{M} b_v(\alpha - \hat{\alpha})^T K_v(\alpha - \hat{\alpha}) + (\alpha + \hat{\alpha})^T Y \\ & - \varepsilon(\alpha + \hat{\alpha})^T \mathbf{1}_n + \frac{\Lambda}{2}\left(\sum_{v=1}^{M} b_v^p\right)^{\frac{2}{p}} \\ & \text{s.t. } \mathbf{0}_n \leq \alpha, \hat{\alpha} \leq \nu\mathbf{1}_n.\end{aligned}$$

Similar to the proceeding in the proof of Lemma 3.22 above, we apply von Neumann's minimax theorem, change the positions of min and max, and consider the minimisation with respect to the linear combination parameters $b_1, \ldots, b_M \geq 0$ first. The corresponding Lagrangian L with Lagrangian multipliers $\gamma \geq \mathbf{0}_M$ is

$$\begin{aligned}L = \; & -\frac{1}{2}\sum_{v=1}^{M} b_v(\alpha - \hat{\alpha})^T K_v(\alpha - \hat{\alpha}) + (\alpha + \hat{\alpha})^T Y - \varepsilon(\alpha + \hat{\alpha})^T \mathbf{1}_n \\ & + \frac{\Lambda}{2}\left(\sum_{v=1}^{M} b_v^p\right)^{\frac{2}{p}} - \sum_{v=1}^{M}\gamma_v b_v.\end{aligned} \tag{6.8}$$

From

$$\frac{\partial L}{\partial b_v} = \Lambda\left(\sum_{k=1}^{M} b_k^p\right)^{\frac{2}{p}-1} b_v^{p-1} - \gamma_v - \frac{1}{2}(\alpha - \hat{\alpha})^T K_v(\alpha - \hat{\alpha}) = 0$$

as well as $h_v = \gamma_v + \frac{1}{2}(\alpha - \hat{\alpha})^T K_v(\alpha - \hat{\alpha})$ and $B = B(p) = \sum_{k=1}^{M} b_k^p$ we conclude

$$\Lambda \; B^{\frac{2}{p}-1} b_v^{p-1} = h_v$$

for $v = 1, \ldots, M$. Consequently,

$$\Lambda \frac{B^{\frac{2}{p}}}{B}\sum_{v=1}^{M} b_v^p = \Lambda \; B^{\frac{2}{p}} = \sum_{v=1}^{M} b_v h_v \tag{6.9}$$

holds true as well. We apply the equality case of *Hölder's inequality* [Werner, 1995] and obtain

$$\sum_{v=1}^{M} b_v \cdot h_v = \left(\sum_{v=1}^{M} b_v^p\right)^{\frac{1}{p}} \cdot \left(\sum_{v=1}^{M} h_v^q\right)^{\frac{1}{q}} = B^{\frac{1}{p}} \left(\sum_{v=1}^{M} h_v^q\right)^{\frac{1}{q}}, \tag{6.10}$$

where $1/p + 1/q = 1$ and $p > 1$. The combination of Equations 6.9 and 6.10 leads us to

$$B^{\frac{2}{p}} = \frac{1}{\Lambda} \sum_{v=1}^{M} b_v h_v = \frac{1}{\Lambda} B^{\frac{1}{p}} \left(\sum_{v=1}^{M} h_v^q\right)^{\frac{1}{q}} = \frac{1}{\Lambda^2} \left(\sum_{v=1}^{M} h_v^q\right)^{\frac{2}{q}}. \tag{6.11}$$

In turn, Equation 6.8 together with Equation 6.11 imply a reformulated Lagrangian

$$\begin{aligned}
L &= (\alpha - \hat{\alpha})^T Y - \varepsilon(\alpha + \hat{\alpha})\mathbf{1}_n + \frac{\Lambda}{2} B^{\frac{2}{p}} - \sum_{v=1}^{M} b_v h_v \\
&= (\alpha - \hat{\alpha})^T Y - \varepsilon(\alpha + \hat{\alpha})\mathbf{1}_n - \frac{\Lambda}{2} B^{\frac{2}{p}} \\
&= (\alpha - \hat{\alpha})^T Y - \varepsilon(\alpha + \hat{\alpha})\mathbf{1}_n - \frac{1}{2\Lambda} \left(\sum_{v=1}^{M} h_v^q\right)^{\frac{2}{q}} \\
&= (\alpha - \hat{\alpha})^T Y - \varepsilon(\alpha + \hat{\alpha})\mathbf{1}_n - \frac{1}{2\Lambda} \left(\sum_{v=1}^{M} \left(\gamma_v + \frac{1}{2}(\alpha - \hat{\alpha})^T K_v (\alpha - \hat{\alpha})\right)^q\right)^{\frac{2}{q}}.
\end{aligned}$$

Finally, we obtain a dual problem in $\alpha, \hat{\alpha}$, and γ

$$\begin{aligned}
\max_{\gamma \geq \mathbf{0}_M, \alpha, \hat{\alpha} \geq \mathbf{0}_n} \; & (\alpha - \hat{\alpha})^T Y - \varepsilon(\alpha + \hat{\alpha})\mathbf{1}_n \\
& - \frac{1}{2\Lambda} \left(\sum_{v=1}^{M} \left(\gamma_v + \frac{1}{2}(\alpha - \hat{\alpha})^T K_v (\alpha - \hat{\alpha})\right)^q\right)^{\frac{2}{q}} \\
\text{s.t. } & \mathbf{0}_n \leq \alpha, \hat{\alpha} \leq \nu \mathbf{1}_n.
\end{aligned} \tag{6.12}$$

The optimal value of Equation 6.12 is taken on for $\gamma = \mathbf{0}_M$ as the ℓ_q-norm is strictly monotonically increasing and $\gamma_v \geq 0$. Hence, Equation 6.12 turns out to be an optimisation problem in the dual variables α and $\hat{\alpha}$ only

$$\begin{aligned}
\max_{\alpha, \hat{\alpha} \geq \mathbf{0}_n} \; & -\frac{1}{8\Lambda} \left(\sum_{v=1}^{M} \left((\alpha - \hat{\alpha})^T K_v (\alpha - \hat{\alpha})\right)^q\right)^{\frac{2}{q}} + (\alpha - \hat{\alpha})^T Y - \varepsilon(\alpha + \hat{\alpha})\mathbf{1}_n, \\
\text{s.t. } & \mathbf{0}_n \leq \alpha, \hat{\alpha} \leq \nu \mathbf{1}_n
\end{aligned}$$

which can be solved via the SMO algorithm (see Section 3.4). From Equation 6.9 we calculate the kernel linear factors b_v

$$\begin{aligned} b_v = & \left(h_v \frac{1}{\Lambda} B^{-(\frac{2}{p}-1)} \right)^{\frac{1}{p-1}} = \frac{1}{\Lambda} h_v^{\frac{q}{p}} \left(\sum_{v=1}^{M} h_v \right)^{\frac{1}{q}-\frac{1}{p}} \\ = & \frac{1}{2\Lambda} \left((\alpha - \hat{\alpha})^T K_v (\alpha - \hat{\alpha}) \right)^{\frac{q}{p}} \left(\sum_{v=1}^{M} \left((\alpha - \hat{\alpha})^T K_v (\alpha - \hat{\alpha}) \right) \right)^{\frac{1}{q}-\frac{1}{p}}. \end{aligned}$$

If we define

$$K_b^x = (k_b(x_1, x), \cdots, k_b(x_n, x)),$$

the actual predictor $f : \mathcal{X} \to \mathbb{R}$ equals

$$\begin{aligned} f(x) = & \ w^T \Phi(x) = \sum_{v=1}^{M} w_v^T \Phi_v(x) = \sum_{v=1}^{M} b_v \sum_{i=1}^{n} (\alpha_i - \hat{\alpha}_i) \Phi_v^T(x_i) \Phi_v(x) \\ = & \sum_{v=1}^{M} b_v \sum_{i=1}^{n} (\alpha_i - \hat{\alpha}_i) k_v(x_i, x) = K_b^x (\alpha - \hat{\alpha}), \end{aligned}$$

where we use Equation 6.7. □

A.3 Proof of Lemma 4.6

Definition 4.5 (CoSVR, ℓ_2-CoSVR, ε-CoSVR). For $v = 1, \ldots, M$ let $\mathcal{H}_v$ be an RKHS, ℓ^U be an arbitrary loss function, and $\varepsilon^L, \nu_v, \lambda > 0$ be hyperparameters. The optimisation problem in Equation 4.1 such that ℓ^L is the ε-sensitive loss with ε^L is called *co-regularised support vector regression* (CoSVR).
(i) Co-regularised support vector regression with $\ell^U = \ell_2$

$$\begin{aligned} \min_{f_v \in \mathcal{H}_v} \sum_{v=1}^{M} \left(\frac{\nu_v}{2} \|f_v\|_{\mathcal{H}_v}^2 + \sum_{i=1}^{n} \max\{|y_i - f_v(x_i)| - \varepsilon^L, 0\} \right) \\ + \lambda \sum_{u,v=1}^{M} \sum_{j=1}^{m} \|f_u(z_j) - f_v(z_j)\|^2, \end{aligned} \tag{4.7}$$

is denoted *ℓ_2-co-regularised support vector regression* (ℓ_2-CoSVR).
(ii) Co-regularised support vector regression where ℓ^U is the ε-insensitive loss

$$\begin{aligned} \min_{f_v \in \mathcal{H}_v} \sum_{v=1}^{M} \left(\frac{\nu_v}{2} \|f_v\|_{\mathcal{H}_v}^2 + \sum_{i=1}^{n} \max\{|y_i - f_v(x_i)| - \varepsilon^L, 0\} \right) \\ + \lambda \sum_{u,v=1}^{M} \sum_{j=1}^{m} \max\{|f_u(z_j) - f_v(z_j)| - \varepsilon^U, 0\} \end{aligned} \tag{4.8}$$

is called *ε-co-regularised support vector regression* (ε-CoSVR).

Lemma 4.6 *Let $\nu_v, \lambda, \varepsilon^L, \varepsilon^U > 0$. We use the notation introduced above. In particular, $\pi_v \in \mathbb{R}^{n+m}$ denote the kernel expansion coefficients of the view predictors f_v from*

Equation 4.3, whereas $\alpha_v, \hat{\alpha}_v \in \mathbb{R}^n$ and $\gamma_{uv} \in \mathbb{R}^m$ are dual variables.

(*i*) *The dual optimisation problem of ℓ_2-CoSVR is*

$$\max_{\alpha_v,\hat{\alpha}_v\in\mathbb{R}^n,\gamma_{uv}\in\mathbb{R}^m} \sum_{v=1}^{M} \left(-\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha_v - \hat{\alpha}_v)^T Y \right.$$
$$\left. -\varepsilon^L (\alpha_v + \hat{\alpha}_v)^T \mathbf{1}_n - \frac{1}{4\lambda} \sum_{u=1}^{M} \gamma_{uv}^T \gamma_{uv} \right)$$
$$\text{s. t.} \left\{ \begin{array}{l} \mathbf{0}_n \leq \alpha_v, \hat{\alpha}_v \leq \mathbf{1}_n \\ \gamma_{uv} = \frac{2\lambda}{\nu_u} U_u \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_u - \frac{2\lambda}{\nu_v} U_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v \end{array} \right\}_{(u,v)\in[\![M]\!]^2},$$

where

$$\begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v = \begin{pmatrix} \alpha_v - \hat{\alpha}_v \\ \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu}) \end{pmatrix}$$

and $\pi_v = \frac{1}{\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$ holds true.

(*ii*) *The dual optimisation problem of ε-CoSVR equals*

$$\max_{\alpha_v,\hat{\alpha}_v\in\mathbb{R}^n,\gamma_{uv}\in\mathbb{R}^m} \sum_{v=1}^{M} \left(-\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha_v - \hat{\alpha}_v)^T Y \right.$$
$$\left. -(\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n - \sum_{u=1}^{m} \gamma_{uv}^T \varepsilon^U \mathbf{1}_m \right)$$
$$\text{s. t.} \left\{ \begin{array}{l} \mathbf{0}_n \leq \alpha_v, \hat{\alpha}_v \leq \mathbf{1}_n \\ \mathbf{0}_m \leq \gamma_{uv} \leq \lambda \mathbf{1}_m \end{array} \right\}_{(u,v)\in[\![M]\!]^2},$$

where

$$\begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v = \begin{pmatrix} \alpha_v - \hat{\alpha}_v \\ \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu}) \end{pmatrix}$$

and $\pi_v = \frac{1}{\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$ are the kernel expansion coefficients from Equation 4.3.

Proof. (*i*) The view predictors f_v, $v = 1, \ldots, M$, have a representation as kernel expansion in the coefficients $\pi_v \in \mathbb{R}^{n+m}$ according to Lemma 4.2. With the slack variables $\xi_v, \hat{\xi}_v \in \mathbb{R}^n$ and $\zeta_{uv} \in \mathbb{R}^m$ we reformulate the kernelised version of ℓ_2-CoSVR as

$$\min_{\pi_v\in\mathbb{R}^{n+m}} \sum_{v=1}^{M} \left(\frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\xi_v + \hat{\xi}_v)^T \mathbf{1}_n + \lambda \sum_{u=1}^{M} \zeta_{uv}^T \zeta_{uv} \right)$$
$$\text{s.t.} \left\{ \begin{array}{rcl} Y - L_v \pi_v & \leq & \varepsilon^L \mathbf{1}_n + \xi_v \\ L_v \pi_v - Y & \leq & \varepsilon^L \mathbf{1}_n + \hat{\xi}_v \\ U_u \pi_u - U_v \pi_v & = & \zeta_{uv} \\ \xi_v, \hat{\xi}_v & \geq & \mathbf{0}_n \end{array} \right\}_{(u,v)\in[\![M]\!]^2}.$$

With Lagrangian multipliers $\alpha_v, \hat{\alpha}_v, \gamma_{uv}, \beta_v$, and $\hat{\beta}_v$ we obtain the corresponding Lagrangian

$$\begin{aligned} L = \sum_{v=1}^{M} \Bigg(& \frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\xi_v + \hat{\xi}_v)^T \mathbf{1}_n + \lambda \sum_{u=1}^{M} \zeta_{uv}^T \zeta_{uv} \\ & + \alpha_v^T (Y - L_v \pi_v - \varepsilon^L \mathbf{1}_n - \xi_v) + \hat{\alpha}_v^T (L_v \pi_v - Y - \varepsilon^L \mathbf{1}_n - \hat{\xi}_v) \\ & + \sum_{u=1}^{M} \gamma_{uv}^T (U_u \pi_u - U_v \pi_v - \zeta_{uv}) - \beta_v^T \xi_v - \hat{\beta}_v^T \hat{\xi}_v \Bigg) . \end{aligned}$$

The partial derivatives with respect to the slack variables yield the constraints $\mathbf{0}_n \leq \alpha_v, \hat{\alpha}_v \leq \mathbf{1}_n$ and the relation $\zeta_{uv} = \frac{1}{2\lambda} \gamma_{uv}$. The insertion into L leads us to

$$\begin{aligned} L = \sum_{v=1}^{M} \Bigg(& \frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\alpha_v - \hat{\alpha}_v)^T Y - (\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n - \sum_{u=1}^{M} \gamma_{uv}^T \, \varepsilon^U \mathbf{1}_m \\ & - (\alpha_v - \hat{\alpha}_v)^T L_v \pi_v - \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu})^T U_v \pi_v - \frac{1}{4\lambda} \sum_{u=1}^{M} \gamma_{uv}^T \gamma_{uv} \Bigg) . \end{aligned}$$

If we put $\partial L / \partial \pi_v = 0$, we deduce the relation between dual variables and the kernel expansion parameters

$$\pi_v = \frac{1}{\nu_v} \begin{pmatrix} \alpha_v - \hat{\alpha}_v \\ \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu}) \end{pmatrix} = \frac{1}{\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v . \tag{6.13}$$

The symbol

$$\begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v \in \mathbb{R}^{n+m}$$

is a stacked vector of view-dependent α- and γ-variables according to Equation 6.13. Furthermore, we obtain the dual objective

$$\begin{aligned} L = \sum_{v=1}^{M} \Bigg(& -\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha_v - \hat{\alpha}_v)^T Y \\ & - (\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n - \frac{1}{4\lambda} \sum_{u=1}^{M} \gamma_{uv}^T \gamma_{uv} \Bigg) . \end{aligned} \tag{6.14}$$

We introduced the extra symbol for the composed vector in Equation 6.13 in order to show analogies between the dual objective in Equation 6.14 and the dual objectives of related problems presented below. Finally, $\partial L / \partial \gamma_{uv} = 0$ leads to the equality constraints of the dual ℓ_2-CoSVR problem

$$\gamma_{uv} = \frac{2\lambda}{\nu_u} U_u \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_u - \frac{2\lambda}{\nu_v} U_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v .$$

(*ii*) Analogous to the proof of part (*i*) we reformulate ε-CoSVR with slack variables ξ_v, $\hat{\xi}_v \in \mathbb{R}^n$ and $\zeta_{uv} \in \mathbb{R}^m$ as follows

$$\min_{\pi_v \in \mathbb{R}^{n+m}} \sum_{v=1}^{M} \left(\frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\xi_v + \hat{\xi}_v)^T \mathbf{1}_n + \lambda \sum_{u=1}^{M} \zeta_{uv}^T \mathbf{1}_m \right)$$
$$\text{s.t.} \left\{ \begin{array}{rcl} Y - L_v \pi_v & \leq & \varepsilon^L \mathbf{1}_n + \xi_v \\ L_v \pi_v - Y & \leq & \varepsilon^L \mathbf{1}_n + \hat{\xi}_v \\ U_u \pi_u - U_v \pi_v & \leq & \varepsilon^U \mathbf{1}_m + \zeta_{uv} \\ \xi_v, \hat{\xi}_v & \geq & \mathbf{0}_n \\ \zeta_{uv} & \geq & \mathbf{0}_m \end{array} \right\}_{(u,v) \in [\![M]\!]^2} .$$

Introducing Lagrangian multipliers $\alpha_v, \hat{\alpha}_v, \gamma_{uv}, \beta_v, \hat{\beta}_v$, and δ_{uv} for the constraints in the order of appearance above, we obtain its Lagrangian

$$\begin{aligned} L = \sum_{v=1}^{M} \Bigg(& \frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\xi_v + \hat{\xi}_v)^T \mathbf{1}_n + \lambda \sum_{u=1}^{M} \zeta_{uv}^T \mathbf{1}_m \\ & + \alpha_v^T (Y - L_v \pi_v - \varepsilon^L \mathbf{1}_n - \xi_v) + \hat{\alpha}_v^T (L_v \pi_v - Y - \varepsilon^L \mathbf{1}_n - \hat{\xi}_v) \\ & + \sum_{u=1}^{M} \gamma_{uv}^T (U_u \pi_u - U_v \pi_v - \varepsilon^U \mathbf{1}_m - \zeta_{uv}) - \beta_v^T \xi_v - \hat{\beta}_v^T \hat{\xi}_v - \sum_{u=1}^{M} \delta_{uv}^T \zeta_{uv} \Bigg) . \end{aligned}$$

The partial derivatives of L with respect to ξ_v, $\hat{\xi}_v$, and ζ_{uv} put to zero lead us to

$$\begin{aligned} L = \sum_{v=1}^{M} \Bigg(& \frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\alpha_v - \hat{\alpha}_v)^T Y - (\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n - \sum_{u=1}^{M} \gamma_{uv}^T \, \varepsilon^U \mathbf{1}_m \\ & - (\alpha_v - \hat{\alpha}_v)^T L_v \pi_v - \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu})^T U_v \pi_v \Bigg) \end{aligned}$$

and the box constraints $\mathbf{0}_n \leq \alpha_v, \hat{\alpha}_v \leq \mathbf{1}_n$ as well as $\mathbf{0}_m \leq \zeta_{uv} \leq \lambda \mathbf{1}_m$. Finally, $\partial L / \partial \pi_v = 0$ for $v = 1, \ldots, M$ imply the relation

$$\pi_v = \frac{1}{\nu_v} \begin{pmatrix} \alpha_v - \hat{\alpha}_v \\ \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu}) \end{pmatrix} = \frac{1}{\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$$

of primal and dual variables. With the substitution of π_v into L we obtain the desired dual objective

$$\begin{aligned} L = \sum_{v=1}^{M} \Bigg(& -\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha_v - \hat{\alpha}_v)^T Y \\ & - (\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n - \sum_{u=1}^{M} \gamma_{uv}^T \varepsilon^U \mathbf{1}_m \Bigg) , \end{aligned}$$

which finishes the proof. □

A.4 Proof of Lemma 4.8

Definition 4.7 (CoSVR$^{\text{mod}}$). For loss functions ℓ^L and ℓ^U as well as hyperparameters $\nu_v, \lambda, \varepsilon^L > 0$, the co-regularised support vector regression problem with modified constraints for the labelled examples (CoSVR$^{\text{mod}}$) is defined as

$$\begin{aligned}\min_{f_v \in \mathcal{H}_v} \sum_{v=1}^{M} \frac{\nu_v}{2} \|f_v\|^2 + \sum_{i=1}^{n} \max\{|y_i - f^{\text{avg}}(x_i)| - \varepsilon^L, 0\} \\ + \lambda \sum_{u,v=1}^{M} \sum_{j=1}^{m} \ell^U \left(f_u(z_j), f_v(z_j)\right),\end{aligned} \tag{4.12}$$

where $f^{\text{avg}} = 1/M \sum_{v=1}^{M} f_v$ is the view predictor average from Equation 4.11. If ℓ^U is the ε-insensitive loss with $\varepsilon^U \geq 0$, the problem in Equation 4.12 is called ε-CoSVR$^{\text{mod}}$. The case $\ell^U = \ell_2$ is denoted with ℓ_2-CoSVR$^{\text{mod}}$.

Lemma 4.8 *Let $\nu_v, \lambda, \varepsilon^L, \varepsilon^U > 0$ be hyperparameters. We utilise dual variables $\alpha, \hat{\alpha} \in \mathbb{R}^n$ and $\gamma_{uv} \in \mathbb{R}^m$ (compare Table 4.1).*

(i) The ℓ_2-CoSVR$^{\text{mod}}$ dual optimisation problem equals

$$\begin{aligned}\max_{\alpha,\hat{\alpha} \in \mathbb{R}^n,\ \gamma_{uv} \in \mathbb{R}^m} \sum_{v=1}^{M} \Bigg(-\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha - \hat{\alpha})^T Y \\ -(\alpha + \hat{\alpha})^T \varepsilon^L \mathbf{1}_n - \frac{1}{4\lambda} \sum_{u=1}^{M} \gamma_{uv}^T \gamma_{uv} \Bigg) \\ \text{s. t.} \left\{ \begin{array}{l} \mathbf{0}_n \leq \alpha, \hat{\alpha} \leq \mathbf{1}_n \\ \gamma_{uv} = \frac{2\lambda}{\nu_u} U_u \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_u - \frac{2\lambda}{\nu_v} U_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v \end{array} \right\}_{v \in [\![M]\!]},\end{aligned}$$

where

$$\begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v = \begin{pmatrix} \frac{1}{M}(\alpha - \hat{\alpha}) \\ \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu}) \end{pmatrix}$$

and $\pi_v = \frac{1}{\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$.

(ii) The ε-CoSVR$^{\text{mod}}$ dual optimisation problem can be written as

$$\begin{aligned}\max_{\alpha,\hat{\alpha} \in \mathbb{R}^n,\ \gamma_{uv} \in \mathbb{R}^m} \sum_{v=1}^{M} \Bigg(-\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha - \hat{\alpha})^T Y \\ -(\alpha + \hat{\alpha})^T \varepsilon^L \mathbf{1}_n - \sum_{u=1}^{M} \gamma_{uv}^T \varepsilon^U \mathbf{1}_m \Bigg) \\ \text{s. t.} \left\{ \begin{array}{l} \mathbf{0}_n \leq \alpha, \hat{\alpha} \leq \mathbf{1}_n \\ \mathbf{0}_m \leq \gamma_{uv} \leq \lambda \mathbf{1}_m \end{array} \right\}_{v \in [\![M]\!]},\end{aligned}$$

where

$$\begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v = \begin{pmatrix}\frac{1}{M}(\alpha-\hat{\alpha})\\ \sum_{u=1}^{M}(\gamma_{uv}-\gamma_{vu})\end{pmatrix},$$

and $\pi_v = \frac{1}{\nu_v}\begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v$.

Proof. The representation of the solution functions as kernel linear combinations and, hence, the kernelised reformulation can be shown analogous to Lemma 4.2 and its corresponding proof. The remainder of the proof follows the same scheme as the proof of Lemma 4.6.

(*i*) Kernelised reformulation of ℓ_2-CoSVR^{mod} with slack variables $\xi, \hat{\xi}$, and ζ_{uv}:

$$\min_{\pi_v\in\mathbb{R}^{n+m}} \sum_{v=1}^{M}\left(\frac{\nu_v}{2}\pi_v^T K_v\pi_v + (\xi+\hat{\xi})^T\mathbf{1}_n + \lambda\sum_{u=1}^{M}\zeta_{uv}^T\zeta_{uv}\right)$$
$$\text{s.t.}\left\{\begin{array}{rcl} Y - \frac{1}{M}\sum_{v=1}^{M}L_v\pi_v & \leq & \varepsilon^L\mathbf{1}_n+\xi\\ \frac{1}{M}\sum_{v=1}^{M}L_v\pi_v - Y & \leq & \varepsilon^L\mathbf{1}_n+\hat{\xi}\\ U_u\pi_u - U_v\pi_v & = & \zeta_{uv}\\ \xi,\hat{\xi} & \geq & \mathbf{0}_n\end{array}\right\}_{(u,v)\in[\![M]\!]^2}$$

Introduction of Lagrangian multipliers $\alpha, \hat{\alpha}, \gamma_{uv}, \beta$, and $\hat{\beta}$:

$$\begin{aligned}L =& \sum_{v=1}^{M}\frac{\nu_v}{2}\pi_v^T K_v\pi_v + (\xi+\hat{\xi})^T\mathbf{1}_n + \lambda\sum_{u,v=1}^{M}\zeta_{uv}^T\zeta_{uv}\\ &+ \alpha^T(Y - \frac{1}{M}\sum_{v=1}^{M}L_v\pi_v - \varepsilon^L\mathbf{1}_n - \xi) + \hat{\alpha}^T(\frac{1}{M}\sum_{v=1}^{M}L_v\pi_v - Y - \varepsilon^L\mathbf{1}_n - \hat{\xi})\\ &+ \sum_{u,v=1}^{M}\gamma_{uv}^T(U_u\pi_u - U_v\pi_v - \zeta_{uv}) - \beta^T\xi - \hat{\beta}^T\hat{\xi}\end{aligned}$$

Equalities $\partial L/\partial\xi = 0, \partial L/\partial\hat{\xi} = 0$, and $\partial L/\partial\zeta_{uv} = 0$:

$$\begin{aligned}L =& \sum_{v=1}^{M}\frac{\nu_v}{2}\pi_v^T K_v\pi_v + (\alpha-\hat{\alpha})^T Y - (\alpha+\hat{\alpha})^T\varepsilon^L\mathbf{1}_n\\ &- (\alpha-\hat{\alpha})^T\frac{1}{M}\sum_{v=1}^{M}L_v\pi_v - \sum_{u,v=1}^{M}(\gamma_{uv}-\gamma_{vu})^T U_v\pi_v - \frac{1}{4\lambda}\sum_{u,v=1}^{M}\gamma_{uv}^T\gamma_{uv}\end{aligned}$$

and

$$\mathbf{0}_n \leq \alpha, \hat{\alpha} \leq \mathbf{1}_n \quad \text{as well as} \quad \gamma_{uv} = \frac{1}{2\lambda}\zeta_{uv}$$

Kernel expansion coefficients:

$$\pi_v = \frac{1}{\nu_v}\begin{pmatrix}\frac{1}{M}(\alpha-\hat{\alpha})\\ \sum_{u=1}^{M}(\gamma_{uv}-\gamma_{vu})\end{pmatrix} = \frac{1}{\nu_v}\begin{pmatrix}\alpha\\ \gamma\end{pmatrix}_v$$

Final dual objective:

$$L = \sum_{v=1}^{M} -\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha - \hat{\alpha})^T Y - (\alpha + \hat{\alpha})^T \varepsilon^L \mathbf{1}_n - \frac{1}{4\lambda} \sum_{u,v=1}^{M} \gamma_{uv}^T \gamma_{uv}$$

Additional equality constraints:

$$\gamma_{uv} = \frac{2\lambda}{\nu_u} U_u \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_u - \frac{2\lambda}{\nu_v} U_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$$

(ii) Kernelised reformulation of ε-CoSVR^{mod} with slack variables $\xi, \hat{\xi}$, and ζ_{uv}:

$$\min_{\pi_v \in \mathbb{R}^{n+m}} \sum_{v=1}^{M} \frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\xi + \hat{\xi})^T \mathbf{1}_n + \lambda \sum_{u,v=1}^{M} \zeta_{uv}^T \mathbf{1}_m$$

$$\text{s.t.} \left\{ \begin{array}{rcl} Y - \frac{1}{M} \sum_{v=1}^{M} L_v \pi_v & \leq & \varepsilon^L \mathbf{1}_n + \xi \\ \frac{1}{M} \sum_{v=1}^{M} L_v \pi_v - Y & \leq & \varepsilon^L \mathbf{1}_n + \hat{\xi} \\ U_u \pi_u - U_v \pi_v & \leq & \varepsilon^U \mathbf{1}_m + \zeta_{uv} \\ \xi, \hat{\xi} & \geq & \mathbf{0}_n \\ \zeta_{uv} & \geq & \mathbf{0}_m \end{array} \right\}_{(u,v) \in [\![M]\!]^2}$$

Introduction of Lagrangian multipliers $\alpha, \hat{\alpha}, \gamma_{uv}, \beta, \hat{\beta}$, and δ_{uv}:

$$\begin{aligned} L = & \sum_{v=1}^{M} \frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\xi + \hat{\xi})^T \mathbf{1}_n + \lambda \sum_{u,v=1}^{M} \zeta_{uv}^T \mathbf{1}_m \\ & + \alpha^T (Y - \frac{1}{M} \sum_{v=1}^{M} L_v \pi_v - \varepsilon^L \mathbf{1}_n - \xi) + \hat{\alpha}^T (\frac{1}{M} \sum_{v=1}^{M} L_v \pi_v - Y - \varepsilon^L \mathbf{1}_n - \hat{\xi}) \\ & + \sum_{u,v=1}^{M} \gamma_{uv}^T (U_u \pi_u - U_v \pi_v - \varepsilon^U \mathbf{1}_m - \zeta_{uv}) \\ & - \beta^T \xi - \hat{\beta}^T \hat{\xi} - \sum_{u,v=1}^{M} \delta_{uv}^T \zeta_{uv} \end{aligned}$$

Equalities $\partial L / \partial \xi = 0, \partial L / \partial \hat{\xi} = 0$, and $\partial L / \partial \zeta_{uv} = 0$:

$$\begin{aligned} L = & \sum_{v=1}^{M} \frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\alpha - \hat{\alpha})^T Y - (\alpha + \hat{\alpha})^T \varepsilon^L \mathbf{1}_n - \sum_{u,v=1}^{M} \gamma_{uv}^T \, \varepsilon^U \mathbf{1}_m \\ & - \frac{1}{M} \sum_{v=1}^{M} (\alpha - \hat{\alpha})^T L_v \pi_v - \sum_{u,v=1}^{M} (\gamma_{uv} - \gamma_{vu})^T U_v \pi_v \end{aligned}$$

and

$$\mathbf{0}_n \leq \alpha, \hat{\alpha} \leq \mathbf{1}_n \quad \text{as well as} \quad \mathbf{0}_m \leq \gamma_{uv} \leq \lambda \mathbf{1}_m$$

Kernel expansion coefficients:

$$\pi_v = \frac{1}{\nu_v} \begin{pmatrix} \frac{1}{M}(\alpha - \hat{\alpha}) \\ \sum_{u=1}^{M} (\gamma_{uv} - \gamma_{vu}) \end{pmatrix} = \frac{1}{\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$$

Final dual objective:

$$L = \sum_{v=1}^{M} -\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha - \hat{\alpha})^T Y - (\alpha + \hat{\alpha})^T \varepsilon^L \mathbf{1}_n - \sum_{u,v=1}^{M} \gamma_{uv}^T \varepsilon^U \mathbf{1}_m$$

□

A.5 Proof of Lemma 4.10

Definition 4.9 ($\text{CoSVR}_{\text{mod}}$). We consider RKHSs $\mathcal{H}_1, \ldots, \mathcal{H}_M$, loss functions ℓ^L and ℓ^U, as well as hyperparameters $\varepsilon^L, \nu_v, \lambda > 0$. The co-regularised support vector regression problem with modified constraints for the unlabelled examples ($\text{CoSVR}_{\text{mod}}$) is defined as

$$\begin{aligned} \min_{f_v \in \mathcal{H}_v} \sum_{v=1}^{M} \left(\frac{\nu_v}{2} \|f_v\|^2 + \sum_{i=1}^{n} \max\{|y_i - f_v(x_i)| - \varepsilon^L, 0\} \right) \\ + \lambda \sum_{v=1}^{M} \sum_{j=1}^{m} \ell^U \left(f_v^{\text{avg}}(z_j),\ f_v(z_j) \right), \end{aligned} \tag{4.13}$$

where $f_v^{\text{avg}} = 1/(M-1) \sum_{u=1}^{M, u \neq v} f_u$. If ℓ^U is the ε-insensitive loss with $\varepsilon^U > 0$ then the optimisation problem in Equation 4.13 is denoted with ε-$\text{CoSVR}_{\text{mod}}$ and the case $\ell^U = \ell_2$ with ℓ_2-$\text{CoSVR}_{\text{mod}}$.

Lemma 4.10 *Let $\nu_v, \lambda, \varepsilon^L, \varepsilon^U > 0$ be hyperparameters. We utilise dual variables $\alpha_v, \hat{\alpha}_v \in \mathbb{R}^n$ and $\gamma_v, \hat{\gamma}_v \in \mathbb{R}^m$, as well as $\gamma_v^{\text{avg}} = \frac{1}{M-1} \sum_{u=1}^{M, u \neq v} \gamma_u$ and $\hat{\gamma}_v^{avg} = \frac{1}{M-1} \sum_{u=1}^{M, u \neq v} \hat{\gamma}_u$ analogous to the residual view predictor average.*

(i) The ℓ_2-CoSVR$_{\text{mod}}$ dual optimisation problem equals

$$\begin{aligned} \max_{\alpha_v, \hat{\alpha}_v \in \mathbb{R}^n,\ \gamma_v \in \mathbb{R}^m} \sum_{v=1}^{M} \Bigg(& -\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha_v - \hat{\alpha}_v)^T Y \\ & -(\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n - \frac{1}{4\lambda} \sum_{u=1}^{M} \gamma_v^T \gamma_v \Bigg) \\ \text{s. t.} \left\{ \begin{array}{l} \mathbf{0}_n \leq \alpha_v, \hat{\alpha}_v \leq \mathbf{1}_n \\ \gamma_v = \frac{1}{M-1} \sum_{u=1}^{M, u \neq v} \frac{2\lambda}{\nu_u} U_u \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_u - \frac{2\lambda}{\nu_v} U_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v \end{array} \right\}_{v \in [\![M]\!]}, \end{aligned}$$

where

$$\begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v = \begin{pmatrix} \alpha_v - \hat{\alpha}_v \\ \gamma_v - \gamma_v^{\text{avg}} \end{pmatrix}$$

and $\pi_v = \frac{1}{\nu_v}\binom{\alpha}{\gamma}_v$.

(*ii*) *The* ε*-CoSVR*$_{\text{mod}}$ *dual optimisation problem can be written as*

$$\max_{\substack{\alpha_v,\hat{\alpha}_v\in\mathbb{R}^n,\ \gamma_v,\hat{\gamma}_v\in\mathbb{R}^m}} \sum_{v=1}^{M} \left(-\frac{1}{2\nu_v}\binom{\alpha}{\gamma}_v^T K_v \binom{\alpha}{\gamma}_v + (\alpha-\hat{\alpha})^T Y \right.$$
$$\left. -(\alpha_v+\hat{\alpha}_v)^T\varepsilon^L\mathbf{1}_n - (\gamma_v+\hat{\gamma}_v)\varepsilon^U\mathbf{1}_m \right)$$
$$\text{s.t.} \left\{ \begin{array}{l} \mathbf{0}_n \leq \alpha_v, \hat{\alpha}_v \leq \mathbf{1}_n \\ \mathbf{0}_m \leq \gamma_v, \hat{\gamma}_v \leq \lambda\mathbf{1}_m \end{array} \right\}_{v\in[\![M]\!]},$$

where

$$\binom{\alpha}{\gamma}_v = \begin{pmatrix} \alpha_v - \hat{\alpha}_v \\ (\gamma_v - \gamma_v^{\text{avg}}) - (\hat{\gamma}_v - \hat{\gamma}_v^{\text{avg}}) \end{pmatrix},$$

and $\pi_v = \frac{1}{\nu_v}\binom{\alpha}{\gamma}_v$.

Proof. The representation of the solution functions as kernel linear combinations and, hence, the kernelised reformulation can be shown analogous to Lemma 4.2 and its corresponding proof. The remainder of the proof follows the same scheme as the proof of Lemma 4.6.

(*i*) Kernelised reformulation of ℓ_2-CoSVR$_{mod}$ with slack variables $\xi_v, \hat{\xi}_v$, and ζ_v:

$$\min_{\pi_v\in\mathbb{R}^{n+m}} \sum_{v=1}^{M} \left(\frac{\nu_v}{2}\pi_v^T K_v \pi_v + (\xi_v+\hat{\xi}_v)^T\mathbf{1}_n + \lambda\zeta_v^T\zeta_v \right)$$
$$\text{s.t.} \left\{ \begin{array}{rcl} Y - L_v\pi_v & \leq & \varepsilon^L\mathbf{1}_n + \xi_v \\ L_v\pi_v - Y & \leq & \varepsilon^L\mathbf{1}_n + \hat{\xi}_v \\ \frac{1}{M-1}\sum_{u=1}^{M,u\neq v} U_u\pi_u - U_v\pi_v & = & \zeta_v \\ \xi_v, \hat{\xi}_v & \geq & \mathbf{0}_n \end{array} \right\}_{v\in[\![M]\!]}$$

Introduction of Lagrangian multipliers $\alpha_v, \hat{\alpha}_v, \gamma_v, \beta_v$, and $\hat{\beta}_v$:

$$L = \sum_{v=1}^{M} \left(\frac{\nu_v}{2}\pi_v^T K_v \pi_v + (\xi_v+\hat{\xi}_v)^T\mathbf{1}_n + \lambda\zeta_v^T\zeta_v \right.$$
$$+ \alpha_v^T\left(Y - L_v\pi_v - \varepsilon^L\mathbf{1}_n - \xi_v\right) + \hat{\alpha}_v^T\left(L_v\pi_v - Y - \varepsilon^L\mathbf{1}_n - \hat{\xi}_v\right)$$
$$\left. +\gamma_v^T\left(\frac{1}{M-1}\sum_{u=1}^{M,u\neq v} U_u\pi_u - U_v\pi_v - \zeta_v\right) - \beta_v^T\xi_v - \hat{\beta}_v^T\hat{\xi}_v \right)$$

Equations $\partial L/\partial \xi_v = 0, \partial L/\partial \hat{\xi}_v = 0$, and $\partial L/\partial \zeta_v = 0$:

$$\begin{aligned} L = \sum_{v=1}^{M} \Big(& \frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\alpha_v - \hat{\alpha}_v)^T Y \\ & - (\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n - (\alpha_v - \hat{\alpha}_v)^T L_v \pi_v \\ & - \gamma_v^T \Big(\frac{1}{M-1} \sum_{u=1}^{M,u\neq v} U_u \pi_u - U_v \pi_v \Big) - \frac{1}{4\lambda} \gamma_v^T \gamma_v \Big) \end{aligned}$$

and

$$\mathbf{0}_n \leq \alpha_v, \hat{\alpha}_v \leq \mathbf{1}_n \quad \text{as well as} \quad \gamma_v = \frac{1}{2\lambda} \zeta_v$$

Kernel expansion coefficients:

$$\pi_v = \frac{1}{\nu_v} \begin{pmatrix} \alpha_v - \hat{\alpha}_v \\ \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} \gamma_u - \gamma_v \end{pmatrix} = \frac{1}{\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$$

Final dual objective:

$$L = \sum_{v=1}^{M} \left(-\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha_v - \hat{\alpha}_v)^T Y - (\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n - \frac{1}{4\lambda} \gamma_v^T \gamma_v \right)$$

Additional equality constraints:

$$\gamma_v = \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} \frac{2\lambda}{\nu_u} U_u \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_u - \frac{2\lambda}{\nu_v} U_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$$

(*ii*) Kernelised reformulation of ε-CoSVR$_{mod}$ with slack variables $\xi_v, \hat{\xi}_v, \zeta_v$, and $\hat{\zeta}_v$:

$$\min_{\pi_v \in \mathbb{R}^{n+m}} \sum_{v=1}^{M} \Big(\frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\xi_v + \hat{\xi}_v)^T \mathbf{1}_n + \lambda (\zeta_v + \hat{\zeta}_v)^T \mathbf{1}_m \Big)$$

$$\text{s.t.} \left\{ \begin{array}{rcl} Y - L_v \pi_v & \leq & \varepsilon^L \mathbf{1}_n + \xi_v \\ L_v \pi_v - Y & \leq & \varepsilon^L \mathbf{1}_n + \hat{\xi}_v \\ \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} U_u \pi_u - U_v \pi_v & \leq & \varepsilon^U \mathbf{1}_m + \zeta_v \\ U_v \pi_v - \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} U_u \pi_u & \leq & \varepsilon^U \mathbf{1}_m + \hat{\zeta}_v \\ \xi_v, \hat{\xi}_v & \geq & \mathbf{0}_n \\ \zeta_v, \hat{\zeta}_v & \geq & \mathbf{0}_m \end{array} \right\}_{v \in [\![M]\!]}$$

Introduction of Lagrangian multipliers $\alpha_v, \hat{\alpha}_v, \gamma_v, \hat{\gamma}_v, \beta_v, \hat{\beta}_v, \delta_v$, and $\hat{\delta}_v$:

$$\begin{aligned} L = \sum_{v=1}^{M} \Big(& \frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\xi_v + \hat{\xi}_v)^T \mathbf{1}_n + \lambda(\zeta_v + \hat{\zeta}_v)^T \mathbf{1}_m \\ & + \alpha_v^T (Y - L_v \pi_v - \varepsilon^L \mathbf{1}_n - \xi_v) + \hat{\alpha}_v^T (L_v \pi_v - Y - \varepsilon^L \mathbf{1}_n - \hat{\xi}_v) \\ & + \gamma_v^T (\frac{1}{M-1} \sum_{u=1}^{M,u\neq v} U_u \pi_u - U_v \pi_v - \varepsilon^U \mathbf{1}_m - \zeta_v) \\ & + \hat{\gamma}_v^T (U_v \pi_v - \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} U_u \pi_u - \varepsilon^U \mathbf{1}_m - \hat{\zeta}_v) \\ & - \beta_v^T \xi_v - \hat{\beta}_v^T \hat{\xi}_v - \delta_v^T \zeta_v - \hat{\delta}_v^T \hat{\zeta}_v \Big) \end{aligned}$$

Equations $\partial L / \partial \xi_v = 0, \partial L / \partial \hat{\xi}_v = 0, \partial L / \partial \zeta_v = 0$, and $\partial L / \partial \hat{\zeta}_v = 0$:

$$\begin{aligned} L = \sum_{v=1}^{M} \Big(& \frac{\nu_v}{2} \pi_v^T K_v \pi_v + (\alpha_v - \hat{\alpha}_v)^T Y - (\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n \\ & - (\gamma_v + \hat{\gamma}_v)^T \ \varepsilon^U \mathbf{1}_m - (\alpha_v - \hat{\alpha}_v)^T L_v \pi_v \\ & - ((\gamma_v - \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} \gamma_u) - (\hat{\gamma}_v - \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} \hat{\gamma}_u)) U_v \pi_v \Big) \end{aligned}$$

and

$$\mathbf{0}_n \leq \alpha_v, \hat{\alpha}_v \leq \mathbf{1}_n \quad \text{and} \quad \mathbf{0}_m \leq \gamma_v \hat{\gamma}_v \leq \lambda \mathbf{1}_m$$

Kernel expansion coefficients:

$$\pi_v = \frac{1}{\nu_v} \begin{pmatrix} \alpha_v - \hat{\alpha}_v \\ (\gamma_v - \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} \gamma_u) - (\hat{\gamma}_v - \frac{1}{M-1} \sum_{u=1}^{M,u\neq v} \hat{\gamma}_u) \end{pmatrix} = \frac{1}{\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v$$

Final dual objective:

$$\begin{aligned} L = \sum_{v=1}^{M} \Big(& -\frac{1}{2\nu_v} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v^T K_v \begin{pmatrix} \alpha \\ \gamma \end{pmatrix}_v + (\alpha_v - \hat{\alpha}_v)^T Y \\ & - (\alpha_v + \hat{\alpha}_v)^T \varepsilon^L \mathbf{1}_n - (\gamma_v + \hat{\gamma}_v)^T \varepsilon^U \mathbf{1}_m \Big) \end{aligned}$$

□

B Ligand Affinity Dataset

The ligand affinity prediction experiments were performed with 29 protein-ligand datasets listed in Tables B.1 and B.2 together with further information on the sets. Each set corresponds to a human protein which can be identified via its Uniprot ID[1] in the *Protein ID* column of Tables B.1 and B.2. For a fixed protein, the set comprises a list of small molecular compounds and their binding affinity with respect to that protein. The molecules were gathered from the BindingDB database[2] and each set comprises between 21 and 2648 annotated compounds. The real-valued non-negative ligand affinities are pK_i-values that measure the strength of the protein-ligand complex (more details can be found in Section 1.3.1 on the biochemical background). As we consider a regression scenario, we call the molecular compounds *ligand (candidates)* independent of whether its corresponding affinity is high or low.

Different fingerprint formats are available for the representation of the small molecular compounds. We utilise the standard fingerprints *ECFP4*, *ECFP6*, *GpiDAPH3*, and *Maccs* which are described in Section 1.3.4. All of the applied fingerprints are binary and high-dimensional formats. A value of 1 at vectorial component c indicates that the respective compound carries the molecular feature associated with c and vice versa in case of value 0. The high dimension results from the fact that all of the used formats are a specific collection of a big number of molecular properties. As a consequence, many vectorial components will be equal to 0, which we also call a *sparse* representation (compare Section 4.3.4).

The dimension of a particular fingerprint type for the representation of instances of interest either results from the number of pre-defined molecular features (such as in the case of Maccs) or from the overall number of features found in the objects to represent (such as the graph patterns found in the union of all molecular compounds in all datasets in the case of ECFP4, ECFP6, or GpiDAPH3). Commonly generated datasets with respect to more than one protein exhibit the described initial fingerprint dimension. If one considers the sets for machine learning approaches individually (such as done in Chapters 3 and 4), the sparsity property mentioned above results in different *true dimensions* of the fingerprints for each of the protein-related datasets. With *true dimension* we refer to the number of fingerprint components that exhibit at least once a value of 1 and once a value of 0 in the respective set. This true dimension will typically scale with an increasing number of ligands. The true dimension of the 29 protein-ligand datasets together with their relative sparsity are listed in Table B.2 for all applied molecular fingerprint types. With relative sparsity we refer to the number of non-zero dimension components (sparsity) divided by the true dimension of the respective dataset. In both Tables B.1 and B.2 the proteins are ordered by increasing number of the comprised molecular compounds. The respective ordinal number can be found in parantheses behind the protein identifier in the *Protein ID* column. We observe that the true dimension increases if more compounds are included as more features can be found if the set of molecules expands. In contrast, the relative sparsity value decreases (by a factor of 20 in the case of ECFP6) with increasing number of molecular compounds which ultimately might hinder a machine learning algorithm to involve or detect a relevant feature in a data subset, for example, as a result of the train-test-split procedure.

[1] `www.uniprot.org`
[2] `www.bindingdb.org`

We used the fingerprint types ECFP6 and Maccs as well as graph patterns for the MKL experiments and protein-ligand sets $8-27$ in Chapter 3. For the empirical analysis of co-regularised algorithms and baselines in Chapter 4 we applied the standard molecular fingeprints ECFP4, GpiDAPH3, and Maccs as well as protein-ligand sets $1-11$, 13, $15-18$, $20-25$. In Chapter 5 we applied the fingerprints ECFP4 and GpiDAPH3 as well as protein-ligand sets $20-22$ and $24-29$. In all chapters, the datasets were chosen within an appropriate range of compound numbers. This range differed for the three chapters as we considered different algorithm classes and different settings with respect to data availability. Further differences occurred because of singular KKT matrices as an integral part of the QP solver used in the empirical section of Chapter 4. In Chapter 5, datasets happened to become too small because of the removal of ligands that belonged to more than one protein dataset. Moreover, in Chapter 5 we utilised a positive semi-definite similarity matrix as kernel values for the involved proteins. The similarity values for pairs of proteins came in addition to the original 29 protein-ligand sets and were originally calculated as a similarity measure for amino acid sequences.

Protein ID	**Ligand Number**	**Affinity Range**	**Protein ID**	**Ligand Number**	**Affinity Range**
P14091 (1)	21	$6.1-10.0$	P07384 (16)	189	$3.1-10.7$
P08311 (2)	23	$3.9-9.8$	P07339 (17)	197	$4.1-11.0$
Q16651 (3)	23	$4.8-7.9$	P08709 (18)	249	$3.9-9.5$
P07288 (4)	28	$7.2-9.7$	P43235 (19)	252	$3.9-11.5$
P04070 (5)	31	$3.7-7.1$	P00750 (20)	268	$2.2-9.5$
O60235 (6)	41	$5.8-7.9$	P07858 (21)	278	$3.0-10.5$
P03952 (7)	76	$3.0-9.3$	P29466 (22)	310	$3.1-9.8$
P23946 (8)	90	$5.4-8.9$	P07711 (23)	357	$3.9-10.6$
Q99895 (9)	91	$2.7-8.0$	P00747 (24)	474	$1.9-11.0$
P09871 (10)	92	$4.8-9.0$	P00749 (25)	600	$0.3-11.1$
P25774 (11)	104	$4.3-9.8$	P08246 (26)	742	$2.7-11.2$
Q9Y5Y6 (12)	125	$4.0-10.1$	P07477 (27)	986	$2.0-10.6$
P17655 (13)	128	$4.8-10.8$	P00742 (28)	2626	$3.0-11.4$
P42574 (14)	133	$4.9-11.9$	P00734 (29)	2648	$2.5-12.5$
P00740 (15)	171	$3.9-8.7$			

TABLE B.1: Ligand number and label range for protein-ligand datasets

TABLE B.2: True dimensions and relative sparsities of the ligand affinity datasets

Protein ID	**Dimension (Sparsity)**	**Dimension (Sparsity)**	**Dimension (Sparsity)**	**Dimension (Sparsity)**
Fingerprint	ECFP4	ECFP6	GpiDAPH3	Maccs
P14091 (1)	392 (0.22)	689 (0.18)	3017 (0.23)	113 (0.64)
P08311 (2)	543 (0.15)	967 (0.12)	3199 (0.14)	118 (0.52)
Q16651 (3)	232 (0.38)	410 (0.33)	2003 (0.37)	98 (0.65)
P07288 (4)	205 (0.33)	371 (0.28)	2482 (0.33)	91 (0.59)
P04070 (5)	501 (0.14)	888 (0.11)	1400 (0.10)	116 (0.64)
O60235 (6)	305 (0.24)	576 (0.19)	1809 (0.13)	102 (0.69)
P03952 (7)	881 (0.07)	1710 (0.05)	2848 (0.06)	125 (0.43)
P23946 (8)	649 (0.12)	1305 (0.09)	2255 (0.14)	127 (0.50)
Q99895 (9)	846 (0.07)	1661 (0.05)	2586 (0.08)	129 (0.40)
P09871 (10)	605 (0.10)	1277 (0.06)	913 (0.06)	118 (0.43)
P25774 (11)	882 (0.08)	1787 (0.05)	3344 (0.08)	124 (0.52)
Q9Y5Y6 (12)	748 (0.11)	1570 (0.08)	4461 (0.10)	124 (0.59)
P17655 (13)	658 (0.09)	1343 (0.07)	2865 (0.12)	114 (0.45)
P42574 (14)	969 (0.07)	1943 (0.05)	3297 (0.07)	135 (0.50)
P00740 (15)	1101 (0.06)	2383 (0.04)	2206 (0.05)	128 (0.38)
P07384 (16)	1045 (0.06)	2237 (0.04)	3788 (0.07)	121 (0.43)
P07339 (17)	1399 (0.06)	2957 (0.04)	7073 (0.08)	127 (0.49)
P08709 (18)	1420 (0.05)	3199 (0.03)	5362 (0.04)	137 (0.39)
P43235 (19)	1479 (0.04)	3259 (0.03)	4579 (0.05)	131 (0.43)
P00750 (20)	1961 (0.03)	4324 (0.02)	4757 (0.02)	136 (0.36)
P07858 (21)	1810 (0.03)	3875 (0.02)	5312 (0.05)	144 (0.37)
P29466 (22)	1667 (0.04)	3757 (0.03)	4110 (0.05)	131 (0.39)
P07711 (23)	1874 (0.04)	4185 (0.02)	5364 (0.05)	137 (0.42)
P00747 (24)	2316 (0.03)	5448 (0.02)	6152 (0.03)	140 (0.42)
P00749 (25)	2623 (0.02)	6446 (0.01)	6468 (0.02)	143 (0.33)
P08246 (26)	3716 (0.02)	8763 (0.01)	8439 (0.03)	150 (0.40)
P07477 (27)	4148 (0.02)	10373 (0.01)	8644 (0.02)	145 (0.39)
P00742 (28)	7433 (0.01)	20965 (0.005)	11671 (0.01)	150 (0.41)
P00734 (29)	7793 (0.01)	21641 (0.005)	13735 (0.02)	151 (0.42)

C Algorithms

C.1 A Heuristic to Detect Aromatic Bonds

A molecular graph representation, e.g., the SDF format introduced in Section 1.3.2, delivers information on atom and bond types and their inter-molecular arrangements. Special circular structures provide the corresponding molecules with particular properties regarding structure and physico-chemical behaviour known under the name of *aromaticity* (see Section 3.1.1). This molecular property cannot be seen directly in the graph representation, but reasoned iteratively from typical graph patterns. Algorithm 1 is a formal description of this deductive reasoning. We point out that the aromaticity of a molecular cycle affects the aromatic property of the neighbouring cycles. Additionally, we indicate that the presented algorithm is only a heuristic, i.e., it detects the vast majority of aromatic structures correctly. We will use the notation $\mathrm{DB}(c)$ for the number of bonds in a cycle c which are either double or aromatic bonds (labels *2* or *a*). The symbol $\mathrm{HA}(c)$ denotes the number of heteroatoms in cycle c. A heteroatom is a nitrogen, sulfur, or oxygen atom (labels *N*, *S*, or *O*) which is able to provide a free atom pair within a molecular cycle. The input of the subsequent algorithm is the molecular graph G with original labelling from a representation format like SDF. The output is the molecular graph G such that all bond labels in a cycle are changed to *aromatic* (label *a*), if the cycle is detected to be an aromatic one. We consider the simple cycles $\mathcal{S}(G)$ according to Section 3.1.3 and their subsets $\mathcal{S}^5(G)$ and $\mathcal{S}^6(G)$ of simple cycles with 5 and 6 nodes, respectively. The bond labels in $\mathcal{S}^5(G)$ and $\mathcal{S}^6(G)$ may change to *aromatic* (label *a*) throughout the algorithm's iterations. With C_i^a we refer to the set of detected aromatic cycles in iteration i.

Algorithm 1 Detection of aromatic bonds in a molecular graph G

Require: Molecular graph G with original labels of bonds (edges) and atoms (nodes)
Ensure: Molecular graph G with updated bond labels

$i \leftarrow 0$
$C_i^a \leftarrow \emptyset$
calculate $\mathcal{S}^5(G)$, $\mathcal{S}^6(G)$
repeat
    $i \leftarrow i + 1$
    $C_i^a \leftarrow C_{i-1}^a$
    for all $c \in \mathcal{S}^5(G) \setminus C_{i-1}^a$ **do**:
        if $(\mathrm{DB}(c) \geq 2)$ **or** $(\mathrm{DB}(c) = 1$ **and** $\mathrm{HA(c)} \geq 1)$ **then**
            update all edge labels of c in $\mathcal{S}^5(G)$ and $\mathcal{S}^6(G)$ to be a
            $C_i^a \leftarrow C_i^a \cup \{c\}$
        end if
    end for
    for all $c \in \mathcal{S}^6(G) \setminus C_{i-1}^a$ **do**:
        if $(\mathrm{DB}(c) \geq 3)$ **or** $(\mathrm{DB}(c) = 2$ **and** $\mathrm{HA}(c) \geq 1)$
                or $(\mathrm{DB}(c) = 1$ **and** $\mathrm{HA}(c) \geq 2)$ **then**
            update all edge labels of c in $\mathcal{S}^5(G)$ and $\mathcal{S}^6(G)$ to be a
            $C_i^a \leftarrow C_i^a \cup \{c\}$
        end if
    end for
until $|C_i^a| = |C_{i-1}^a|$

The numbers of double or aromatic bonds and heteroatoms in simple cycles are checked iteratively. A simple cycle is determined to be aromatic if the corresponding numbers fulfill certain conditions. For example a 6-cycle with three double bonds (label *2*) and three single bonds (label *1*) is classified to be aromatic and, consequently, all of its six bond labels are changed into *aromatic* (label *a*). The changed labels have an influence on the classification result of cycles in later iterations. The algorithm stops if no new aromatic cycles are found. Lines 7 and 13 of Algorithm 1 refer to the simple 5-cycles and 6-cycles c, respectively, that have not been detected to be aromatic in previous iterations.

C.2 Contracted Graph Construction

We consider the decomposition of a labelled, undirected graph G into biconnected components according to Definition 3.6 and bridges $\mathcal{B}(G)$. Algorithm 2 describes the construction of the contracted graph $\overline{G}$ from Section 3.1.4 as pseudocode. For every biconnected component B we add a new vertex v_B with label l_{bc} to the contracted graph as a representative of B. We denote the set of vertices contained in cycles of G with V_{cycles}. If $v \in V$ is a vertex of G, $\overline{v} = \lambda_{\overline{G}}(v) \in \overline{V}$ is the relabelled vertex of $\overline{G}$.

Algorithm 2 Contracted graph construction for the labelled undirected graph G

Require: Labelled, undirected graph $G = (V, E)$ and labelling function λ_G
Ensure: Contracted graph $\overline{G} = (\overline{V}, \overline{E})$ with labelling function $\lambda_{\overline{G}}$
1: $\overline{V} \leftarrow V \setminus V_{\text{cycles}}$
2: $\overline{E} \leftarrow \mathcal{B}(G)$
3: **for all** $v \in \overline{V}$ **do**:
4: $\quad \lambda_{\overline{G}}(v) \leftarrow \lambda_G(v)$
5: **end for**
6: **for all** biconnected components B **do**:
7: $\quad \overline{V} \leftarrow \overline{V} \cup \{v_B\}$
8: $\quad \lambda_{\overline{G}}(v_B) \leftarrow l_{\text{bc}}$
9: **end for**
10: **for all** $\{v, w\} \in \mathcal{B}(G)$ **do**:
11: $\quad \lambda_{\overline{G}}(\{v, w\}) \leftarrow \lambda_G(\{v, w\})$
12: **end for**

Regarding line 11 we refer to the comments in Section 3.1.4.

C.3 Iterative Solution of ℓ_2-MKL

Algorithm 3 delivers an iterative procedure to calculate the solution b and π of the ℓ_2-MKL optimisation presented in Section 3.3.1. In the pseudocode formulation the Gram matrix of the kernel linear combination k_b is $K_b = \sum_{v=1}^{M} b_v K_v$ for $b \geq \mathbf{0}_M$. Hence, the Gram matrix of the initial kernel linear combination is $K^0 = \sum_{v=1}^{M} (b^0)_v K_v$ for initial linear coefficients $b^0 \geq \mathbf{0}_M$. Regarding further details on the solution of ℓ_2-MKL and the properties of Algorithm 3 we refer to Cortes et al. [2009].

Algorithm 3 Solution of ℓ_2-MKL

Require: Gram matrices K_v, $v = 1, \ldots, M$, label vector Y, initial linear coefficients $b^0 \geq \mathbf{0}_M$, hyperparameters $\eta \in (0, 1)$, $\nu, \Lambda, \varepsilon > 0$

Ensure: Parameterisation $b \geq \mathbf{0}_M$ and $\pi \in \mathbb{R}^n$ of the predictor function

1: $\pi' \leftarrow (K^0 + \frac{1}{\nu}\mathbf{I}_n)^{-1}Y$
2: **repeat**
3: $\quad \pi \leftarrow \pi'$
4: $\quad w \leftarrow (\pi^T K_1 \pi, \ldots, \pi^T K_M \pi)^T$
5: $\quad b \leftarrow b^0 + \Lambda \frac{w}{\|w\|}$
6: $\quad \pi' \leftarrow \eta\pi + (1 - \eta)(K_b + \frac{1}{\nu}\mathbf{I}_n)^{-1}Y$
7: **until** $\|\pi - \pi'\| < \varepsilon$

C.4 Corresponding Projections Algorithm

Algorithm 4 is a formal description of the CP algorithm from Section 5.3.1 which outputs a hypothesis for an orphan target, i.e., a target without an available training set. For the algorithm below we assume that training examples $E_i \subseteq \mathcal{X} \times \mathcal{Y}$ are available for the so-called supervised targets t_i, $i = 1, \ldots, n$. With SKM we denote an arbitrary supervised kernel method that calculates a prediction model h_i from labelled instances E_i for target t_i. With CPO we refer to the CP optimisation in Equation 5.7.

Algorithm 4 Calculation of orphan hypothesis

Require: Training sets $E_1, \ldots, E_n$, targets $t_1, \ldots, t_n$, orphan target t_o

Ensure: Hypothesis h_o for the orphan target t_o

1: **for** $i = 1, \ldots, n$ **do**
2: $\quad h_i \leftarrow \mathrm{SKM}(E_i)$
3: **end for**
4: $h_o \leftarrow \mathrm{CPO}(t_o, t_1, \ldots, t_n, h_1, \ldots, h_n)$

Bibliography

W. A. Abbasi, F. U. Hassan, A. Yaseen, and F. U. A. A. Minhas. ISLAND: In-Silico Prediction of Proteins Binding Affinity Using Sequence Descriptors. ArXiv, 2017.

Q. U. Ain, A. Aleksandrova, F. D. Roessler, and P. J. Ballester. Machine-Learning Scoring Functions to Improve Structure-based Binding Affinity Prediction and Virtual Screening. *WIREs Computational Molecular Science*, 5:405–424, 2015.

N. Aronszajn. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68, 1950.

M. Awad and R. Khanna. Support Vector Regression. In *Efficient Learning Machines*, pages 67–80, 2015.

F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple Kernel Learning, Conic Duality, and the SMO Algorithm. In *Proceedings of the Twentyfirst International Conference on Machine Learning*, pages 41–48, 2004.

M.-F. Balcan and A. Blum. A PAC-Style Model for Learning from Labeled and Unlabeled Data. In *Proceedings of the Eighteenths Annual Conference on Learning Theory*, pages 111–126, 2005.

J. Balfer and J. Bajorath. Systematic Artifacts in Support Vector Regression-Based Compound Potency Prediction Revealed by Statistical and Activity Landscape Analysis. *PLoS ONE*, 10, 2015.

P. J. Ballester and J. B. O. Mitchell. A Machine Learning Approach to Predicting Protein-Ligand Binding Affinity with Applications to Molecular Docking. *BMC Bioinformatics*, 26:1169–1175, 2010.

R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A Simple Proof of the Restricted Isometry Property for Random Matrices. *Constructive Approximation*, 28:253–263, 2008.

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research*, 3:463–482, 2002.

A. Bender and R. C. Glen. Molecular Similarity: A Key Technique in Molecular Informatics. *Organic & Biomolecular Chemistry*, 2:3204–3218, 2004.

A. Bender, J. L. Jenkins, J. Scheiber, S. C. K. Sukuru, M. Glick, and J. W. Davies. How Similar Are Similarity Searching Methods? A Principal Component Analysis of Molecular Descriptor Space. *Journal of Chemical Information and Modeling*, 49: 108–119, 2009.

K. P. Bennett and A. Demiriz. Semi-Supervised Support Vector Machines. In *Advances in Neural Information Processing Systems*, pages 368–374, 1998.

M. R. Berthold, C. Borgelt, F. Höppner, and F. Klawonn. *Guide to Intelligent Data Analysis*. Springer Verlag, 2010.

A. Blum and T. Mitchell. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100, 1998.

J. R. Bock and D. A. Gough. A New Method to Estimate Ligand-Receptor Energetics. *Molecular & Cellular Proteomics*, 1:904–910, 2002.

J. R. Bock and D. A. Gough. Virtual Screen for Ligands of Orphan G Protein-Coupled Receptors. *Journal of Chemical Information and Modeling*, 45:1402–1414, 2005.

K. Borgwardt and H.-P. Kriegel. Shortest-Path Kernels on Graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 74–81, 2005.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

U. Brefeld and T. Scheffer. Semi-Supervised Learning for Structured Output Variables. In *Proceedings of the Twenty–Third International Conference of Machine Learning*, 2006.

U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. Efficient Co-Regularised Least Squares Regression. In *Proceedings of the Twenty–Third International Conference of Machine Learning*, pages 137–144, 2006.

R. D. Burbidge, M. Trotter, B. F. Buxton, and S. B. Holden. Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis. *Computers & Chemistry*, 26:5–14, 2001.

C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

C. Z. Cai, W. L. Wang, L. Z. Sun, and Yu Z. Chen. Protein Function Classification via Support Vector Machine Approach. *Mathematical Biosciences*, 185:111–122, 2003.

R. Caruana. Multitask Learning. *Machine Learning*, 28:41–75, 1997.

A. B. Chan, N. Vasconcelos, and G. R. G. Lanckriet. Direct Convex Relaxations of Sparse SVM. 2007.

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.

O. Chapelle, V. Sindhwani, and S. S. Keerthi. Optimization Techniques for Semi-Supervised Support Vector Machines. *Journal of Machine Learning Research*, 9:203–233, 2008.

A. Cherkasov, E. N. Muratov, D. Fourches, A. Varnek, I. I. Baskin, M. Cronin, J. Dearden, P. Gramatica, Y. C. Martin, R. Todeschini, V. Consonni, V. E. Kuz'min, R. Cramer, R. Benigni, C. Yang, J. Rathman, L. Terfloth, J. Gasteiger, A. Richard, and A. Tropsha. QSAR Modeling: Where Have You Been? Where Are You Going to? *Journal of Medical Chemistry*, 57:4977–5010, 2013.

V. Cherkassky and F. Mulier. *Learning from Data – Concepts, Theory, and Methods.* Wiley, 1998.

C. M. Christoudias, R. Urtasun, and T. Darrell. Multi-View Learning in the Presence of View Disagreement. In *Proceedings of the Twenty–Fourth Conference on Uncertainty in Artificial Intelligence*, pages 88–96, 2008.

M. Collins and N. Duffy. Convolution Kernels for Natural Language. In *Proceedings of the Fourteenth International Conference on Neural Information Processing Systems*, pages 625–632, 2001.

C. Cortes, M. Mohri, and A. Rostaminzadeh. L_2 Regularization for Learning Kernels. In *Proceedings of the Twenty–Fifth Conference on Uncertainty in Artificial Intelligence*, pages 109–116, 2009.

C. Cortes, M. Mohri, and A. Rostaminzadeh. Generalization Bounds for Learning Kernels. In *Proceedings of Twentyseventh International Conference on Machine Learning*, 2010.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, 2000.

F. Cucker and D. X. Zhou. *Learning Theory – An Approximation Theory Viewpoint.* Cambridge University Press, 2007.

S. Dasgupta and A. Gupta. An Elementary Proof of the Johnson-Lindenstrauss Lemma. *Random Structures & Algorithms*, 22:60–65, 2003.

S. Dasgupta, M. L. Littman, and D. McAllester. PAC Generalization Bounds for Co-Training. In *Advances in Neural Information Processing Systems*, pages 375–82, 2002.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

T. Diethe, D. R. Hardoon, and J. Shawe-Taylor. Multiview Fisher Discriminant Analysis. In *Proceedings of the NIPS Workshop on Learning from Multiple Sources*, 2008.

H. Ding, I. Takigawa, H. Mamitsuka, and S. Zhu. Similarity-Based Machine Learning Methods for Predicting Drug-Target Interactions: A Brief Review. *Briefings in Bioinformatics*, 15:734–747, 2013.

J. L. Durant, B. A. Leland, D. R. Henry, and J. G. Nourse. Reoptimization of MDL Keys for Use in Drug Discovery. *Journal of Chemical Information and Computer Sciences*, 42:1273–1280, 2002.

D. Erhan, P.-J. L'Heureux, S. Y. Yue, and Y. Bengio. Collaborative Filtering on a Family of Biological Targets. *Journal of Chemical Information Modeling*, 46:626–635, 2006.

J. D. R. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak. Two View Learning: SVM-2K, Theory and Practice. In *Advances in Neural Information Processing Systems*, pages 355–362, 2005.

L. L. G. Ferreira and A. D. Andricopulo. From Chemoinformatics to Deep Learning: an Open Road to Drug Discovery. *Future Medicinal Chemistry*, 11:371—-374, 2019.

P. Flach. *Machine Learning*. Cambridge University Press, 2012.

R. W. Floyd. Shortest Path. *Communications of the Association for Computing Machinery*, 5:345, 1962.

D. P. Foster, S. M. Kakade, and T. Zhang. Multi-View Dimensionality Reduction via Canonical Correlation Analysis. Technical report, Toyota Technological Institute Chicago, 2008.

H. Fröhlich, J. K. Wegner, F. Sieker, and A. Zell. Kernel Functions for Attributed Molecular Graphs – A New Similarity-Based Approach to ADME Prediction in Classification and Regression. *QSAR & Combinatorial Sciences*, 25:317–326, 2005.

T. Gärtner. A Survey of Kernels for Structured Data. *ACM SIGKDD Explorations Newsletters*, 5:49–58, 2003.

T. Gärtner, P. Flach, and S. Wrobel. On Graph Kernels: Hardness Results and Efficient Alternatives. In *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory*, pages 129–143, 2003.

B. Gaüzère, P.-A. Grenier, L. Brun, and D. Villemin. Treelet Kernel Incorporating Cyclic, Stereo and Inter Pattern Information in Chemoinformatics. *Pattern Recognition*, 48:356–367, 2014.

H. Geppert, T. Horváth, T. Gärtner, S. Wrobel, and J. Bajorath. Support-Vector-Machine-Based Ranking Significantly Improves the Effectiveness of Similarity Searching Using 2D Fingerprints and Multiple Reference Compounds. *Journal of Chemical Information and Modeling*, 48:742–746, 2008.

H. Geppert, J. Humrich, D. Stumpfe, T. Gärtner, and J. Bajorath. Ligand Prediction from Protein Sequence and Small Molecule Information Using Support Vector Machines and Fingerprint Descriptors. *Journal of Chemical Information and Modeling*, 49:767–779, 2009.

H. Geppert, M. Vogt, and J. Bajorath. Current Trends in Ligand-Based Virtual Screening: Molecular Representations, Data Mining Methods, New Application Areas, and Performance Evaluation. *Journal of Chemical Information and Modeling*, 50:205–216, 2010.

S. Giesselbach, K. Ullrich, M. Kamp, D. Paurat, and T. Gärtner. Corresponding Projections for Orphan Screening. In *Proceedings of the NIPS Workshop on Machine Learning for Health (ML4H)*, 2018.

M. Gönen and E. Alpaydin. Multiple Kernel Learning Algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

C. Hansch, P. P. Maloney, T. Fujita, and R. M. Muir. Correlation of Biological Activity of Phenoxyacetic Acids with Hammet Substituent Constants and Partition Coefficients. *Nature*, 194:178–180, 1962.

D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical Correlation Analyis: An Overview with Application to Learning Methods. *Neural Computation*, 16:2639–2664, 2004.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Verlag, 2001.

K. Heikamp. *Application and Development of Computational Methods for Ligand-Based Virtual Screening*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2014.

K. Heikamp and J. Bajorath. Fingerprint Design and Engineering Strategies: Rationalizing and Improving Similarity Search Performance. *Future Medicinal Chemistry*, 4: 1945–1959, 2012.

K. Heikamp and J. Bajorath. Support Vector Machines for Drug Discovery. *Expert Opinion on Drug Discovery*, 9:93–104, 2014.

H. Herold, B. Lurz, and J. Wohlrab. *Grundlagen der Informatik – Praktisch – Technisch – Theoretisch*. Pearson Studium, 2007.

T. Horváth. Cyclic Pattern Kernels Revisited. In *Advances in Knowledge Discovery and Data Mining*, pages 791–801, 2005.

T. Horváth, T. Gärtner, and S. Wrobel. Cyclic Pattern Kernels for Predictive Graph Mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 158–167, 2004.

H. Hotelling. Relations between Two Sets of Variates. *Biometrika*, 28:321–377, 1936.

J. D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9:90–95, 2007.

J. J. Irwin. Community Benchmarks for Virtual Screening. *Journal of Computer–Aided Molecular Design*, 22:193–199, 2008.

L. Jacob and J.-P. Vert. Protein-Ligand Interaction Prediction: An Improved Chemogenomics Approach. *BMC Bioinformatics*, 24:2149–2156, 2008.

L. Jacob, B. Hoffmann, V. Stoven, and J.-P. Vert. Virtual Screening of GPCRs: An in Silico Chemogenomics Approach. *BMC Bioinformatics*, 9:363–378, 2008.

Y.-S. Ji, J.-J. Chen, G. Niu, L. Shang, and X.-Y. Dai. Transfer Learning via Multi-View Principal Component Analysis. *Journal of Computer Science and Technology*, 26: 81–98, 2011.

J. Jiménez, M. Škalič, G. Martínez-Rosell, and G. De Fabritiis. K_{DEEP}: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *Journal of Chemical Information and Modeling*, 58:287–296, 2018.

T. Joachims. Transductive Inference for Text Classification Using Support Vector Machines. In *Proceedings of the International Conference of Machine Learning*, pages 200–209, 1999.

S. M. Kakade and D. P. Foster. Multi-View Regression via Canonical Correlation Analysis. In *Proceedings of the Twentieth Annual Conference on Learning Theory*, pages 82–86, 2007.

A. Kaplan and M. Haenlein. Siri, Siri, in My Hand: Who's the Fairest in the Land? On the Interpretations, Illustrations, and Implications of Artificial Intelligence. *Business Horizons*, 62:15–25, 2019.

H. Kashima, K. Tsuda, and A. Inokuchi. *Kernels for Graphs*, pages 155–170. MIT Press, 2004. In Kernel Methods in Computational Biology (Editors B. Schölkopf, K. Tsuda, and J.-P. Vert).

A. Klenke. *Wahrscheinlichkeitstheorie*. Springer Verlag, 2006.

M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien. Efficient and Accurate ℓ_p-Norm Multiple Kernel Learning. In *Advances in Neural Information Processing Systems*, pages 997–1005, 2009.

M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. ℓ_p-Norm Multiple Kernel Learning. *Journal of Machine Learning Research*, 12:953–997, 2011.

T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing, and Jupyter Development Team. Jupyter Notebooks – A Publishing Format for Reproducible Computational Workflows. In *Proceedings of the Twentieth International Conference on Electronic Publishing*, pages 87–90, 2016.

R. I. Kondor and J. Lafferty. Diffusion Kernels on Graphs and Other Discrete Input Spaces. In *Proceedings of the International Conference on Machine Learning*, pages 315–322, 2002.

E. Kondratovich, I. I. Baskin, and A. Varnek. Transductive Support Vector Machines: Promising Approach to Model Small and Unbalanced Datasets. *Molecular Informatics*, 32:261–266, 2013.

A. Koutsoukasa, S. Paricharak, W. R. J. D. Galloway, D. R. Spring, A. P. Ijzerman, R. C. Glen, D. Marcus, and A. Bender. How Diverse Are Diversity Assessment Methods? A Comparative Analysis and Benchmarking of Molecular Descriptor Space. *Journal of Chemical Information and Modeling*, 54:230–242, 2013.

M. K. Kozlov, S. P. Karasov, and L. G. Khachiyan. The Polynomial Solvability of Convex Quadratic Programming. *USSR Computational Mathematics and Mathematical Physics*, 20:223–228, 1980.

H. W. Kuhn and A. W. Tucker. John von Neumann's Work in the Theory of Games and Mathematical Economics. *Bulletin of the American Mathematical Society*, 64: 100–122, 1958.

I. Kundu, G. Paul, and R. Banerjee. A Machine Learning Approach Towards the Prediction of Protein-Ligand Binding Affinity Based on Fundamental Molecular Properties. *Royal Society of Chemistry Advances*, 8:12127–12137, 2018.

G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A Statistical Framework for Genomic Data Fusion. *Bioinformatics*, 20:2626–2635, 2004a.

G. R. G. Lanckriet, N. Christianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the Kernel Matrix with Semidefinite Programming. *Journal of Machine Learning Research*, 5:27–72, 2004b.

G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-Based Data Fusion and Its Application to Protein Function Prediction in Yeast. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 300–311, 2004c.

B. Leskes. The Value of Agreement, a New Boosting Algorithm. In *Proceedings of the Eighteenth Annual Conference on Learning Theory*, pages 95–110, 2005.

H. Li, K.-S. Leung, M.-H. Wong, and P. J. Ballester. Substituting Random Forest for Multiple Linear Regression Improves Binding Affinity Prediction of Scoring Functions: Cyscore as a Case Study. *BMC Bioinformatics*, 15:291–302, 2014.

L. Li, B. Wang, and S. O. Meroueh. Support Vector Regression Scoring of Receptor-Ligand Complexes for Rank-Ordering and Virtual Screening of Chemical Libraries. *Journal of Chemical Information and Modeling*, 51:2132–2138, 2011.

Q. Liu, C. K. Kwoh, and J. Li. Binding Affinity Prediction for Protein-Ligand Complexes Based on β Contacts and B Factor. *Journal of Chemical Information and Modeling*, 53:3076–3085, 2013.

W. Liu, X. Meng, Q. Xu, D. R. Flower, and T. Li. Quantitative Prediction of Mouse Class I MHC Peptide Binding Affinity Using Support Vector Machine Regression (SVR) Models. *BMC Bioinformatics*, 7:182–194, 2006.

Y.-C. Lo, S. E. Rensi, W. Torng, and R. B. Altmann. Machine Learning in Chemoinformatics and Drug Discovery. *Drug Discovery Today*, 23:1538–1546, 2018.

J. Mack. A Graph Pattern Kernel for Ligand Prediction. Master's thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2014.

D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

P. Mahé and J.-P. Vert. Virtual Screening with Support Vector Machines and Structure Kernels. *Combinatorial Chemistry & High Throughput Screening*, 12:409–423, 2009.

E. T. Matsubara, M. C. Monard, and G. E. A. P. A. Batista. An Approach to Obtain Different Views from Text Datasets. pages 97–104, 2005.

A. Maunz and C. Helma. Predition of Chemical Toxicity with Local Support Vector Regression and Activity-Specific Kernels. *SAR and QSAR in Environmental Research*, 19:413–431, 2008.

J. L. Medina-Franco, K. Martínez-Mayorga, A. Bender, R. M. Marín, M. A. Giulianotti, C. Pinilla, and R. A. Houghten. Characterization of Activity Landscapes Using 2D and 3D Similarity Methods: Consensus Activity Cliffs. *Journal of Chemical Information and Modeling*, 49:477–491, 2009.

L. Michielan and S. Moro. Pharmaceutical Perspectives of Nonlinear QSAR Strategies. *Journal of Chemical Information Modelling*, 50:961–978, 2010.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher Discriminant Analysis with Kernels. In *IEEE Proceedings of Neural Networks for Signal Processing*, pages 41–48, 1999.

H. Q. Minh, P. Niyogi, and Y. Yao. Mercer's Theorem, Feature Maps, and Smooting. In *Proceedings of the International Conference on Learning Theory*, pages 154–168, 2006.

T. M. Mitchell. *Machine Learning*. McGraw–Hill, 1997.

D. Nelson and M. Cox. *Lehninger Biochemie*. Springer Verlag, 2001.

K. Nigam and R. Ghani. Analyzing the Effectiveness and Applicability of Co-Training. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pages 86–93, 2000.

X. Ning, H. Rangwala, and G. Karypis. Multi-Assay-Based Structure-Activity-Relationship Models: Improving Structure-Activity-Relationship Models by Incorporating Activity Information from Related Targets. *Journal of Chemical Information and Modeling*, 49:2444–2456, 2009.

B. Nisius. *Development of Novel Fingerprint Engineering Methods Adressing Principal Complications of Similarity Searching*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2010.

B. Nisius and J. Bajorath. Molecular Fingerprint Recombination: Generating Hybrid Fingerprints for Similarity Searching from Different Fingerprint Types. *ChemMedChem*, 4:1859–1863, 2009.

B. Nisius and J. Bajorath. Reduction and Recombination of Fingerprints of Different Design Increase Compound Recall and the Structural Diversity of Hits. *Chemical Biology & Drug Design*, 75:152–160, 2010.

D. Oglic, D. Paurat, and T. Gärtner. Interactive Knowledge-Based Kernel PCA. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 501–516, 2014.

L. Oneto, S. Ridella, and D. Anguita. Tikhonov, Ivanov and Morozov Regularization for Support Vector Machine Learning. *Machine Learning*, 103:103–136, 2016.

A. R. Ortiz, M. T. Pisabarro, F. Gago, and R. C. Wade. Prediction of Drug Binding Affinities by Comparative Binding Energy Analysis. *Journal of Medicinal Chemistry*, 38:2681–2691, 1995.

H. Öztürk, E. Ozkirimli, and A. Özgür. DeepDTA: Deep Drug-Target Binding Affinity Prediction. *Bioinformatics*, 34, 2018.

S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010.

J. C. Platt. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In *Advances in kernel methods: support vector learning*, pages 185–208, 1999.

S. Qiu and T. Lane. Multiple Kernel Support Vector Regression for siRNA Efficacy Prediction. In *Bioinformatics Research and Applications: Fourth International Symposium*, pages 367–378, 2008.

A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi. Graph Kernels for Chemical Informatics. *Neural Networks*, 18:1093–1110, 2005.

J. Ramon and T. Gärtner. Expressivity versus Efficiency of Graph Kernels. In *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*, pages 65–74, 2003.

R. C. Read and R. E. Tarjan. Bounds on Backtrack Algorithms for Listing Cycles, Paths, and Spanning Trees. *Networks*, 5:237–252, 1975.

D. Rogers and M. Hahn. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50:742–754, 2010.

D. S. Rosenberg and P. L. Bartlett. The Rademacher Complexity of Co-Regularized Kernel Classes. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.

B. Schölkopf, A. J. Smola, and K.-R. Müller. Kernel Principal Component Analysis. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 583–588, 1997.

B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299–1319, 1998.

B. Schölkopf, R. Herbrich, A. J. Smola, and R. Williamson. A Generalized Representer Theorem. In *Proceedings of the Annual Conference on Computational Learning Theory*, pages 416–426, 2001.

B. Schölkopf and A. J. Smola. *Learning with Kernels – Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning*. Cambridge University Press, 2014.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

R. P. Sheridan and S. K. Kearsley. Why Do We Need So Many Chemical Similarity Search Methods? *Drug Discovery Today*, 7:903–911, 2002.

R. P. Sheridan, D. R. McMasters, J. H. Voigt, and M. J. Wilde. eCounterscreening: Using QSAR Predictions to Prioritize Testing for Off-Target Activities and Setting the Balance Between Benefit and Risk. *Journal of Chemical Information Modelling*, 55:231–238, 2015.

N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2010.

B. K. Shoichet. Virtual Screening of Chemical Libraries. *Nature*, 432:862–865, 2004.

V. Sindhwani and D. S. Rosenberg. An RKHS for Multi-View Learning and Manifold Co-Regularization. In *Proceedings of the Twentyfifth International Conference on Machine learning*, pages 976–983, 2008.

V. Sindhwani, P. Niyogi, and M. Belkin. A Co-Regularization Approach to Semi-Supervised Learning with Multiple Views. In *Proceedings of the ICML Workshop on Learning with Multiple Views*, 2005.

D. Singh, A. M. Ibrahim, T. Yohanna, and J. N. Singh. An Overview of the Applications of Multisets. *Novi Sad Journal of Mathematics*, 37:73–92, 2007.

A. J. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. *Statistics and Computing*, 14:199–222, 2004.

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

A. Speck-Planche and M. N. D. S. Cordeiro. Chemoinformatics in Drug Design. Artificial Neural Networks for Simultaneous Prediction of Anti-Enterococci Activities and Toxicological Profiles. In *Proceedings of the International Joint Conference on Computational Intelligence*, 2014.

K. Sridharan and S. M. Kakade. An Information Theoretic Framework for Multi-View Learning. In *Proceedings of the Twentyfirst Annual Conference on Learning Theory*, 2008.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

M. M. Stepniewska-Dziubinska, P. Zielenkiewicz, and P. Siedlecki. Development and Evaluation of a Deep Learning Model for Protein-Ligand Binding Affinity Prediction. *Bioinformatics*, 34:3666–3674, 2018.

N. Sugaya. Training Based on Ligand Efficiency Improves Prediction of Bioactivities of Ligands and Drug Target Proteins in a Machine Learning Approach. *Journal of Chemical Information and Modeling*, 53:2525–2537, 2013.

N. Sugaya. Ligand Efficiency-Based Support Vector Regression Models for Predicting Bioactivities of Ligands to Drug Target Proteins. *Journal of Chemical Information and Modeling*, 54:2751–2763, 2014.

S. Sun. Multi-View Laplacian Support Vector Machines. In *Proceedings of the International Conference on Advanced Data Mining and Applications*, pages 209–222, 2011.

S. Sun. A Survey of Multi-View Machine Learning. *Neural Computing & Applications*, 23:2031–2038, 2013.

I. V. Tetko and O. Engkvist. From Big Data to Artificial Intelligence: Chemoinformatics Meets New Challenges. *Journal of Chemoinformatics*, 12.

K. Tsuda, H. Shin, and B. Schölkopf. Fast Protein Classification with Multiple Networks. *Bioinformatics*, 21:59–65, 2005.

A. Turing. Computing Machinery and Intelligence. *Mind*, 49:433–460, 1950.

K. Ullrich and T. Gärtner. Kernel Corresponding Projections for Orphan Targets. Extended Abstract at the ECML Workshop on Multi-Target Prediction (KERMIT), 2014.

K. Ullrich, C. Stahr, and T. Gärtner. Counting-Based Output Prediction for Orphan Screening. In *Proceedings of the Conference Lernen, Wissen, Adaptation (LWA)*, pages 163–166, 2010.

K. Ullrich, M. Kamp, T. Gärtner, M. Vogt, and S. Wrobel. Ligand-Based Virtual Screening with Co-Regularised Support Vector Regression. In *Proceedings of the ICDM Workshop on Data Mining in Biomedical Informatics and Healthcare (DMBIH)*, pages 261–268, 2016a.

K. Ullrich, J. Mack, and P. Welke. Ligand Affinity Prediction with Multi-Pattern Kernels. In *Proceedings of the International Conference on Discovery Science*, pages 474–489, 2016b.

K. Ullrich, M. Kamp, T. Gärtner, M. Vogt, and S. Wrobel. Co-Regularised Support Vector Regression. In *Proceedings of the European Conference on Machine Learning*, pages 338–354, 2017.

V. N. Vapnik. An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks*, 10:988–999, 1999.

J.-P. Vert and L. Jacob. Machine Learning for in Silico Virtual Screening and Chemical Genomics: New Strategies. *Combinatorial Chemistry & High Throughput Screening*, 11:677–685, 2008.

J.-P. Vert and M. Kanehisa. Graph-Driven Features Extraction from Microarray Data using Diffusion Kernels and Kernel CCA. In *Proceedings of the Fifteenth International Conference on Neural Information Processing Systems*, pages 1449–1405, 2002.

J.-P. Vert, K. Tsuda, and B. Schölkopf. A Primer on Kernel Methods. *Kernel Methods in Computational Biology*, 47:35–70, 2004.

S. V. N. Vishwanathan, Z. Sun, N. Theera-Ampornpunt, and M. Varma. Multiple Kernel Learning and the SMO Algorithm. In *Proceedings of the Twentythird International Conference on Neural Information Processing Systems*, pages 2361–2369, 2010.

M. Vogt and J. Bajorath. Predicting the Performance of Fingerprint Similarity Searching. *Methods in Molecular Biology*, 672:159–173, 2010.

X. Wan. Co-Regression for Cross-Language Review Rating Prediction. In *Proceedings of the Fifty–First Annual Meeting of the Association for Computational Linguistics*, pages 526–531, 2013.

X. Wang, L. Ma, and X. Wang. Apply Semi-Supervised Support Vector Regression for Remote Sensing Water Quality Retrieving. In *Proceedings of IEEE International Geoscience & Remote Sensing Symposium*, pages 2757–2760, 2010a.

Z. Wang, Y. Li, C. Ai, and Y. Wang. In Silico Prediction of Estrogen Receptor Subtype Bindig Affinity and Selectivity Using Statistical Methods and Molecular Docking with 2-Arylnaphtalenes and 2-Arylquinoles. *International Journal of Molecular Sciences*, 11:3434–3458, 2010b.

A. M. Wassermann, H. Geppert, and J. Bajorath. Ligand Prediction for Orphan Targets Using Support Vector Machines and Various Target-Ligand Kernels Is Dominated by Nearest Neighbor Effects. *Journal of Chemical Information and Modeling*, 49:2155–2167, 2009a.

A. M. Wassermann, H. Geppert, and J. Bajorath. Searching for Target-Selective Compounds Using Different Combinations of Multiclass Support Vector Machine Ranking Methods, Kernel Functions, and Fingerprint Descriptors. *Journal of Chemical Information and Modeling*, 49:582–592, 2009b.

M. Welling. Kernel Canonical Correlation Analysis. Max Welling's Classnotes in Machine Learning, `www.ics.uci.edu/~welling/`.

D. Werner. *Funktionalanalysis*. Springer Verlag, 1995.

P. Willett. Similarity-Based Virtual Screening Using 2D Fingerprints. *Drug Discovery Today*, 11:1046–1053, 2006.

P. Willett, J. M. Barnard, and G. M. Downs. Chemical Similarity Searching. *Journal of Chemical Information and Computer Sciences*, 38:983–996, 1998.

D. J. Wood, J. de Vlieg, M. Wagener, and T. Ritschel. Pharmacophore Fingerprint-Based Approach to Binding Site Subpocket Similarity and Its Application to Bioisostere Replacement. *Journal of Chemical Information and Modeling*, 52:2031–2043, 2012.

C. Xu, D. Tao, and C. Xu. A Survey on Multi-View Learning. ArXiv, 2013.

S. Xu, X. An, X. Qiao, L. Zhu, and L. Li. Semi-Supervised Least-Squares Support Vector Regression Machines. *Journal of Information & Computational Science*, 8: 885–892, 2011.

J. Zhao, X. Xie, X. Xu, and S. Sun. Multi-View Learning Overview: Recent Progress and New Challenges. *Information Fusion*, 38:43–54, 2017.

H. Zhou and J. Skolnick. FINDSITE^X: A Structure-Based, Small Molecule Virtual Screening Approach with Application to All Identified Human GPCRs. *Molecular Pharmaceutics*, 9:1775–1784, 2012.

Z.-H. Zhou and M. Li. Semi-Supervised Regression with Co-Training. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 908–913, 2005.

X. Zhu. Semi-Supervised Learning Literature Survey. Technical report, University of Winsconsin, 2006.