Institut für Geodäsie und Geoinformation
Bereich Photogrammetrie

Tractable Learning for a Class of Global Discriminative Models for Context Sensitive Image Interpretation

Inaugural-Dissertation
zur
Erlangung des Grades
Doktor-Ingenieur
(Dr.-Ing.)
der
Hohen Landwirtschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität
zu Bonn

vorgelegt am 13. Oktober 2011 von
Filip Korč
aus Český Těšín, Tschechische Republik

Referent:	Prof. Dr.-Ing. Dr. h.c. mult. Wolfgang Förstner
Korreferent:	Prof. Dr. Lutz Plümer
Tag der mündlichen Prüfung:	9. Februar 2012
Erscheinungsjahr:	2012

Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn elektronisch publiziert.

Zusammenfassung

Wir beschreiben eine Klasse von bedingten Markov Zufallsfeldern für die kontextsensitive Bildinterpretation. Die beschriebene Klasse beinhaltet Modelle mit affinen Log-Potentialfunktionen und mit paarweisen Label Interaktionen, die label-paar-spezifisch, datenabhängig und asymmetrisch sein können. Instanzen der beschriebenen Klasse verknüpfen digitale Bilder und deren semantische Beschreibungen mittels einer globalen bedingten Wahrscheinlichkeitsverteilung. Die unbekannten Parametern der Verteilung werden aus Beispielen automatisch und gemeinsam gelernt. Unbekannte semantische Beschreibungen werden aus dem gelernten globalen statistischen Modell automatisch berechnet. Unser erster Beitrag ist, die asymmetrischen Label Interaktionen zu untersuchen, die in der Literatur selten behandelt werden. Unser zweiter Beitrag ist, es zu zeigen, dass ein Modell mit den Log-Potentialfunktionen in der Form von affinen Funktionen äquivalent zu einem Modell mit den Log-Potentialfunktionen in der Form von Modellen für die logistische Regression für Klassifikation ist. Nur wenige Ansätze für das gemeinsame Lernen der Parameter wenden das im Allgemeinen nicht berechenbare Maximum Likelihood Prinzip an. Die berechenbaren modernen Ansätze für das gemeinsame Lernen der Parameter können schnell zufriedenstellende Parameter Schätzungen liefern. Sie werden jedoch heuristisch aus Approximationen mit einem oszillierenden Verhalten berechnet. Unser dritter Beitrag ist daher, eine konsistente konvergierende Approximation zu identifizieren, die die Form eines berechenbaren stark konvexen Optimierungsproblems hat. Wir wenden die konvexe von Julian Besag vorgeschlagene Pseudolikelihood Approximation an und kombinieren sie mit den stark konvexen Prior Wahrscheinlichkeitsverteilungen. Wir zeigen, dass die Pseudolikelihood basierten Ansätze rechnerisch effizient sind, indem wir vorschlagen, effiziente Methoden der konvexen Optimierung anzuwenden. Unser vierter Beitrag ist, zu zeigen, dass die Pseudolikelihood basierten Lernansätze im Vergleich zum Stand der Forschung kompetitive Ergebnisse liefern. Unser fünfter Beitrag ist, einen Weg vorzuschlagen, die Leistungsfähigkeit der Spezialmodelle, unter anderem das bekannte Potts Modell, zu vergleichen. Die Spezialmodelle können gelernt und verglichen werden, indem das konvexe Optimierungsproblem mit linearen Gleichheitsbedingungen erweitert wird. Wir zeigen dies an Beispielen aus drei Anwendungsbereichen, nämlich der Interpretation von den Bildern von Straßenszenen, von den multi-spektralen Bildern von erkrankten Pflanzenblättern und von den volumetrischen Bildern aus der Magnetresonanztomographie von den menschlichen Knien.

Abstract

We propose a class of conditional Markov random fields for context sensitive image interpretation. The proposed class includes multi-class models with affine log-potential functions and with pairwise label interactions that are label-pair-specific, data-dependent and asymmetric. Instances of the proposed class relate observed images to unknown configurations of object class labels through global conditional probability distribution from the exponential family parametrized by unknown parameters that we jointly learn from examples. Unknown label configurations are jointly inferred from the learned global image model. The state-of-the-art models include pairwise label interactions that are label-pair-specific and data-dependent. Our first contribution is to investigate the in the proposed class included and in the literature rarely reported pairwise label interactions that are also asymmetric. State-of-the-art models include log-potential functions parametrized as affine functions or alternatively as popular multi-class logistic regression models for classification. Our second contribution is to show that a model with log-potentials of the former form is a simpler equivalent form of the latter. Parameter learning approaches commonly treat components of a global model independently. Isolated literature on joint parameter learning adopts the in general intractable maximum likelihood principle. Tractable state-of-the-art approaches to joint parameter learning yield satisfactory parameter estimates fast, however, obtained heuristically from approximations with oscillatory behavior. Our third contribution is to identify a consistent approximation in the form of a tractable strongly convex optimization problem. We adopt the convex pseudolikelihood approximation proposed by Julian Besag and combine it with the strongly convex parameter prior distributions. We provide the first partial derivative equations of the pseudolikelihood based learning objective needed to compute the solution with efficient algorithms of convex optimization. Our fourth contribution is to counterbalance reported statements that pseudolikelihood based approaches yield unsatisfactory results by providing state-of-the-art results. Our fifth contribution is to propose a way to compare the performance of models from the subclasses of models like the Potts model which can be learned by adding linear equality constraints to the described optimization problem. In experiments we compare the performance of the data-dependent asymmetric interaction model with the performance of the popular contrast sensitive Potts models. We present application examples of pixel level object class segmentation for interpreting images of street scenes, multi-spectral images of diseased plant leafs and volumetric human knee images from magnetic resonance.

Contents

Notation

Specific sets

$\mathbb{R}$	Real numbers.
$\mathbb{R}^D$	Real D-vectors.
$\mathbb{R}_+, \mathbb{R}_{++}$	Nonnegative, positive real numbers.

Vectors and matrices

x	A scalar.
$\boldsymbol{x}$	A vector.
$\boldsymbol{x}^\mathsf{T}$	Transpose of a vector x.
$\mathbf{0}$	Vector with all components zero.
$\mathbf{0}_D$	D-vector with all components zero.
$\mathbf{1}$	Vector with all components one.
$\mathbf{1}_D$	D-vector with all components one.
$\boldsymbol{X}$	A matrix.
$\boldsymbol{X}^\mathsf{T}$	Transpose of a matrix X.
$\boldsymbol{I}$	Identity matrix.
$\mathbf{Diag}(\boldsymbol{x})$	Diagonal matrix with diagonal entries formed by vector $\boldsymbol{x}$.

Functions and derivatives

∇f	Gradient of function f.

Specific functions

$\delta(x)$	Kronecker delta function.

Probability

$\underline{x}$	Random variable.
x	Sample of a random variable.
$\underline{\boldsymbol{x}}$	Random vector.
$\boldsymbol{x}$	Sample of components of a random vector.
$P\{\underline{x} = x\}$	Probability of the event $\{\underline{x} = x\}$.
p	Probability mass function or probability density function.
$\langle x \rangle_p$	Expected random variable value for a given distribution p.

Chapter 1

Introduction

> Ich stehe am Fenster und sehe ein Haus, Bäume, Himmel.
>
> Und könnte nun, aus theoretischen Gründen, abzuzählen versuchen und sagen: da sind . . . 327 Helligkeiten (und Farbtöne).
>
> (Habe ich “327”? Nein; Himmel, Haus, Bäume; und das Haben der “327” als solcher kann keiner realisieren.)[1]
>
> *Max Wertheimer*[2] *1923*, [Wertheimer, 1923][3]

Our highly developed human brains have been trained lifelong to recognize objects we encounter in our environment. We as humans are able to use our brains to relate images sensed through our eyes to the meaning of words we use to verbally describe the environment. Thus we are able to turn our visual sensation into perception by extending iconic representations with semantic representations. Indeed human interpretation of images is seamless and instant. We can recognize objects, we observe for the first time, as instances of to us familiar classes of objects and from very few examples are capable of learning new object classes, instances of which we have not encountered before. Still, there are tasks involving interpretation of images, performance of which by humans is too time consuming, too costly or simply undesired.

[1]

> I stand at the window and see a house, trees, sky.
>
> Theoretically I might say there were 327 brightnesses and nuances of colour. Do I *have* “327”? No. I have sky, house, and trees. It is impossible to achieve “327” as such.
>
> *Max Wertheimer 1923*, [Ellis, 1950].

[2] Max Wertheimer (1880–1943), one of the founders of Gestalt psychology [Ellis, 1950].

[3] The translation can be found in [Ellis, 1950].

The overall goal of this work is to develop a function that can be trained from examples to interpret images automatically. We use the term function as an abstraction of a possibly complex computer program, an abstract computing machine or simply as a function that can be implemented in a computer. In the following we first describe the function in terms of its input and in terms of its output. Afterwards we specify the function to be developed regarding its desired properties.

Let us first describe the trainable function for automatic image interpretation in terms of its input. Let the input images be any representations of parts of the physical world. More specifically let an image be a collection of measurements of multiple physical phenomena in a part of space. For instance regular arrangements of measurements in the space forming up to five-dimensional $X \times Y \times Z \times T \times D$ arrays of numerical values are common in imaging. The dimensions X, Y, Z, T and D of these arrays are the three numbers X, Y, Z of coordinates in the spatial dimensions, the number T of the coordinate in the temporal dimension and the number D of physical phenomena being measured. These arrays include, for instance, $X \times Y$ grayscale image arrays, $X \times Y \times 3$ color image arrays represented in the three-dimensional RGB color space, $X \times Y \times T \times 3$ time series of color image arrays (video) represented in the three-dimensional RGB color space, $X \times Y \times D$ multi-spectral image arrays, volumetric $X \times Y \times Z$ magnetic resonance image arrays and $X \times Y \times Z \times 3$ vector volume data arrays capturing flow vector fields in space. It is our aim however to develop a concept that could be directly applied to irregular arrangements of measurements in space. Irregular arrangements of measurements can either be seen as forming partially filled arrays in case the spatial coordinates of the measurements are known or they can form graphs in case only proximity structures of the irregular arrangements of measurements in space are known. These irregular arrangements of measurements include, for instance, laser scan point clouds, where each point is associated with a measurement describing the spectrum of the reflected laser beam. These irregular arrangements of measurements in space further include, for instance, excavated objects from multiple archaeological sites. In summary it is our goal to develop a function that accepts irregular arrangements of measurements in space as its input.

Let us now describe the trainable function for automatic image interpretation in terms of its output. Let the outcomes of interpretation of images be semantic descriptions of components of the images. In our work we restrict our attention to individual image components and do not consider semantic description of their relations. An image component can be the image itself. On the other hand image components can be the individual measurements in the image. It is our aim however to develop a function that maps more generally input images on configurations of semantic object class labels, where individual labels in the configuration are assigned to subsets of measurements from arbitrary partitions of measurements in the input images. In summary it is our goal to develop a function that returns irregular configurations of semantic labels as its output.

We digress momentarily from describing the trainable function for automatic image interpretation to make the following remark. Image interpretation phrased as a mapping from irregular arrangements of measurements in space on irregular configurations of semantic labels can be seen as a generalization of classification, where single or multiple measurements are mapped on a single semantic class label. Such generalized mapping is sometimes referred to as predicting structured objects and generally also phrased as the structured prediction or the structured output learning [Bakir et al., 2007]. Here the aim is to predict an object from an arbitrary output space based on an object from an arbitrary input space. Prediction shall be thought of as interpretation in our context. In particular the input domain in our case is the set of all conceivable arrangements of measurements in space. The output domain in our case is the set of all corresponding conceivable configurations of semantic labels. Both the input arrangements of measurements in space and the output configurations of semantic labels are structured in the sense of in general unknown underlying statistical interdependence of the individual measurement variables and of the individual semantic variables. The statistical interdependencies can be represented in many ways, for instance as graphs or as sets of rules. We assume that they can be expressed in terms of graphs, specifically using the concept of the graphical models [Lauritzen, 1996].

We now characterize the trainable function for automatic image interpretation, that we want to develop, in terms of its desired conceptual properties. It is our aim to develop a function that is statistical in nature and relates internally observed images to semantic descriptions in terms of probabilities. In particular we consider the image interpretation process, mapping images on semantic descriptions, rather than the image generation process, mapping semantic descriptions on images. Thus we adopt the discriminative statistical approach. Such an approach could be based on data, it could be based on a model or it could be based on the combination of both. The data based approach could be based on a large set of examples of images and semantic descriptions. In this approach the interpretation of query images not included in the data set could be based on the exploitation of the semantic descriptions of image examples included in the data set that are similar to the query images. However images and their semantic descriptions are in general objects of very high dimension and representing the spaces, in which these objects exist, with representative examples would require extremely high number of these examples. Since obtaining the representative examples in general involves some kind of human annotation, we argue that it would be difficult to obtain the data set in the first place. Even if we had such data set, we argue that such an approach shall be demanding in terms of memory and computation. While we acknowledge purely data based approaches in the context of problems involving less or no human input, in this thesis we adopt a model based approach. In this approach we adopt models that relate images to semantic descriptions in a way that does not involve storing a set of examples. In summary it is our goal to develop a model based function that is conceptually statistical and discriminative.

We characterize the trainable function for automatic image interpretation in terms of its further desired conceptual properties. We argue that assigning subsets of image measurements with semantic object class labels based only on the measurements from the subsets themselves is often ambiguous. Statistical approaches offers a theoretically well founded way to model such ambiguity. It is widely accepted that a context sensitive model represents a vital way to suppress ambiguity. We shall include two types of context in the model that the function is based on. First it is our aim to develop a function that is capable of viewing the measurements from a particular subset of measurements in the context of the measurements from the rest of the image in order to assign this subset of measurements with a semantic label from a set of multiple semantic labels. We refer to this concept as to the measurement context sensitivity or simply as to the data context sensitivity. Second it is our aim to develop a function that is capable of viewing decisions regarding semantic labels in an image also in the context of other decisions elsewhere in the image. We refer to this concept as to the semantic context sensitivity. It is our aim to develop a function that can model the semantic context in the data-dependent manner and is capable of viewing decisions both in the context of other decisions and in the context of other measurements in the rest of the image. We refer to this joint concept as to the semantic context sensitivity in the data-dependent manner. In particular it is our aim to model such data-dependent semantic context in an asymmetric manner. We argue that data-dependent and asymmetric semantic context is a vital component in suppressing ambiguity. The model shall then conceptually combine these ambiguous context sensitive decision votes in a global semantic description vote. We argue that this is a viable element for ambiguity suppression. In summary it is our goal to develop a function that possesses the general conceptual properties of being data context sensitive and semantic context sensitive in a data-dependent manner. Specifically it is our goal to develop a function that further possesses the conceptual property of being sensitive to the semantic context in an asymmetric manner.

Let us illustrate the role of context sensitivity on an example. We consider an image of an urban street scene taken with a camera hold in an upright position. Let us consider the case where the trainable function for automatic image interpretation attempts to assign a blue colored pixel somewhere in the image and a green colored pixel elsewhere in the image with semantic labels “car”, “sky” or “tree”. Let us assume that the blue pixel is in a local image region of other blue colored pixels and that similarly the green pixel is in a local image region with other green colored pixels. Based on our experience with street scene images and based on the local data context it may seam sensible to us humans to assign the blue pixel with the “sky” label and the green pixel with the “tree” label. In addition we may consider each of the two individual semantic decisions in the context of the other decision. Based again on our experience it may seem sensible to agree that such semantic labels do co-occur in semantic descriptions of street scene images and that the semantic context supports this particular joint decision. Let us now assume that the trainable function for automatic image interpretation models the semantic context in a data dependent manner. This can be thought of as if we were looking at the image while considering the joint decision. And let us consider the case where the function only uses data from the image in the form of the image location. In case the blue pixel is in the image above the green pixel, we may strongly prefer the blue “sky” above the green “tree” labeling as opposed to the green “tree” above the blue “sky” labeling. In this case our preference is strongly asymmetric. Let us consider another case where the blue colored pixel is at the bottom of the image and where the green pixel is very much above. Based on our experience and based on the data dependent semantic context it may be sensible to us humans to consider the option that the blue bottom pixel is a part of a blue car parked under a tree. In this case the data context model component prefers the blue “sky” and the green “tree” labeling of the two pixels, whereas the data-dependent semantic context model component votes for the competing “car” below “tree” labeling. The global discriminative statistical model then combines these local ambiguous context sensitive decision votes in a more global decision vote.

Let us further characterize the trainable function for automatic image interpretation in terms of yet another desired conceptual property. It is widely accepted that due to in general high complexity designing the functions for image interpretation manually is infeasible. It turns out that it is rather feasible to design the functions that learn to interpret images from examples automatically. The global discriminative model, that the function is based on, shall thus include a component that is variable and can be adapted in the process of the training. In summary it is our goal to develop a function that is conceptually trainable.

Let us now characterize the trainable function for automatic image interpretation in terms of its computational properties. We wish to develop a function that eventually in practice can be turned into a computer program that receives as its input test images together with the training examples and returns as its output translations of the test images into semantic descriptions. It is our major intention to formulate a function that learns from the training examples and infers semantic descriptions from the test images reliably and efficiently. With reliably we mean being formulated as a problem from the class of the convex optimization problems, where an iterative descent algorithm is guaranteed in the limit to converge to the global optimum. With efficiently we mean that the number of iterations that the iterative descent algorithm needs to solve the problem up to a precision, expressed as a function of the problem size, is small. While the empirically observed number of iterations of efficient iterative descent algorithms are typically independent of the problem size, let us note that for general nonlinear convex optimization problems it is currently unknown whether it can be guaranteed that the number of iterations needed to solve these problems up to a precision, expressed as a function of the problem size, is upper bounded by a polynomial [Boyd and Vandenberghe, 2004]. Hence we yet cannot say that mechanisms, formulated in this manner, are guaranteed to be efficient. In this general case we can only rely on empirical observations that however are good-natured. Still there are classes of convex functions for which convergence analysis is available and guarantees an iterative algorithm to converge. These classes include the strongly convex functions and the so called self-concordant functions [Boyd and Vandenberghe, 2004]. The convergence analysis for the strongly convex functions yields an upper bound on the number of iterations that depends on in general unknown constants. This bound is conceptual and establishes that an iterative algorithm will converge. The convergence analysis for the self-concordant functions yields an upper bound that depends on known constants and can be evaluated before starting the iterative algorithm. With efficient mechanisms we mean then those that are based on the convex optimization problems for which an iterative descent algorithm is guaranteed to converge. For simplicity we will refer to efficient mechanisms as those based on the strongly convex optimization problems.

Until now we have characterized efficiency only in terms of the number of iterations of the algorithm. The overall complexity however depends still on the complexity of a single iteration. Hence we also require that the strongly convex optimization problem is tractable. With tractable convex optimization problem we mean that the complexity of an evaluation of the objective function, its first derivative and possibly its second derivative, that are needed to perform a descent step in a single iteration, expressed as a function of the problem size, is upper bounded by a low degree polynomial. In summary it is our goal to develop a function that is based on learning and inference mechanisms that are reliable and efficient in a sense of being formulated as the tractable strongly convex optimization problems. In short we simply refer to the tractable learning and inference mechanisms.

In summary we have described our overall goal to develop a model based function for automatic image interpretation that possesses the general conceptual properties of being statistical, discriminative, data context sensitive, semantic context sensitive in the data-dependent manner and trainable. Part of our overall goal is to develop a function for automatic image interpretation based on learning and inference mechanisms that possesses the general computational properties of being tractable.

The rest of current chapter is structured as follows. In Section 1.1 we list three examples of the application areas that may benefit from the development of the trainable functions for automatic image interpretation with the above described properties. In Section 1.2 we motivate our approach with respect to existing literature on the global statistical models and on tractable approximate approaches to the training and the inference with the global statistical models. Eventually in Section 1.3 we give an outline of the thesis.

1.1 Relevance to Applications

Let us list three application areas that may benefit from the development of the trainable functions for automatic image interpretation with the previously described properties. We include the area of 3D city modeling, the area of precision agriculture and the area of medical imaging.

Urban planning applications ranging from 3D city modeling [Cornelis et al., 2008, Kluckner and Bischof, 2010] over large scale updating of geographic information systems to vision based outdoor navigation in urban environments want to exploit large amounts of available image data. Such applications may benefit from automation of man-made scene interpretation and motivate variety of approaches [Korč and Förstner, 2008c, Ripperda and Brenner, 2009] to the challenging problem of interpreting images of the complex scenes such as the one in figure 1.1.

Agricultural applications aiming at treatment of leaf diseases in the field rely on early and large scale disease detection [Mahlein, 2010]. Early and localized treatment may be boosted by means of automatic methods for in-

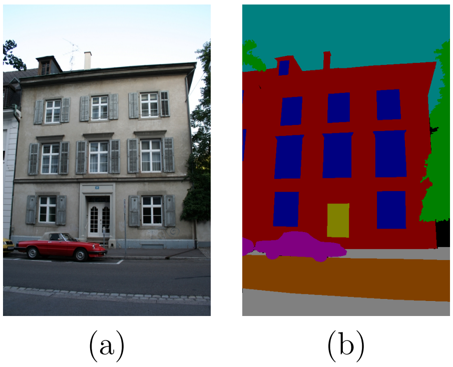


Figure 1.1: (a) Image of a street scene. (b) Manual pixel level object class segmentation showing *building, car, door, pavement, road, sky, vegetation, window* and *background* labels. Example from the 8–Class eTRIMS Dataset [Korč and Förstner, 2009].

terpreting high-resolution multi-spectral images of diseased plants [Bauer, Korč, and Förstner, 2011] such as the ones in figure 1.2.

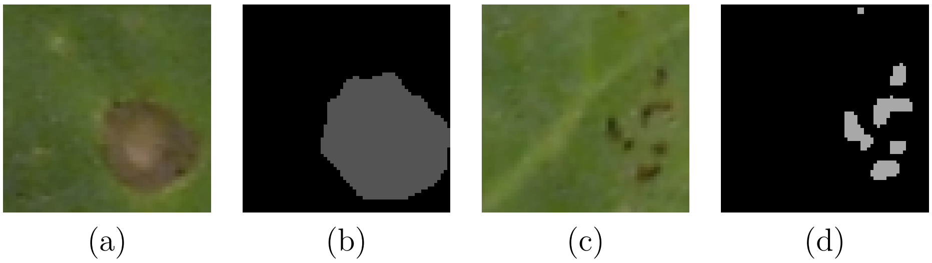


Figure 1.2: (a)(c) Images of diseased sugar beet leafs. (b)(d) Manual pixel level object class segmentation showing *Cercospora beticola, Uromyces betae* and *healthy leaf* labels. Examples from [Bauer, Korč, and Förstner, 2011].

Medical applications as three-dimensional semantic segmentation of magnetic resonance images of the human knee involve interpretation of large amount of volumetric images [Heimann et al., 2010]. A prevalent approach to the task in clinical practice is a manual or semi-automated two-dimensional slice by slice segmentation. We show an example of a two-dimensional slice of an magnetic resonance image in figure 1.3. Such approach may possibly result in several hours of trained radiologist's manual slices annotation. Medical applications may thus also benefit from fast automatic methods for three-dimensional image interpretation [Korč, Schneider, and Förstner, 2010].

What these diverse application areas have in common is the need to interpret large amounts of image data at a scale where manual interpretation

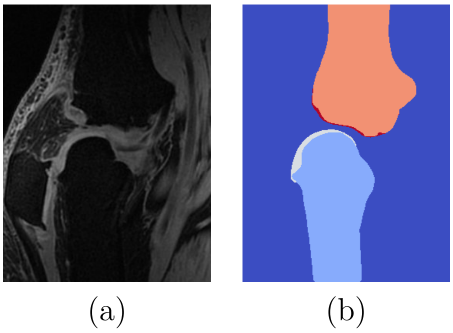


Figure 1.3: (a) two-dimensional slice of a three-dimensional (volumetric) magnetic resonance image. (b) Manual voxel level object class segmentation of the magnetic resonance image in (a), where we show the corresponding two-dimensional slice, showing *femur, femoral cartilage, tibia, tibial cartilage* and *background* labels. Example from [Korč, Schneider, and Förstner, 2010], data from [Heimann et al., 2010].

is too time consuming and thus too costly. The trainable functions for automatic image interpretation with the previously described properties may then represent a viable alternative.

1.2 Related Work

It is our overall goal in this thesis to develop a model based function for automatic image interpretation that possesses certain desired conceptual and computational properties. The development of the function involves the development of three components. The first component represents internal models that relate images and semantic descriptions. The second component represents mechanisms for learning parts of the internal models from the training examples. The third component represents further mechanisms for inference of semantic descriptions from newly observed images based on the learned internal models. In this section we review with respect to the desired properties literature on models, learning and inference mechanisms.

In the structured prediction literature we first review families of models with respect to the general conceptual properties of being statistical, discriminative, data context sensitive, semantic context sensitive in the data-dependent manner and trainable. Afterwards we specifically review potentially admissible families of models with respect to their ability to capture semantic context. In particular we consider the specific property of being sensitive to the semantic context in a way that is data-dependent and asymmetric. Afterwards for the potentially admissible families of models that possess the desired conceptual properties we review the literature on the learning mechanisms and the inference mechanisms with respect to the computational property of being tractable.

1.2.1 Structured Prediction Models

Let us first identify the families of the structured prediction models that possess desired conceptual properties of the model that the trainable function for automatic image interpretation shall be based on. There are four sources of the structured prediction models, namely grammars [Zhu and Mumford, 2006], more recently conditional random fields [Lafferty et al., 2001], recently structured support vector machines [Tsochantaridis et al., 2005, Bakir et al., 2007] and specifically for image segmentation variational models [Cremers et al., 2011]. We now describe these four types of models with respect to our goals.

In the context of image segmentation problems conditional random field models often have direct variational model counterparts. For instance the Ising model [Ising, 1925] and the Potts model [Potts, 1952] are well studied discrete models that have direct continuous analogues referred to as the minimal partition problems [Pock et al., 2009, Cremers et al., 2011]. Analogies, strengths and limitations of the spatially discrete formulation and of the spatially continuous formulation are experimentally studied in [Klodt et al., 2008]. Main focus of variational approaches so far has been on non semantic image segmentation problems though. Variational models are discriminative, however lack the statistical formulation of their discrete counterparts. They are context sensitive, however they only have been employed to model semantic context locally and mainly to the limited extent of imposing smooth solutions. Their ability of being trainable is limited to training the component that models the context in data.

Structured support vector machines [Tsochantaridis et al., 2004, Taskar et al., 2004, Blaschko and Lampert, 2008] are discriminative, however, they are not statistical in nature. They are context sensitive and trainable similarly to the conditional random fields. Learning of the structured support vector machines is closely related to the learning of the conditional random fields for a specific choice of the loss function [Bakir et al., 2007, Hazan and Urtasun, 2010a,b, Shi et al., 2010]. Discriminative training for the structured prediction with references to the global statistical models is reported in [Franc and Savchynskyy, 2008]. However as opposed to the conditional random field learning, structured support vector learning is not consistent in the infinite data limit.

We now proceed with the stochastic grammars. Let us first note that conditional random fields are conditional model instances of the more general Markov random fields. Stochastic grammars and Markov random field models are both the structured prediction models that are statistical, context sensitive and trainable. Grammars have been with success applied in the context of language interpretation. Markov random field on the other hand have become prominent in the context of image interpretation and that being largely due to their suitability for representing image textures. However the question of when to represent semantic image descriptions in terms of grammar production rules and when in terms of Markov random field spatial context is an open research question [Zhu and Mumford, 2006]. We leave image interpretation models based on stochastic grammars for future exploration.

Conditional random fields have been proposed in [Lafferty et al., 2001] in the context of segmentation and labeling of one-dimensional text sequences as a family of models that possess the conceptual properties of being statistical, discriminative, data context sensitive, semantic context sensitive in the data-dependent manner and trainable.

Since conditional random fields on an abstract level possess the desired general conceptual properties of the global discriminative statistical models, that the trainable function for automatic image interpretation is to be based on, we in this thesis adopt this family of the structured prediction models.

1.2.2 Conditional Random Fields

In this section we want to find out to what extent does the family of conditional random fields satisfy the desired specific conceptual properties. We first review the conditional random field literature in general and afterwards we review literature on the conditional random fields specifically with respect to their ability to model semantic context. In particular we consider the specific conceptual property of being sensitive to the semantic context in an asymmetric manner.

Early a posteriori conditional Markov random field models are mentioned in [Modestino and Zhang, 1992, Gimel'farb and Zalesny, 1993, Gimel'farb, 1996, Grenander, 1996]. The name conditional random field has been proposed in [Lafferty et al., 2001] in the context of segmentation and labeling of one-dimensional text sequences. The wide spread of conditional Markov random fields is largely attributed to the latter work in [Lafferty et al., 2001]. In [Lafferty et al., 2004] the authors extend the model to a kernel based formulation and apply the model to the problem of protein structure prediction. General introduction to conditional random fields in the context of image interpretation can be found in [Sutton and McCallum, 2007a, Nowozin and Lampert, 2011, Sutton and McCallum, 2011]. Conditional random fields belong to the large family of undirected graphical models. Interested reader may find it helpful to refer for background to a standard text on graphical models [Lauritzen, 1996], to an introductory text on graphical models [Bishop, 2006], to an introductory and reference text on graphical models [Koller and Friedman, 2009], to a mathematically oriented text on graphical models [Winkler, 2006], to a text on graphical models from the exponential family [Wainwright and Jordan, 2008] or to texts on graphical models with focus on image interpretation [Gimel'farb, 1999, Schlesinger and Hlaváč, 2002, Winkler, 2006, Li, 2009]. In the following we review literature on the conditional random fields and group conditional random field models into seven categories with respect to their ability to capture semantic context.

Semantic context sensitive model component can be seen as a sum of interactions of pairs of semantic variables involved in the model. The simplest models, that we consider in this thesis, actually ignore interacting variables and the semantic labels that they take. Such models can be seen as a special case of conditional random fields, namely as conditional random fields that have been deprived of their capability to capture semantic context. Context insensitive models are out of scope of this thesis and hence we do not review here the vast literature describing this topic. We refer to models without any label interaction as to *local classifiers*. This is our first category of conditional random field models.

Perhaps the most widely adopted form of modeling context is to favor interacting variables to take the same semantic label. Here we will also not attempt to provide any systematic review of the large body of literature describing models with this kind of interaction. Such interactions are sometimes also referred to as attractive interactions. As a result such modeling favors semantic label configurations that can be described as smooth. Such semantic context sensitive model is based on the widely accepted assumption that locally neighboring semantic labels tend to posses similar semantic labels. In [Berg et al., 2007] the authors introduce a model that is only loosely related to a conditional random field. The authors view image interpretation as a sequence of processing steps and model the semantic context implicitly as one of the processing steps that involves smoothing an intermediate semantic label configuration. Hence the semantic context sensitive model component is not an explicit part of the model. More commonly model components modeling semantic context in this manner take the form of the well studied Potts model [Potts, 1952]. Semantic context sensitivity is then an explicit part of an overall model. Here no attempt is made to review the extensive literature on Potts model based approaches. In [Roscher et al., 2010] the authors combine the Potts model with a sparse kernel logistic regression classifier for the interpretation of land cover images. In [Korč et al., 2010] the authors adopt the Potts model as one of the components in a global statistical model that favors smooth voxel level semantic segmentation result of volumetric magnetic resonance images. We refer to the models with the data-independent label-pair-nonspecific attractive interactions as to the *Potts models*. This is our second category of conditional random field models.

One way to generalize the Potts models is to treat label interactions in a label-pair-specific manner that is to treat label interactions depending on the particular assigned semantic label pair. The model in [Rabinovich et al., 2007] that was employed for incorporating a semantic object class context treats pairs of same semantic labels in a label-dependent manner and it treats pairs of different semantic labels in a label-independent way. In [Plath et al., 2009, Nowozin and Lampert, 2010, Bauer et al., 2011] the authors employ models that treat also pairs of different semantic labels in a label-dependent and symmetric manner. In [Plath et al., 2009] the authors employ this concept in combining local and global features. In [Nowozin and Lampert, 2010] the authors make use of the this model component to impose global connectedness in the conditional random field formulation. In [Bauer, Korč, and Förstner, 2011] the authors employed this model and from the training examples learn the specific interactions of neighboring semantic labels. We refer to the models with the data-independent label-pair-specific symmetric interactions as to the *generalized Potts models* [Boykov et al., 1998]. This is our third category of conditional random field models.

Immediate generalization of previous variable interactions are interactions that are asymmetric. This kind of label interaction has been employed in a hidden variable based multi-scale model formulation for semantic image segmentation [He et al., 2004]. In [Quattoni et al., 2004, Wang et al., 2006, Quattoni et al., 2007] the authors employ the concept in hidden parts based model for object and gesture recognition. Modeling temporal contextual dependencies in video sequences is described in [Sminchisescu et al., 2005]. In [Nowozin et al., 2010] the authors employ the concept to model asymmetric parent-child relation in hierarchical and multi-scale conditional random field model for pixel level semantic segmentation. In [Barth et al., 2010] the authors employ an asymmetric interaction model to encode prior knowledge that certain combinations of semantic object class labels, for instance a car and the ground, appear in scenes mainly in particular configurations, that is a car above the ground in this particular case. We refer to the models with the data-independent label-pair-specific asymmetric interactions as to the *asymmetric generalized Potts models*. This is our fourth category of conditional random field models.

Widely adopted models that are semantic context sensitive in the data-dependent manner include data-dependent attractive interactions [Boykov and Jolly, 2001, Blake et al., 2004]. After the concept of a conditional random field has been introduced in the context of labeling one-dimensional text sequences, in [Kumar and Hebert, 2003, 2004a] the authors apply the concept of conditional random field with contrast sensitive smoothing to graphs with loops in the context of image interpretation. In [Kumar et al., 2005, Korč and Förstner, 2008a] the authors study this model formulation from the learning perspective. In [He et al., 2006] the authors control the degree of smoothing with a superpixel boundary classifier. Discussion of the differences between generative and discriminative formulation can be found in [Kumar and Hebert, 2006, Korč and Förstner, 2008c] together with comparison of performance of these models. Conditional random field for action classification in videos is reported in [Wang and Suter, 2007]. The concept of conditional random fields can be employed for incorporating a semantic object context [Shotton et al., 2009]. In [Micusik and Kosecka, 2009] the authors employ difference of color at two neighboring superpixels to compensate smoothing of the resulting label field. In [Fulkerson et al., 2009] the authors control the degree of smoothing of a labeling of superpixels by considering their color and also the length of their shared boundary. Contrast sensitive smoothing in the context of a hierarchical formulation combining semantic segmentation of pixels, segments and groups of segments can be found in [Ladicky et al., 2009]. In [Lucchi et al., 2011] authors adopt a contrast sensitive smoothing model to show in their experiments that a model that is insensitive to the global semantic context and that is primarily based on global image features, yields comparable results as a model that is also sensitive to the global context. We refer to the models with the data-dependent label-pair-nonspecific attractive interactions as to the *contrast sensitive Potts models*. This is our fifth category of conditional random field models.

Semantic context can further be modeled in the form of data-dependent label-pair-specific and symmetric label interaction. A multi-class formulation of a conditional random field model over graph with non-lattice topology applied to part-based object detection is described in [Kumar and Hebert, 2004b]. Further, formulation with hierarchical interactions can be found in [Kumar and Hebert, 2005]. Conditional random field for three-dimensional point cloud segmentation is described in [Anguelov et al., 2005]. Here the authors use a slightly simpler form and only model the pairs of the same semantic labels in this way. The other label interactions are data-independent and equal to a constant. We refer to the models with the data-dependent label-pair-specific symmetric interactions as to the *data-dependent symmetric interaction models*. This is our sixth category of conditional random field models.

In the following we in more detail discuss the data-dependent symmetric interaction models that partially possess desired specific properties regarding context sensitivity. They are based on a model proposed in [Kumar and Hebert, 2004a]. There are two components in the model, namely the unary term and the pairwise term that are responsible for modeling the data context and the semantic context. The unary term models the assignment of a single semantic label to a particular location or site in the image and that in the context of the data from the rest of the image. The pairwise term models the assignment of two semantic labels at two distinct locations in the image and that again in the context of the data in the rest of the image. The conditional random field model then combines both the unary and the pairwise terms in a global conditional probability distribution. There are two forms of this model, namely a 2-class formulation and a multi-class formulation that we discuss next.

The 2-class formulation is proposed in [Kumar and Hebert, 2004a]. The authors of this work use class label variables that can only take values in the particular set $\{-1, +1\}$ of class labels, to arrive at a particular 2-class CRF model. In this model the unary term distinguishes between label -1 and label $+1$. The pairwise term distinguishes between a "smooth" label pair, i.e., label pair $(-1, -1)$ or label pair $(+1, +1)$, and a "non-smooth" label pair, i.e., label pair $(+1, -1)$ or label pair $(-1, +1)$. The 2-class formulation is also adopted in [Kumar et al., 2005],[Kumar and Hebert, 2006],[Korč and Förstner, 2008c] and [Korč and Förstner, 2008a]. In [Korč and Förstner, 2008c] the authors extend the model with application specific image features and evaluate the potential of the model formulation in the context of urban scene image interpretation. In [Korč and Förstner, 2008a] the authors adopt the 2-class formulation and counterbalance statements regarding the potential of certain approximate schemes for parameter learning, made in [Kumar et al., 2005].

The multi-class formulation proposed in [Kumar and Hebert, 2004b] involves class label variables that now take values k in the set $\{1, \ldots, K\}$ of class labels. In this model the unary term distinguishes labels $\{1, \ldots, K\}$. The pairwise term now models different label pairs (k, k'). However, the pairwise term does not distinguish a label pair (k, k') from a label pair (k', k), when label k is different from k'. For $K = 2$ this means that the pairwise term distinguishes "smooth" label pair $(1, 1)$ from "smooth" label pair $(2, 2)$, which can be regarded a generalization of the 2-class formulation both in the number of modeled class labels and in the ability of the pairwise term to model semantic context. However, the pairwise term does not distinguish "non-smooth" label pair $(1, 2)$ from "non-smooth" label pair $(2, 1)$. The authors note that this fact is implied by the associated edge in the graph being undirected. The multi-class formulation is also employed in [Kumar and Hebert, 2005] as one of two layers in a hierarchical image interpretation model. As we have previously described, in our view it is vital to be able to model data-dependent semantic context in an asymmetric manner. The symmetric label interaction in the described models is then an undesired restriction in the ability of the model to capture semantic context.

It is our goal to generalize both the 2-class formulation proposed in [Kumar and Hebert, 2004a] and the multi-class formulation proposed in [Kumar and Hebert, 2004b] by extending their formulation with asymmetric label interaction. Specifically it is our goal to formulate a multi-class model with semantic label variables that take values in the set $\{0, \ldots, K-1\}$ and where specifically for the case $K = 2$ we wish that each of our pairwise terms is capable of distinguishing all four label pairs $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$.

Let us now consider more closely the parametrization of the previously described conditional random field models. In [Kumar et al., 2005] the authors propose to model the unary and the pairwise terms using the logistic regression model for classification. The authors replace, however the logistic pairwise term with a simpler affine function to arrive at a model that has already been proposed in [Kumar and Hebert, 2004a]. In [Kumar et al., 2005] the authors view the affine pairwise term as a simplified form of the logistic pairwise term. In [Kumar and Hebert, 2006] the authors argue the other way round that it would be possible to generalize the form of the affine pairwise term to the form of the logistic pairwise term. In [Kumar and Hebert, 2004b] the authors adopt a multi-class extension of the logistic regression model for classification used in the 2-class formulation, the so called softmax function, also known as normalized exponential, to model the unary term in the multi-class formulation. Similarly the authors adopt a multi-class extension of the affine pairwise term used in the 2-class formulation. In none of the above publications do the authors provide clear motivation for the replacement of the logistic pairwise term with an affine pairwise term. In [Roscher et al., 2010] the authors employ a generalization of the multi-class logistic regression model for classification, a sparse kernel logistic regression classifier, to design the unary term, which they combine with a Potts model based pairwise term. It is our goal to clarify the relation between the formulations involving the logistic regression, the softmax and the affine functions.

We are particularly interested in semantic context being modeled in a data-dependent, label-pair-specific and asymmetric manner. This type of interaction is scarce in the experiments reported in the literature. In [Wojek and Schiele, 2008] the authors employ the concept to model inter-layer label interaction in a two layer conditional random field that combines object and scene modeling. In [Schnitzspan et al., 2008, 2009] the authors employ this label interaction in a multi-layer hierarchical formulation. In [Schnitzspan et al., 2010] the authors employ the data-dependent label-pair-specific and asymmetric label interaction to model flexible constellations of parts for part based object recognition. We refer to the models with the data-dependent label-pair-specific and asymmetric interactions as to the *data-dependent asymmetric interaction models*. This is our seventh and last category of conditional random field models.

In summary we have grouped conditional random fields into seven categories of models according to their ability to capture semantic context. We have identified the group of data-dependent asymmetric interaction models as the category of models that also possess the desired specific properties regarding context sensitivity. In conclusion so far we have identified conditional random fields as being the family of models that posses both the desired general and the desired specific conceptual properties.

It is our goal to unify the above groups of models in a way that should illuminate their relation and facilitate their comparison. Specifically it is our goal to formulate a class of models that should include groups of model formulations reviewed above as subclasses. Further it is our goal to clarify the relation between alternative parameterizations of the model in [Kumar and Hebert, 2004b] that are present in the literature.

1.2.3 Learning Global Model Parameters

In this section we review literature on learning parameters of conditional random fields with respect to our goal to develop a function that employs learning mechanisms that are computationally tractable in a sense of being based on tractable convex optimization problems.

Conditional random fields are global statistical models that in general pose a problem of learning a complex probability distribution. A widely adopted approach is to first formulate a model of specific functional form governed by some number of unknown parameters and subsequently to learn these parameters from a training data set according to some principle. In this thesis we restrict ourselves to the maximum likelihood principle, placing in this work fully Bayesian approach out of scope. In this thesis we further restrict ourselves to learning in a supervised manner. A semi-supervised learning approach to learning in conditional random field can be found in [Lee et al., 2007]. Learning parameters of a global statistical model in the form of the conditional random field poses an optimization problem that is convex however where the convex optimization problem involves an objective function that is in general intractable to evaluate [Winkler, 2006, Wainwright and Jordan, 2008]. Learning unknown parameters of a global probabilistic model is a task that can in principle be solved by a Monte Carlo simulation [Winkler, 2006] or, alternatively, by methods of variational inference [Wainwright and Jordan, 2008]. Even though both approaches are inherently exact, in practice both approaches in general inevitably lead to approximations. In case of Monte Carlo simulation, approximation results from limited computational resources and practical time constraints. In case of variational inference, approximation stems from reformulation of an implicitly posed exact variational principle as an explicitly posed convex approximation that can efficiently be solved. Monte Carlo simulation based learning methods, though exact in infinite time limit, generally lack rapid convergence rate guarantees. Monte Carlo simulations have been observed to be slow due to a long "burn-in" phase and to exhibit large estimate variance in practice [Hinton, 2002]. Variational inference can be used to arrive at tractable approximate learning and that in the form of convex optimization problems. Problems in this form can be solved, very reliably and efficiently, drawing upon the benefits of readily available methods for convex optimization.

There are two widely adopted approximation approaches that can be understood in terms of the variational inference, namely the approaches based on the Bethe variational problem and the approaches based on the mean field approximations [Wainwright and Jordan, 2008]. Learning approaches based on the former maximize in general an approximation of the likelihood function, whereas approaches based on the latter maximize in general an upper bound of the likelihood function. In [Sudderth et al., 2007] the authors show that for the models with attractive label interactions learning methods based on the Bethe variational problem actually maximize a lower bound of the likelihood function. Both Bethe methods and mean field methods are based on in general nonconvex optimization problems. There is a class of the convex approximations of the Bethe variational problem that yields a lower bound on the likelihood function value in general [Wainwright et al., 2005b]. These lower bounds complement the upper bounds yielded by the mean field approximation based methods. Bethe approximation based learning is described [Ganapathi et al., 2008]. Spanning tree based approximations, empirically investigated in [Pletscher et al., 2009] in the context of statistical image modeling, can be seen as a convex approximation of the Bethe variational problem as proposed in [Wainwright et al., 2005b]. In summary the variational inference framework offers an appealing way to arrive at approximations of the learning problem that are both tractable and convex.

One of the early approaches to approximate learning in the statistical image modeling context is based on the pseudolikelihood proposed already in [Besag, 1975] and analyzed in [Besag, 1977]. It has been proved that the pseudolikelihood approximation converges to the true parameters in the infinite data limit if the model class includes the true distribution [Gidas, 1988, Mase, 1995, Hyvärinen, 2006, Winkler, 2006]. Hence we will in the following refer to the pseudolikelihood approximation as to a consistent approximation. Analysis with respect to model mis-specification is given in [Liang and Jordan, 2008]. The pseudolikelihood function can be maximized by maximizing the negative log pseudolikelihood function that is twice continuously differentiable and convex [Winkler, 2006]. Furthermore the running time of evaluation of both the pseudolikelihood function and its first derivative is linear in the number of semantic variables in the training set [Winkler, 2006]. Since the negative log pseudolikelihood function can be minimized by iterative gradient descent algorithms, where the complexity of each iteration is upper bounded by a low degree polynomial, we refer to the approximation as being tractable in the sense of being formulated as a tractable convex optimization problem. Introductory comments on how to optimize the pseudolikelihood approximation computationally are given in [Sutton and McCallum, 2011]. Rarely the pseudolikelihood approximation has been reported to be competitive [Toutanova et al., 2003, Korč and Förstner, 2008a, Pletscher et al., 2009]. In the context of statistical image modeling the pseudolikelihood approximation is commonly viewed as performing poorly or failing completely [Kumar and Hebert, 2003, Blake et al., 2004, Sutton and Mccallum, 2005, Vishwanathan et al., 2006, Sutton and McCallum, 2007b, Hazan and Urtasun, 2010a,b]. It has been reported that the performance of the pseudolikelihood approximation can be improved by combining it with a parameter prior distribution [Kumar and Hebert, 2004a, 2006, Korč and Förstner, 2008a, Pletscher et al., 2009], the negative logarithm of which is convex and parameters of which can be chosen to be strongly convex. Summing then the convex negative log pseudolikelihood function and the strongly convex negative log prior function yields a strongly convex function [Boyd and Vandenberghe, 2004]. Hence we refer to this approximation as to the strongly convex approximation. Let us point out that since the pseudolikelihood based approximation function is twice continuously differentiable and strongly convex, the classical convergence analysis of Newton's method applies and yields an upper bound on the number of iterations required to solve the problem to a given accuracy [Boyd and Vandenberghe, 2004]. Empirical comparison of the pseudolikelihood approximation with other approximations can be found in [Kumar et al., 2005, Korč and Förstner, 2008a]. In [Kumar et al., 2005] the authors identify the pseudolikelihood based learning as one of the worst performing among the approximations under considerations. This comparison has been counterbalanced by the results in [Korč and Förstner, 2008a] and in particular in [Korč and Förstner, 2008b], where the authors show that the pseudolikelihood based learning can yield competitive results. Brief explanation of the pseudolikelihood approximation and its intuitive relation to the Gibbs sampler is given in [Sutton and McCallum, 2007b, Nowozin and Lampert, 2011, Sutton and McCallum, 2011]. In summary we have identified a pseudolikelihood based approximation of the likelihood function that is consistent tractable and strongly convex.

There are several methods related to the pseudolikelihood approximation. The piecewise pseudolikelihood described in [Sutton and McCallum, 2007b] combines piecewise training with the pseudolikelihood approximation. The pseudolikelihood approximation has been generalized in the mathematical literature to the form of the composite likelihood [Lindsay, 1988]. Similar to the pseudolikelihood approximation there are theoretical results concerning consistency of composite likelihood. Experiments with composite likelihood are reported in [Dillon and Lebanon, 2009, 2010a,b]. In [Sutton and McCallum, 2011] the authors claim that the more general composite likelihood yields better parameter estimates compared to the pseudolikelihood special case. A learning method distinct in nature however inspired by the pseudolikelihood approximation is described in [Sontag et al., 2010].

Let us now describe learning methods based on the likelihood gradient approximation. These approximations are either stochastic or deterministic. We begin with the learning method based on a stochastic approximation of the likelihood gradient. It is a Markov Chain Monte Carlo sampling inspired method proposed in [Hinton, 2002] and called the contrastive divergence. Further there are learning methods based on deterministic approximations of the likelihood gradient. Discrete approximations based on the saddle point approximation [Geiger and Girosi, 1991], pseudo-marginal approximation [McCallum et al., 2003] and the maximum marginal approximation are described in [Kumar et al., 2005]. These methods approximate the gradient at a particular parameter vector by employing approximate algorithms for the inference of a label configuration. A graph-cut algorithm based learning is described in [Szummer et al., 2008]. These approaches lead to heuristic iterative optimization methods based purely on the approximate gradient. The interdependence of approximate learning method and approximate inference method in this context is investigated empirically in [Kumar et al., 2005] and studied theoretically in [Wainwright, 2006, Wainwright and Jordan, 2008, Kulesza and Pereira, 2008]. In [Wainwright, 2006] the authors prove that likelihood gradient approximations leading to inconsistent parameter estimator can in fact be beneficial in practical situations. In [Kumar et al., 2005] the authors show that tractable state-of-the-art approaches to joint parameter learning yield satisfactory parameter estimates fast, however, obtained heuristically from approximations with oscillatory behavior. More specifically it was found in [Kumar et al., 2005] that for the maximum a posteriori probability inference the saddle point approximation based learning is the most accurate as well as time efficient. However, it was also showed that this approximation leads to a limit cycle convergence behavior dependent on the parameter initialization. As the convergence is not guaranteed, a parameter selection heuristics has to be chosen for the oscillatory case. This is the general computational drawback of learning methods based on the likelihood gradient approximation.

There are several approaches that aim at speeding up the learning of the parameters. In [Vishwanathan et al., 2006] the authors use stochastic gradient methods to accelerate the mean field approximation based learning method and the Bethe approximation based learning method. Further learning in conditional random fields can also be accelerated using piecewise training [Sutton and Mccallum, 2005] and the piecewise pseudolikelihood [Sutton and McCallum, 2007b].

In the image interpretation context parameter learning approaches commonly treat components of global models independently. Training model components independently and training model components jointly in an approximate manner is empirically compared in [Nowozin et al., 2010]. Likelihood function based formulation of parameter learning in conditional random fields leads to convex optimization problems, unless the conditional random field model components are parameterized with functions that do not preserve convexity [Boyd and Vandenberghe, 2004]. For instance in [Kumar and Hebert, 2003] the authors do formulate a model, where likelihood function based parameter learning is not convex in all model parameters. In [Kumar and Hebert, 2004a, 2006] the authors then modify the previous model formulation to arrive at a likelihood based parameter learning in the form of a convex optimization problem.

In summary the globally reasoning functions for automatic image interpretation need to be globally trained to eventually perform reasoning in a global manner. While there are concepts that we can employ for training the global functions for automatic image interpretation, exact global training is in general computationally very difficult. Our conclusion however is that the function for automatic image interpretation that is based on an internal conditional random field model is approximately trainable in a computationally tractable way. Our contribution is to identify a consistent tractable strongly convex form of the approximate learning.

In this thesis we adopt the general idea of replacing the intractable likelihood function with a both tractable and convex surrogate likelihood function, which is a general idea also studied in the context of variational inference [Wainwright and Jordan, 2008]. In this thesis we adopt a consistent tractable convex tractable surrogate likelihood in the form of the pseudolikelihood approximation [Besag, 1975] that we control with the convex parameter prior distribution [Kumar and Hebert, 2004a, 2006, Korč and Förstner, 2008a, Pletscher et al., 2009]. As we have pointed out in this section, the approximation possesses the conceptually desired property of being consistent. The approximation possess the desired computational property of being both tractable and strongly convex. It is then our goal to compare the pseudolikelihood based learning with an instance of the learning methods based on the convex approximation of the Bethe variational problem [Wainwright et al., 2005b], namely with the spanning tree based approximation described in [Pletscher et al., 2009]. This approximation also possess the desired computational property of being tractable and convex.

1.2.4 Inference of Semantic Descriptions

We briefly point to existing literature on tractable approaches to probabilistic inference in the context of the global statistical models. Next to established sources of algorithms for inferring the most probable label configuration like Markov Chain Monte Carlo methods [Winkler, 2006], there are more recent algorithms based on graph cuts [Greig et al., 1989, Boykov et al., 2001], loopy belief propagation [Pearl, 1988, Murphy et al., 1999] and tree-reweighted message passing [Wainwright et al., 2005a] that have proven to be efficient in finding approximate solutions to the problem, see [Szeliski et al., 2008] for empirical comparison. Even more recently convex relaxation approaches have proven to be a powerful alternative to the previously mentioned algorithms. Specifically, the linear programming relaxation is proposed in [Schlesinger, 1976] for a special case and independently in [Koster et al., 1998, Chekuri et al., 2001, Wainwright et al., 2005a] for the general case. In [Kumar et al., 2009, Wainwright and Jordan, 2008] the authors proved that the linear programming relaxation provides better approximation than other convex relaxations proposed recently. Let us point out that linear programs can be solved with the barrier method, a variant of the interior point methods, for which convergence analysis for self-concordant convex functions applies and hence rigorous upper bound on the complexity of obtaining a solution can be evaluated [Boyd and Vandenberghe, 2004]. The linear programming relaxation, the loopy belief propagation, the tree reweighted message passing all have a natural interpretation in the context of distributions from the exponential family and in the context of the variational inference [Wainwright and Jordan, 2008].

1.3 Overview of the Work

We now give an outline of the thesis. We begin with Chapter 2 that will delineate our development in the thesis.

In Chapter 2 we on an abstract level specify the trainable function for automatic image interpretation that we develop in this thesis. We first specify the function in terms of its input and its output. This allows us then to define an image interpretation task that the function is meant to solve. Eventually we give an abstract level description of the function's internal components that become the subject of subsequent chapters.

Chapter 3 proposes a class of the global statistical models that the trainable function for automatic image interpretation is based on. The models of the proposed class have different structure and free parametric form. The structure of the model is fixed manually or derived by some preprocessing step, for instance by using the region adjacency graph of an image partitioning [Yang et al., 2010]. The parametric form of the model is free and is learned automatically from training examples. It is our goal to formulate a class of models that should include existing models as special cases. Hence we derive existing models, like the widely adopted Potts model, as the subclasses of the proposed class of models.

Chapter 4 describes how the trainable function for automatic image interpretation learns from examples the relation between images and semantic descriptions. More specifically the function with the use of training examples automatically learns the parameters of an internal global statistical model that maps images and their semantic descriptions on a high probability value. This in expectation minimizes the interpretation error on yet unseen images. The function's internal global statistical model is a member of the class of models proposed in Chapter 3. We formulate training of the function as a consistent tractable strongly convex approximation of an intractable exact learning problem. Further we propose a way to compare performance of the function with functions that are based on the existing global statistical models like the Potts model. Specifically we describe how learning parameters of the internal global statistical model of the class of the data-dependent asymmetric interaction models proposed in Chapter 3 can be reduced to learning existing models like the Potts model by confining the learning to the subclasses of models described in Section 3.2.

Chapter 5 first describes how the trainable function for automatic image interpretation infers the semantic descriptions from newly observed images. This computational challenge translates to the problem of computing a mode of the global statistical model, that the function is based on, and learning of which is described in Chapter 4. Second chapter 5 investigates the change in the performance of the trainable function for automatic image interpretation, described in Chapter 2, that is varied first in terms of the model, it is based on, and second in terms of the approximate learning method, it is based on. The chapter further illustrates the potential relevance of this work in the field of 3D city modeling, precision agriculture and medical imaging.

Throughout Chapter 3, Chapter 4 and Section 5.1 we provide a set of simple interrelated examples that illustrate introduced concepts and most of which can be readily verified with a pocket calculator. There are two lines of examples that build on top of each other and that are meant to illuminate in a simple way the core conceptual and computational problems in this thesis. The first line of examples involves a model from the class models that the trainable function for automatic image interpretation will be based on. The second line of examples involves a model from the subclass of the conceptually simpler Potts models. We provide these examples as additional means to clarify the presented material. In case the presentation is clear enough, the examples can be skipped without breaking continuity of the presentation. The examples are largely intended for an interested reader who wishes to implement the material in the form of a computer program. Good grasp of such simple calculations is in our view a necessary prerequisite to a correct implementation.

Chapter 2

A Trainable Function for Context Sensitive Image Interpretation

This chapter aims to specify the trainable function for automatic image interpretation that we described in Chapter 1. In Section 2.1 we describe how both images and semantic descriptions are represented. This enables us in Section 2.2 to specify our function in terms of its input and its output. In Section 2.3 we formally state an image interpretation task that the trainable function for automatic image interpretation is meant to solve. Eventually in Section 2.4 we describe statistical nature of the function and specify the function on an abstract level in terms of its internal components. This will serve as an outline for our further development.

2.1 An Image Representation

In this section we describe how both images and semantic descriptions are represented in our work. Let the vector $\boldsymbol{y}$,

$$\boldsymbol{y} = [\boldsymbol{y}_0^\mathsf{T}, \boldsymbol{y}_1^\mathsf{T}, \ldots, \boldsymbol{y}_i^\mathsf{T}, \ldots, \boldsymbol{y}_{I-1}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{IC} \tag{2.1}$$

denote data from an observed image. We denote the set of *sites* from the observed image by the set $\mathcal{I}$,

$$\mathcal{I} = \{0, 1, \ldots, i, \ldots, I-1\} \tag{2.2}$$

where the scalar I denotes the number of sites. The C-dimensional real vector $\boldsymbol{y}_i \in \mathbb{R}^C$ associated with a site i can be in the following interpreted, for instance, as a 3-dimensional color vector in the RGB color space. The constant C can be thought of as the number of channels in a multi-spectral image. In this thesis we are interested in describing components of an observed image with class labels that are hidden. Here we for simplicity choose the components to be the sites of the observed image and denote the class labels associated with the sites $\mathcal{I}$ by the vector $\boldsymbol{x}$,

$$\boldsymbol{x} = [x_0, x_1, \ldots, x_i, \ldots, x_{I-1}]^\mathsf{T} \in \mathcal{K}^I \tag{2.3}$$

A class label at a site i, denoted by the scalar x_i, takes value from the set $\mathcal{K}$,

$$\mathcal{K} = \{0, 1, \ldots, k, \ldots, K-1\} \tag{2.4}$$

of class labels, where the scalar K is a finite number of class labels.

Let us maintain that in our development we do not restrict ourselves to regular arrangements of measurements. It is our goal to develop a trainable function for automatic image interpretation that can accept irregular arrangements of measurements, described in Chapter 1, as its input.

2.2 A Trainable Function

In this section we specify our trainable function for automatic image interpretation in terms of its input and its output.

We start by describing involved data. We consider a data set $\mathcal{D}$,

$$\mathcal{D} = \left\{\boldsymbol{x}^l, \boldsymbol{y}^l\right\}_{l\in\mathcal{L}} \tag{2.5}$$

that consists of images $\boldsymbol{y}^l$ in equation (2.1) and corresponding ground truth label configurations $\boldsymbol{x}^l$ in equation (2.3). Both images and ground truth label configurations are indexed with the index set $\mathcal{L}$,

$$\mathcal{L} = \{1, \ldots, l, \ldots, L\} \tag{2.6}$$

The scalar L denotes the number of labeled images $(\boldsymbol{x}^l, \boldsymbol{y}^l)$. Labeled images are typically image examples hand labeled by a human expert.

We split the index set $\mathcal{L}$ in two disjoint index sets $\mathcal{M}$ and $\mathcal{N}$. Thus the data set comprises two disjoint sets, namely the *training set* $\mathcal{D}_\mathcal{M}$,

$$\mathcal{D}_\mathcal{M} = \{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m\in\mathcal{M}} \tag{2.7}$$

and the *test set* $\mathcal{D}_\mathcal{N}$,

$$\mathcal{D}_\mathcal{N} = \{\boldsymbol{x}^n, \boldsymbol{y}^n\}_{n\in\mathcal{N}} \tag{2.8}$$

We specify the trainable function f,

$$\hat{\boldsymbol{x}}^n = f(\boldsymbol{y}^n, \mathcal{D}_\mathcal{M}) \tag{2.9}$$

for automatic image interpretation as a transformation that maps the training set $\mathcal{D}_\mathcal{M}$ in equation (2.7) and a test image $\boldsymbol{y}^n$ from the test set $\mathcal{D}_\mathcal{N}$ in equation (2.8) to a label configuration $\hat{\boldsymbol{x}}^n$, which ideally should be equal or at least close to the ground truth label configuration $\boldsymbol{x}^n$. The assumption is that the trainable function f for automatic image interpretation can eventually also interpret other images coming from the same class of images as the training and the test images.

In this section we have specified the trainable function for automatic image interpretation in terms of its input and its output. Our next step is to first to specify the task the function is meant to solve and then to describe the function in terms of its internal components.

2.3 Image Interpretation Tasks

Here we describe the task that the function f in equation (2.9) is meant to solve. This involves specification of a criterion by means of which we evaluate performance of the function on the task.

We test the performance of the function f in equation (2.9) by evaluating a criterion that compares an inferred test label configuration $\hat{\boldsymbol{x}}^n$ in equation (2.9) with a ground truth test label configuration $\boldsymbol{x}^n$ in equation (2.8). Our task is to find a function that in terms of the criterion performs well on the test set $\mathcal{D}_\mathcal{N}$, while during testing having access to the training set $\mathcal{D}_\mathcal{M}$ and a single unlabeled test image $\boldsymbol{y}^n \in \mathcal{D}_\mathcal{N}$ only.

We consider two task evaluation criteria. The first criterion is the *class error* $e_\mathcal{N}$,

$$e_\mathcal{N} = \frac{1}{\sum_{n\in\mathcal{N}} I^n} \sum_{n\in\mathcal{N}} \sum_{i\in\mathcal{I}^n} (1 - \delta(\hat{x}_i^n - x_i^n)) \tag{2.10}$$

that counts the number of misclassified sites and yields an estimate of the probability of misclassification. The function $\delta(\cdot)$ denotes the Kronecker delta. Here we denote the set of sites in the n-th image by the set $\mathcal{I}^n = \{0, 1, \ldots, i, \ldots, I^n - 1\}$, where the scalar I^n denotes the number of sites in the n-th image. The class error is a simple and appropriate performance measure in case when all classes are equally important and when at the same time each of the classes is represented by equal number of sites. The more these two conditions are violated the less appropriate the measure is.

The segmentation accuracy, that was introduced in [Everingham et al., 2010], is a criterion of evaluation that we mention as an alternative. For each class k separately we compute three quantities, namely the number of true positives N_k^{tp},

$$N_k^{\mathrm{tp}} = \sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{I}^n} \delta(\hat{x}_i^n - k)\delta(x_i^n - k)$$

of sites correctly classified as a class k, the number of false negatives N_k^{fn},

$$N_k^{\mathrm{fn}} = \sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{I}^n} (1 - \delta(\hat{x}_i^n - k))\delta(x_i^n - k)$$

of sites falsely identified with a class other than the class k and the number of false positives N_k^{fp},

$$N_k^{\mathrm{fp}} = \sum_{n \in \mathcal{N}} \sum_{i \in \mathcal{I}^n} \delta(\hat{x}_i^n - k)(1 - \delta(x_i^n - k))$$

of sites falsely identified with the class k. We evaluate the class k specific segmentation accuracy $a_{\mathcal{N},k}$ as

$$a_{\mathcal{N},k} = \frac{N_k^{\mathrm{tp}}}{N_k^{\mathrm{fn}} + N_k^{\mathrm{tp}} + N_k^{\mathrm{fp}}}$$

The number N_k^{tp} in the nominator can be viewed as size of intersection of the set of sites associated with the class k in ground truth label configurations and the set of sites associated with the class k in the inferred label configurations. The sum in the denominator can be viewed as size of union of the two sets. Thus we can phrase the class k specific segmentation accuracy as an intersection-over-union score. Finally we evaluate the *segmentation accuracy* $a_{\mathcal{N}}$,

$$a_{\mathcal{N}} = \frac{1}{K} \sum_{k \in \mathcal{K}} a_{\mathcal{N},k} \tag{2.11}$$

as the mean of the class specific segmentation accuracy values. The segmentation accuracy $a_{\mathcal{N}}$ is an appropriate performance measure still in the case when all classes are equally important, however also in the case where each of the classes is represented by different number of sites.

We have described the task that the trainable function for automatic image interpretation is meant to solve as to in terms of a criterion perform well on the test set and we have included examples of two such criteria. In the following we on an abstract level specify the internal components of the trainable function for automatic image interpretation.

2.4 A Statistical Approach

In this section we specify the trainable function f for automatic image interpretation in equation (2.9) in terms of its internal components. These components will outline our further development.

The function f in equation (2.9) makes use of the training set $\mathcal{D}_{\mathcal{M}} = \{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m\in\mathcal{M}}$ in equation (2.7) to map a test image $\boldsymbol{y}^n$ from the test set $\mathcal{D}_{\mathcal{N}} = \{\boldsymbol{x}^n, \boldsymbol{y}^n\}_{n\in\mathcal{N}}$ in equation (2.8) on a label configuration $\hat{\boldsymbol{x}}^n$. The function f is based on an internal statistical image model and it treats the image data as samples of a random vector $\underline{\boldsymbol{y}}$ that takes values in the continuous space $\mathbb{R}^{IC}$ of image data vectors. The function f treats image label configurations as samples of a random vector $\underline{\boldsymbol{x}}$ that takes values in the discrete space $\mathcal{K}^I$ of image label configurations. The function f is based on the internal statistical model that relates a hidden event $\{\underline{\boldsymbol{x}}^n = \boldsymbol{x}\}$, that a test image label configuration $\underline{\boldsymbol{x}}^n$ takes a value $\boldsymbol{x}$, to an observed event $\{\underline{\boldsymbol{y}}^n = \boldsymbol{y}^n\}$, that a test image $\underline{\boldsymbol{y}}^n$ takes a value $\boldsymbol{y}^n$. The function f is based on an internal class $\mathcal{P}$,

$$\mathcal{P} = \{p_u\} \tag{2.12}$$

of the global statistical models forming the global probability mass functions p_u,

$$p_u : \mathcal{K}^I \times \mathbb{R}^{IC} \to \mathbb{R}_{++} \tag{2.13}$$

that map a label configuration $\boldsymbol{x} \in \mathcal{K}^I$ and an image $\boldsymbol{y} \in \mathbb{R}^{IC}$ on a strictly positive number. The global conditional probability mass function p_u in equation (2.13) is parametrized by the vector $\boldsymbol{u}$,

$$\boldsymbol{u} \in \mathbb{R}^{D_u} \tag{2.14}$$

of free parameters. The conditional probability mass function p_u is normalized such that the condition

$$\sum_{\boldsymbol{x}\in\mathcal{K}^I} p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u}) = 1 \tag{2.15}$$

is true and where we switch from the notation $p_u(\boldsymbol{x} \mid \boldsymbol{y})$ to the notation $p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u})$ that we keep from now on. Free model parameters $\boldsymbol{u}$ are unknown and need to be learned from the labeled training examples. The function f makes use of the training set $\mathcal{D}_{\mathcal{M}} = \{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m\in\mathcal{M}}$ in equation (2.7) to learn free parameters $\boldsymbol{u}^{\mathrm{ML}}$,

$$\boldsymbol{u}^{\mathrm{ML}} \in \arg\max_{\boldsymbol{u}} p(\{\boldsymbol{x}^m\}_{m\in\mathcal{M}} \mid \{\boldsymbol{y}^m\}_{m\in\mathcal{M}}, \boldsymbol{u}) \tag{2.16}$$

of the internal statistical model according to the maximum likelihood (ML) principle by maximizing the conditional likelihood of parameters $\boldsymbol{u}$, given the training set $\mathcal{D}_{\mathcal{M}} = \{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m\in\mathcal{M}}$. In this way the function f learns parameters $\boldsymbol{u}^{\mathrm{ML}}$ of a particular conditional probability mass function $p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u}^{\mathrm{ML}})$ that assigns joint events $\{\underline{\boldsymbol{x}}^m = \boldsymbol{x}^m, \underline{\boldsymbol{y}}^m = \boldsymbol{y}^m\}$ from the training set $\mathcal{D}_{\mathcal{M}}$ to a high conditional probability $P\{\underline{\boldsymbol{x}}^m = \boldsymbol{x}^m \mid \underline{\boldsymbol{y}}^m = \boldsymbol{y}^m\}$. Here we learn a statistical model independently of a loss function. The learned model would ideally represent the true statistical relationship between the modeled quantities and hence ideally it would yield optimal decisions under a particular loss function. That would then be true for each particular loss function. Based on the learned internal statistical model the function f infers an event $\{\underline{\boldsymbol{x}}^n = \hat{\boldsymbol{x}}^n\}$,

$$\hat{\boldsymbol{x}}^n \in \arg\max_{\boldsymbol{x}} p(\boldsymbol{x} \mid \boldsymbol{y}^n, \boldsymbol{u}^{\mathrm{ML}}) \tag{2.17}$$

that maximizes the joint conditional probability mass function $p(\boldsymbol{x} \mid \boldsymbol{y}^n, \boldsymbol{u}^{\mathrm{ML}})$ of a label configuration $\boldsymbol{x}$ given an observed test image $\boldsymbol{y}^n$ from the test set $\mathcal{D}_{\mathcal{N}} = \{\boldsymbol{x}^n, \boldsymbol{y}^n\}_{n\in\mathcal{N}}$ in equation (2.8) and given the learned free parameters $\boldsymbol{u}^{\mathrm{ML}}$. In probability terms the function f infers an event $\{\underline{\boldsymbol{x}}^n = \hat{\boldsymbol{x}}^n\}$, that maximizes the conditional probability $P\{\underline{\boldsymbol{x}}^n = \boldsymbol{x} \mid \underline{\boldsymbol{y}}^n = \boldsymbol{y}^n\}$ of the hidden test event $\{\underline{\boldsymbol{x}}^n = \boldsymbol{x}\}$, given the observed test event $\{\underline{\boldsymbol{y}}^n = \boldsymbol{y}^n\}$. Having adopted the maximum conditional probability mass as a criterion we imply the 0-1 loss function. However adopting such criterion for the evaluation on the test images with many thousands of sites would in practice most likely be uninformative. Mislabeling of only one out of the many thousands of correctly labeled image sites would always yield an error of one. An error of zero would only be yielded when all the many thousands of sites would be labeled correctly, which in practice will likely be a rare event. We can view the sitewise class error suggested as the evaluation criterion in equation (2.10) as an approximation of the global 0-1 loss function.

We have specified the trainable function f for automatic image interpretation in equation (2.9) in terms of its internal components. The first component is conceptual and involves the representation of the relationship between images and their semantic descriptions in terms of a global statistical model. The second component is computational and involves learning part of the global statistical model from examples. The third component is again computational and involves the inference of a semantic description from an observed image. These components will be further described in the following.

Chapter 3

A Class of Global Discriminative Models for Context Sensitive Image Interpretation

This chapter deals with the core conceptual problem of how should the trainable function for automatic image interpretation, specified on abstract level in Chapter 2, in probability terms represent the relationship between images and their semantic descriptions. We propose a class of global statistical models, that the trainable function for automatic image interpretation is based on, and that relate images to semantic descriptions in context-sensitive manner. In Section 3.1 we propose the class of such statistical models that includes models of different structure and free parametric form. Structure of the model is fixed manually according to our experience regarding modeling of image semantic descriptions in general or according to our insight regarding modeling image semantic descriptions in a particular application domain. The fixed structure of the model can also be derived by a preprocessing step, for instance, as we pointed out previously, by using the region neighborhood graph of an image partitioning. The parametric form of the model is free and represents the ability of the model to capture automatically what is unknown and specific to a particular image interpretation task from the training examples. It is our goal to formulate a class of models that should include existing models as special cases. In Section 3.2 we derive existing models like the widely adopted Potts model as the subclasses of the proposed class of models.

3.1 A Class of Global Discriminative Models

The contribution of this section is a class of the global statistical models that the trainable function f in equation (2.9) for automatic image interpretation is based on. The internal global statistical model of the proposed class forms a conditional probability mass function p in equation (2.13) and models conditional probability mass of a label configuration $\boldsymbol{x} \in \mathcal{K}^I$ in equation (2.3) given an image $\boldsymbol{y} \in \mathbb{R}^{IC}$ in equation (2.1). We propose a class of conditional probability mass functions p of different structure and free parametric form. The structure of the model is given by a graph that is fixed according to the statistical dependence among the components of the vector $\boldsymbol{x}$. The parametric form is governed by the vector $\boldsymbol{u} \in \mathbb{R}^{D_u}$ in equation (2.14) of free parameters that need to be learned from the training set in equation (2.7).

We propose a class $\mathcal{P}$,

$$\mathcal{P} = \left\{ p \mid p : \mathcal{K}^I \times \mathbb{R}^{IC} \times \mathbb{R}^{D_u} \to \mathbb{R}_{++}, \sum_{\boldsymbol{x} \in \mathcal{K}^I} p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u}) = 1 \right\} \tag{3.1}$$

of data-dependent asymmetric interaction models with pairwise label interaction that is data-dependent label-pair-specific and asymmetric. The models are conditional probability mass functions in the form of multi-class conditional random fields p. The form of the conditional probability mass functions in equation (3.1) is by far to general to be handled. Therefore we restrict the conditional probability mass functions to a specific form of conditional random fields. We describe the form of the proposed class $\mathcal{P}$ of conditional random fields step by step in the following.

A multi-class conditional random field (CRF) models the conditional probability mass function $p : \mathcal{K}^I \times \mathbb{R}^{IC} \times \mathbb{R}^{D_u} \to \mathbb{R}_{++}$,

$$p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u}) = \frac{1}{Z(\boldsymbol{y}, \boldsymbol{u})} \exp\left(-E(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u})\right) \tag{3.2}$$

of a label configuration $\boldsymbol{x}$ given observed image data $\boldsymbol{y}$ and given the vector $\boldsymbol{u} \in \mathbb{R}^{D_u}$ of free model parameters. Free model parameters $\boldsymbol{u}$ are unknown and need to be learned from the training examples. The so called partition function $Z : \mathbb{R}^{IC} \times \mathbb{R}^{D_u} \to \mathbb{R}_{++}$,

$$Z(\boldsymbol{y}, \boldsymbol{u}) = \sum_{\boldsymbol{x} \in \mathcal{K}^I} \exp\left(-E(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u})\right) \tag{3.3}$$

maps image data $\boldsymbol{y}$ and model parameters $\boldsymbol{u}$ on a strictly positive real number. For fixed value $\boldsymbol{y}$ and for fixed value $\boldsymbol{u}$ the partition function plays the role of a normalization constant and ensures that the equality

$\sum_{\boldsymbol{x}\in\mathcal{K}^I} p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u}) = 1$ is true. The normalization constant is in general intractable to evaluate because the number of summands in equation (3.3) grows exponentially with the number I of sites. The so called energy function $E : \mathcal{K}^I \times \mathbb{R}^{IC} \times \mathbb{R}^{D_u} \to \mathbb{R}$ is a function that for given model parameters $\boldsymbol{u}$ and image data $\boldsymbol{y}$ maps a label configuration $\boldsymbol{x}$ on a real number.

In principle any CRF with discrete random variables can be transformed into a CRF with label interactions only between pairs of variables that is equivalent. With equivalent we mean that the resulting function represents the same conditional probability mass function. See for instance [Yedidia et al., 2002] or the derivation in Appendix E3 in [Wainwright and Jordan, 2008]. However from practical viewpoint we now restrict our consideration to the pairwise CRF models and disregard the fact that any CRF model has an equivalent pairwise representation. Typically we specify an undirected graphical model with an undirected graph. In the following we specify an undirected graphical model with a graph that is directed. This is not to be confused with a directed graphical model, specified with a directed graph. A pairwise CRF has an energy function E,

$$E(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u}) = \sum_{i\in\mathcal{I}} E_i(x_i \mid \boldsymbol{y}, \boldsymbol{w}_i) + \sum_{(i,i')\in\mathcal{E}} E_{ii'}(x_i, x_{i'} \mid \boldsymbol{y}, \boldsymbol{v}_{ii'}) \tag{3.4}$$

that can be expressed as a sum of functions, defined on a directed graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$. The graph $\mathcal{G}$ is defined on the set $\mathcal{I}$ of sites and the set $\mathcal{E}$, $\mathcal{E} \subset \mathcal{I} \times \mathcal{I}$, denotes the set of directed edges, where the oriented pair (i, i') denotes a directed edge being associated with the sites i and i' and where the oriented pair (i', i) is not included in the set $\mathcal{E}$. The unary term E_i of the energy function E is a function that for given unary parameters $\boldsymbol{w}_i$ and image data $\boldsymbol{y}$ maps single label x_i on a real number. Functional form and parametrization $\boldsymbol{w}_i$ of the unary term E_i are specific to the location i. The pairwise term $E_{ii'}$ of the energy function E is a function that for given pairwise parameters $\boldsymbol{v}_{ii'}$ and image data $\boldsymbol{y}$ maps label pair $(x_i, x_{i'})$ on a real number. Functional form and parametrization $\boldsymbol{v}_{ii'}$ of the pairwise term $E_{ii'}$ are specific to the location pair $(i, i') \in \mathcal{E}$. The unary and the pairwise terms of the energy function are in the literature sometimes also referred to as the unary and the pairwise log-potential functions or simply as log-potentials. The parameter vector $\boldsymbol{u}$, $\boldsymbol{u} = [\ldots, \boldsymbol{w}_i^\mathsf{T}, \ldots, \boldsymbol{v}_{ii'}^\mathsf{T}, \ldots]^\mathsf{T}$, comprises unary parameters $\boldsymbol{w}_i$ and pairwise parameters $\boldsymbol{v}_{ii'}$.

A homogeneous and isotropic CRF has an energy function of the form

$$E(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u}) = \sum_{i\in\mathcal{I}} E_1(x_i \mid \boldsymbol{y}, \boldsymbol{w}) + \sum_{(i,i')\in\mathcal{E}} E_2(x_i, x_{i'} \mid \boldsymbol{y}, \boldsymbol{v}) \tag{3.5}$$

Functional form and unary parameters $\boldsymbol{w}$ of the unary term E_1 are now common to all sites i. Also functional form and pairwise parameters $\boldsymbol{v}$ of the pairwise term E_2 are now common to all site pairs $(i, i') \in \mathcal{E}$. We express the free model parameters $\boldsymbol{u}$,

$$\boldsymbol{u} = \left[\boldsymbol{w}^\mathsf{T}, \boldsymbol{v}^\mathsf{T}\right]^\mathsf{T} \in \mathbb{R}^{D_u} \tag{3.6}$$

in terms of unary parameters $\boldsymbol{w}$ and in terms of pairwise parameters $\boldsymbol{v}$. The independence on the location makes the energy function homogeneous and the independence on the location pair makes the energy function isotropic. A homogeneous and isotropic energy function seems to be appropriate in cases, where the camera location and orientation is not fixed, and reduces substantially the length of the vector $\boldsymbol{u}$ of free model parameters. This is sometimes referred to as parameter sharing.

Finally a CRF with affine log-potential functions and with pairwise label interaction that is label-pair-specific, data-dependent and asymmetric includes energy function of the form

$$E(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u}) = -\sum_{i \in \mathcal{I}} \boldsymbol{w}_{x_i}^\mathsf{T} \boldsymbol{h}_i(\boldsymbol{y}) - \sum_{(i,i') \in \mathcal{E}} \boldsymbol{v}_{x_i x_{i'}}^\mathsf{T} \boldsymbol{\mu}_{ii'}(\boldsymbol{y}) \tag{3.7}$$

The unary term $E_1(x_i \mid \boldsymbol{y}, \boldsymbol{w})$ in equation (3.5) takes the form $E_1 : \mathcal{K} \times \mathbb{R}^{IC} \times \mathbb{R}^{KD_w} \to \mathbb{R}$ of an affine function $\boldsymbol{w}_{x_i}^\mathsf{T} \boldsymbol{h}_i(\boldsymbol{y})$ in equation (3.7). The functional value of the unary term can be viewed as a cost of assigning a site i with a label x_i. The unary term at a site i is thus associated with the K affine functions $\boldsymbol{w}_k^\mathsf{T} \boldsymbol{h}_i(\boldsymbol{y})$, parametrized by the unary parameter vector $\boldsymbol{w}$,

$$\boldsymbol{w} = [\boldsymbol{w}_0^\mathsf{T}, \ldots, \boldsymbol{w}_k^\mathsf{T}, \ldots, \boldsymbol{w}_{K-1}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{KD_w} \tag{3.8}$$

The affine function $\boldsymbol{w}_k^T \boldsymbol{h}_i(y)$ evaluating the assignment with a class label k is parametrized by the vector $\boldsymbol{w}_k$,

$$\boldsymbol{w}_k = [w_{k;0}, \ldots, w_{k;j}, \ldots, w_{k;D_w-1}]^\mathsf{T} \in \mathbb{R}^{D_w} \tag{3.9}$$

Here we denote the component of the vector $\boldsymbol{w}$ at the index $k \cdot D_w + j$ with the scalar $w_{k;j}$. Dimension D_w of the vector $\boldsymbol{w}_k$ is determined by the dimension of the unary feature vector $\boldsymbol{h}_i(\boldsymbol{y})$,

$$\boldsymbol{h}_i(\boldsymbol{y}) = [1,\ h_{i;1}(\boldsymbol{y}), \ldots, h_{i;D_w-1}(\boldsymbol{y})]^\mathsf{T} \in \mathbb{R}^{D_w} \tag{3.10}$$

that describes image data $\boldsymbol{y}$ with respect to a site i. We expand the original unary feature vector by introducing a dummy value $h_{i;0}(\boldsymbol{y}) = 1$ to accommodate offset of the affine function in compact notation. Hence, we can view the function $\boldsymbol{w}_k^T \boldsymbol{h}_i(y)$ as an affine function in the original unary feature vector space $\mathbb{R}^{D_w-1}$ or as a linear function in the expanded unary feature vector space $\mathbb{R}^{D_w}$. The pairwise term $E_2(x_i, x_{i'} \mid \boldsymbol{y}, \boldsymbol{v})$ in equation (3.5) takes in equation (3.7) the form $E_2 : \mathcal{K}^2 \times \mathbb{R}^{IC} \times \mathbb{R}^{K^2 D_v} \to \mathbb{R}$ of an affine function $\boldsymbol{v}^{\mathsf{T}}_{x_i x_{i'}} \boldsymbol{\mu}_{ii'}(\boldsymbol{y})$. The functional value of the pairwise term can be viewed as a cost of assigning a site pair (i, i') with a label pair $(x_i, x_{i'})$. A pairwise term at a site pair (i, i') is associated with the K^2 affine functions $\boldsymbol{v}^{\mathsf{T}}_{kk'} \boldsymbol{\mu}_{ii'}(\boldsymbol{y})$, parametrized by the pairwise parameter vector $\boldsymbol{v}$,

$$\boldsymbol{v} = [\boldsymbol{v}^{\mathsf{T}}_{00}, \ldots, \boldsymbol{v}^{\mathsf{T}}_{kk'}, \ldots, \boldsymbol{v}^{\mathsf{T}}_{K-1K-1}]^{\mathsf{T}} \in \mathbb{R}^{K^2 D_v} \tag{3.11}$$

The affine function $\boldsymbol{v}^{\mathsf{T}}_{kk'} \boldsymbol{\mu}_{ii'}(\boldsymbol{y})$ evaluating the assignment with a class label pair (k, k') is parametrized by the vector $\boldsymbol{v}_{kk'}$,

$$\boldsymbol{v}_{kk'} = [v_{kk';0}, \ldots, v_{kk';j}, \ldots, , v_{kk';D_v-1}]^{\mathsf{T}} \in \mathbb{R}^{D_v} \tag{3.12}$$

Here we denote the component of the vector $\boldsymbol{v}$ at the index $(k \cdot K + k') \cdot D_v + j$ with the scalar $v_{kk';j}$. Dimension D_v of the vector $\boldsymbol{v}_{kk'}$ is determined by the dimension of the pairwise feature vector $\boldsymbol{\mu}_{ii'}(\boldsymbol{y})$,

$$\boldsymbol{\mu}_{ii'}(\boldsymbol{y}) = [1,\ \mu_{ii';1}(\boldsymbol{y}), \ldots, \mu_{ii';D_v-1}(\boldsymbol{y})]^{\mathsf{T}} \in \mathbb{R}^{D_v} \tag{3.13}$$

that describes image data $\boldsymbol{y}$ with respect to a site pair (i, i'). Again for compactness we expand the original pairwise feature vector space by introducing a dummy value $\mu_{ii';0}(\boldsymbol{y}) = 1$. Hence, we can view the function $\boldsymbol{v}^{\mathsf{T}}_{kk'} \boldsymbol{\mu}_{ii'}(\boldsymbol{y})$ as an affine function in the original pairwise feature vector space $\mathbb{R}^{K^2 D_v - 1}$ or as a linear function in the expanded pairwise feature vector space $\mathbb{R}^{K^2 D_v}$.

We evaluate an example of the energy function in equation (3.7) and an example of the conditional probability mass function in equation (3.2) of the class $\mathcal{P}$ of models in equation (3.1). The example involves two semantic variables and two observations. The two semantic variable $\mathcal{P}$ CRF is a recurrent theme that we repeatedly use in the rest of this work to illuminate the presented material.

Example 3.1 (Two variable $\mathcal{P}$ CRF)

We adopt the image representation according to Section 2.1 and let the set $\mathcal{I}$ of pixel sites contain two sites 0 and 1. We let the pixel intensities and the pixel class labels both take values from the set $\mathcal{K} = (0, 1)$. Let the two pixel intensities $y_0 = 1$ and $y_1 = 0$ be given and let the associated pixel class labels be denoted by x_0 and x_1. We set the unary feature vector $\boldsymbol{h}_0(\boldsymbol{y}) = [1,\ y_0]^{\mathsf{T}}$, the unary feature vector $\boldsymbol{h}_1(\boldsymbol{y}) = [1,\ y_1]^{\mathsf{T}}$ and the pairwise feature vector $\boldsymbol{\mu}_{01}(\boldsymbol{y}) = [1,\ |y_0 - y_1|]^{\mathsf{T}}$. This is a symmetric model. Alternatively we could model $\boldsymbol{\mu}_{01}(\boldsymbol{y}) = [1,\ y_0 - y_1]^{\mathsf{T}}$, which then would be asymmetric. We consider *the CRF energy function in equation (3.7) that is specified by the parameter vector* $\boldsymbol{u} \in \mathbb{R}^{12}$*,*

$$\boldsymbol{u}^\mathsf{T} = [\boldsymbol{w}^\mathsf{T}, \boldsymbol{v}^\mathsf{T}] = [[\boldsymbol{w}_0^\mathsf{T}, \boldsymbol{w}_1^\mathsf{T}], [\boldsymbol{v}_{00}^\mathsf{T}, \boldsymbol{v}_{10}^\mathsf{T}, \boldsymbol{v}_{01}^\mathsf{T}, \boldsymbol{v}_{11}^\mathsf{T}]] = [[w_{0;0}, w_{0;1}], \dots, [v_{11;0}, v_{11;1}]]$$

and we choose to set the parameter vector

$$\boldsymbol{u} = [[-0.5, 1], [0.5, -1], [-0.5, 1], [0.5, -1], [0.5, -1], [-0.5, 1]]^\mathsf{T} \in \mathbb{R}^{12} \tag{3.14}$$

where we do not choose to set the dependent components to 0*, but to other real value. In order to fully specify the energy function in equation (3.7) we let the set* $\mathcal{E}$ *contain the pixel site pair* $(0, 1)$*. We compute local contributions of the unary energy function terms*

$$\begin{array}{rcr} E_1(x_0 = 0 \mid \boldsymbol{y}, \boldsymbol{w}) = \boldsymbol{w}_0^\mathsf{T} \boldsymbol{h}_0(\boldsymbol{y}) & = & 0.5 \\ E_1(x_0 = 1 \mid \boldsymbol{y}, \boldsymbol{w}) = \boldsymbol{w}_1^\mathsf{T} \boldsymbol{h}_0(\boldsymbol{y}) & = & -0.5 \\ E_1(x_1 = 0 \mid \boldsymbol{y}, \boldsymbol{w}) = \boldsymbol{w}_0^\mathsf{T} \boldsymbol{h}_1(\boldsymbol{y}) & = & -0.5 \\ E_1(x_1 = 1 \mid \boldsymbol{y}, \boldsymbol{w}) = \boldsymbol{w}_1^\mathsf{T} \boldsymbol{h}_1(\boldsymbol{y}) & = & 0.5 \end{array}$$

and local contributions of the pairwise energy function terms

$$\begin{array}{rcr} E_2(x_0 = 0, x_1 = 0 \mid \boldsymbol{y}, \boldsymbol{v}) = \boldsymbol{v}_{00}^\mathsf{T} \boldsymbol{\mu}_{01}(\boldsymbol{y}) & = & 0.5 \\ E_2(x_0 = 1, x_1 = 0 \mid \boldsymbol{y}, \boldsymbol{v}) = \boldsymbol{v}_{10}^\mathsf{T} \boldsymbol{\mu}_{01}(\boldsymbol{y}) & = & -0.5 \\ E_2(x_0 = 0, x_1 = 1 \mid \boldsymbol{y}, \boldsymbol{v}) = \boldsymbol{v}_{01}^\mathsf{T} \boldsymbol{\mu}_{01}(\boldsymbol{y}) & = & -0.5 \\ E_2(x_0 = 1, x_1 = 1 \mid \boldsymbol{y}, \boldsymbol{v}) = \boldsymbol{v}_{11}^\mathsf{T} \boldsymbol{\mu}_{01}(\boldsymbol{y}) & = & 0.5 \end{array}$$

We add these values according to equation (3.7) to obtain a value of the overall energy function. Overall energy function values for four possible labellings of the CRF are listed in table 3.1. Energy values in table 3.1 allow us to evaluate the partition function in equation (3.3), here the value of the partition function is $Z(\boldsymbol{y}, \boldsymbol{u}) = 6.3013$*, and subsequently the probability mass in equation (3.2). Probability masses of the four possible labeling are listed in table 3.1.* $\ast$

In summary the conditional random fields p in the class $\mathcal{P}$ proposed in equation (3.1) and further specified in equation (3.7) are pairwise homogeneous and isotropic models. They posses affine log-potential functions and include pairwise label interaction that is data-dependent, label-pair-specific and asymmetric. In the following we refer to the the class $\mathcal{P}$ as to the class $\mathcal{P}$ of the data-dependent label-pair-specific and asymmetric interaction models.

Table 3.1: *Two variable $\mathcal{P}$ CRF.* Energy function values and probability masses of four possible labellings of CRF in example 3.1. The energy values are computed using equation (3.7) and the probability masses are computed using equation (3.2). The partition function involved in equation (3.2) is computed as in equation (3.3). Here the value of the partition function is $Z(\boldsymbol{y}, \boldsymbol{u}) = 6.3013$.

Labeling $\boldsymbol{x}$		Energy	Probability
x_0	x_1		mass
0	0	0.5	0.0963
1	0	-1.5	0.7112
0	1	0.5	0.0963
1	1	0.5	0.0963

3.2 Subclasses of Models

In this section we derive existing model formulations reviewed in Section 1.2.2 like the widely adopted Potts model as the subclasses of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1). Further we clarify the relation between alternative parameterizations of the model in [Kumar et al., 2005] that are present in [Kumar and Hebert, 2004a,b, 2006] as discussed in Section 1.2.2.

3.2.1 Data-dependent Pairwise Terms

We derive models with data-dependent label interactions reviewed in Section 1.2.2 as the subclasses of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1). Namely we derive the subclass of the data-dependent symmetric interaction models and the subclass of the contrast sensitive Potts models.

The class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1) comprises the conditional random fields with the affine log-potential functions in equation (3.7). The affine log-potential functions are parametrized with the vector $\boldsymbol{u} = [\boldsymbol{w}^\mathsf{T}, \boldsymbol{v}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{D_u}$ in equation (3.6) of free parameters composed of the unary parameter vector $\boldsymbol{w} \in \mathbb{R}^{KD_w}$ in equation (3.8) and of the pairwise parameter vector $\boldsymbol{v} \in \mathbb{R}^{K^2 D_v}$ in equation (3.11). Hence, the parameter space dimension $D_u = KD_w + K^2 D_v$. We derive the subclasses of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models by imposing linear equality constraints on the pairwise parameter vector

$\boldsymbol{v} \in \mathbb{R}^{K^2 D_v}$. Models from a particular subclass are then parametrized by vectors forming a subspace in the parameter space $\mathbb{R}^{D_u}$ of the class $\mathcal{P}$. More specifically models from a particular subclass are parametrized by vectors forming a subspace in the pairwise parameter space $\mathbb{R}^{D_v}$ of the class $\mathcal{P}$. Subspaces of the unary parameter space $\mathbb{R}^{D_w}$ of the class $\mathcal{P}$ will not play a role here.

We now derive the subclass of the data-dependent symmetric interaction models reviewed in Section 1.2.2. We impose linear equality constraints

$$\boldsymbol{v}_{kk'} = \boldsymbol{v}_{k'k} \qquad \forall k, \forall k', k \neq k' \tag{3.15}$$

on pairwise parameters $\boldsymbol{v}$ and obtain the subclass $\mathcal{P}_1$,

$$\mathcal{P}_1 = \{p \mid p \in \mathcal{P} \text{ s.t. } (3.15)\} \tag{3.16}$$

of the data-dependent symmetric interaction models with pairwise label interaction that is data-dependent, label-pair-specific and symmetric. The subclass $\mathcal{P}_1$ of the data-dependent symmetric interaction models in equation (3.16) is parametrized by points forming a subspace in the pairwise parameter vector space of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1). $\frac{1}{2}(K^2 - K)D_v$ linear equality constraints in equation (3.15) form a $\frac{1}{2}(K^2 + K)D_v$-dimensional subspace in the pairwise parameter vector space $\mathbb{R}^{K^2 D_v}$ of the class $\mathcal{P}$. Let us now clarify the relation between alternative parameterizations of the model from this subclass in [Kumar et al., 2005] that are present in [Kumar and Hebert, 2004a,b, 2006] as discussed in Section 1.2.2. In [Kumar et al., 2005] the authors propose to parametrize the energy function as

$$E(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{u}) = -\sum_{i \in \mathcal{I}} \log p_1(x_i|\boldsymbol{y}, \boldsymbol{w}) - \sum_{(i,i') \in \mathcal{E}} \log p_2(x_i, x_{i'}|\boldsymbol{y}, \boldsymbol{v}) \tag{3.17}$$

We include the energy function using our notation. In [Kumar et al., 2005] the authors propose to model the unary term $E_1(x_i \mid \boldsymbol{y}, \boldsymbol{w})$ in equation (3.5) using a model of the conditional probability mass $p_1(x_i|\boldsymbol{y}, \boldsymbol{w})$ of a class label x_i in equation (3.17). Similarly the authors propose to model the pairwise term $E_2(x_i, x_{i'} \mid \boldsymbol{y}, \boldsymbol{v})$ in equation (3.5) using a model of the conditional probability mass $p_2(x_i, x_{i'}|\boldsymbol{y}, \boldsymbol{v})$ of a pair $(x_i, x_{i'})$ of class labels in equation (3.17). The authors propose to model the conditional probability mass functions with multi-class logistic regression model for classification, a form of the generalized linear model. The alternative parameterizations of the model in [Kumar et al., 2005] that are present in [Kumar and Hebert, 2004a,b, 2006] are based on combining the parametrization in equation (3.17) and the parametrization in equation (3.5). Even though the energy function in equation (3.5) in our model is different from the energy function in equation (3.17), it is our contribution to show in Appendix A that the global conditional probability functions involving the two energy functions and their combinations are indeed equivalent.

We now derive the subclass of the contrast sensitive Potts models reviewed in Section 1.2.2. We impose linear equality constraints

$$\boldsymbol{v}_{kk} = \boldsymbol{v}_{ll} \qquad \forall k, \forall l \tag{3.18}$$

$$\boldsymbol{v}_{kk'} = \boldsymbol{v}_{ll'} \qquad \forall k, \forall k', k \neq k', \forall l, \forall l', l \neq l' \tag{3.19}$$

on pairwise parameters $\boldsymbol{v}$ and obtain the subclass $\mathcal{P}_2$,

$$\mathcal{P}_2 = \{p \mid p \in \mathcal{P} \text{ s.t. } (3.18), (3.19)\} \tag{3.20}$$

of the contrast sensitive Potts models with pairwise label interaction that is label-pair-nonspecific and data-dependent attractive. The subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20) is parametrized by points forming a subspace in the pairwise parameter vector space of the subclass $\mathcal{P}_1$ of the data-dependent symmetric interaction models. $(K^2 - 2)D_v$ linear equality constraints in equation (3.18) and in equation (3.19) form a $2D_v$-dimensional subspace in the pairwise parameter vector space $\mathbb{R}^{K^2 D_v}$ of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1).

In table 3.2 we compare the subclasses of the models with the data-dependent label interaction in terms of the numbers of pairwise parameters. Top three lines in table 3.2 compare the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1) with the subclass $\mathcal{P}_1$ of the data-dependent symmetric interaction models in equation (3.16) and with the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20).

We have derived, as the subclasses of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1), the models with the data-dependent label interactions reviewed in Section 1.2.2. Our next step is to derive models with the data-independent label interactions.

3.2.2 Data-independent Pairwise Terms

We now derive models with data-independent label interactions reviewed in Section 1.2.2 as the subclasses of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1). Namely we derive the subclass of the asymmetric generalized Potts models, the subclass of the generalized Potts models and the subclass of the Potts models.

Table 3.2: *Subclasses of models: Pairwise parameters.* Number of pairwise parameters in K-class CRF model with D_v-dimensional pairwise feature vector. The table includes models with data-dependent label interaction, namely the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1), the subclass $\mathcal{P}_1$ of the data-dependent symmetric interaction models in equation (3.16) and the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20). The table also includes models with data-independent label interaction, namely the subclass $\mathcal{P}_3$ of the asymmetric generalized Potts models in equation (3.22), the subclass $\mathcal{P}_4$ of the generalized Potts models in equation (3.25) and the subclass $\mathcal{P}_5$ of the Potts models in equation (3.29). Eventually the table includes models with no label interaction, namely the subclass $\mathcal{P}_6$ of the local classifiers in equation (3.31).

Model class	Linear equality constraints	Parameter subspace dimension	Constraint equation
$\mathcal{P}$	0	$K^2 D_v$	–
$\mathcal{P}_1$	$\frac{1}{2}(K^2 - K)D_v$	$\frac{1}{2}(K^2 + K)D_v$	(3.15)
$\mathcal{P}_2$	$(K^2 - 2)D_v$	$2D_v$	(3.18),(3.19)
$\mathcal{P}_3$	$K^2(D_v - 1)$	K^2	(3.21)
$\mathcal{P}_4$	$\frac{1}{2}(K^2(2D_v - 1) - K)$	$\frac{1}{2}(K^2 + K)$	(3.23),(3.24)
$\mathcal{P}_5$	$K^2 D_v - 2$	2	(3.26),(3.27),3.28
$\mathcal{P}_6$	$K^2 D_v$	0	(3.30)

We derive the subclass of the asymmetric generalized Potts models reviewed in Section 1.2.2. We impose linear equality constraints

$$v_{kk';i} = 0 \qquad \forall k, \forall k', i = 1, \ldots, D_v - 1 \tag{3.21}$$

on pairwise parameters $\boldsymbol{v}$. The variable $v_{kk';i}$ is the i-th component of the vector $\boldsymbol{v}_{kk'}$ in equation (3.12). Here only the data independent component $v_{kk';0}$ is left. We obtain the subclass $\mathcal{P}_3$,

$$\mathcal{P}_3 = \{p \mid p \in \mathcal{P} \text{ s.t. } (3.21)\} \tag{3.22}$$

of the asymmetric generalized Potts models with pairwise label interaction that is data-independent, label-pair-specific and asymmetric. The subclass $\mathcal{P}_3$ of the asymmetric generalized Potts models in equation (3.22) is parametrized by points forming a subspace in the pairwise parameter vector space of the subclass $\mathcal{P}_1$ of the data-dependent symmetric interaction models. $K^2(D_v - 1)$ linear equality constraints in equation (3.21) form a

K^2-dimensional subspace in the pairwise parameter vector space $\mathbb{R}^{K^2 D_v}$ of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models.

We continue by deriving the subclass of the generalized Potts models reviewed in Section 1.2.2. We impose linear equality constraints

$$v_{kk';i} = 0 \qquad \forall k, \forall k', i = 1, \ldots, D_v - 1 \tag{3.23}$$

$$v_{kk';0} = v_{k'k;0} \qquad \forall k, \forall k', k' \neq k \tag{3.24}$$

on pairwise parameters $\boldsymbol{v}$ and obtain the subclass $\mathcal{P}_4$,

$$\mathcal{P}_4 = \{p \mid p \in \mathcal{P} \text{ s.t. } (3.23), (3.24)\} \tag{3.25}$$

of the generalized Potts models with pairwise label interaction that is data-independent, label-pair-specific and symmetric. The subclass $\mathcal{P}_4$ of the generalized Potts models in equation (3.25) is parametrized by points forming a subspace in the pairwise parameter vector space of the subclass $\mathcal{P}_1$ of the data-dependent symmetric interaction models. $\frac{1}{2}(K^2(2D_v - 1) - K)$ linear equality constraints in equation (3.23) and in equation (3.24) form a $\frac{1}{2}(K^2 + K)$-dimensional subspace in the pairwise parameter vector space $\mathbb{R}^{K^2 D_v}$ of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models.

Let us now derive the subclass of the Potts models reviewed again in Section 1.2.2. We impose linear equality constraints

$$v_{kk';i} = 0 \qquad \forall k, \forall k', i = 1, \ldots, D_v - 1 \tag{3.26}$$

$$v_{kk;0} = v_{ll;0} \qquad \forall k, \forall l \tag{3.27}$$

$$v_{kk';0} = v_{ll';0} \qquad \forall k, \forall k', k' \neq k, \forall l, \forall l', l' \neq l \tag{3.28}$$

on pairwise parameters $\boldsymbol{v}$ and obtain the subclass $\mathcal{P}_5$,

$$\mathcal{P}_5 = \{p \mid p \in \mathcal{P} \text{ s.t. } (3.26), (3.27), (3.28)\} \tag{3.29}$$

of the Potts models with pairwise label interaction that is data-independent, label-pair-nonspecific and attractive. The subclass $\mathcal{P}_5$ of the Potts models in equation (3.29) is parametrized by points forming a subspace in the pairwise parameter vector space of both the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models and the subclass $\mathcal{P}_4$ of the generalized Potts models. Specifically $K^2 D_v - 2$ linear equality constraints in equation (3.26), in equation (3.27) and in equation (3.28) form a 2-dimensional subspace in the pairwise parameter vector space $\mathbb{R}^{K^2 D_v}$ of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models.

We evaluate an example of the energy function in equation (3.7) and an example of the conditional probability mass function in equation (3.2) of the subclass $\mathcal{P}_5$ of the Potts models in equation (3.29). The example is analogous to the example 3.1. The example also involves two semantic variables and two observations. Both the two semantic variable $\mathcal{P}$ CRF in example 3.1 and the two semantic variable $\mathcal{P}_5$ CRF in the example 3.2 are recurrent themes that we repeatedly use in the rest of this work to illuminate the presented material.

Example 3.2 (Two variable $\boldsymbol{\mathcal{P}_5}$ CRF)
We adopt the image representation described in Section 2.1. For simplicity we reduce pixel color vectors $\boldsymbol{y}_i$ to intensity scalars y_i and let both the intensities y_i and the class labels x_i take values it the set $\mathcal{K}$ in equation (2.4). We set the functional form of the unary energy terms in equation (3.5) as

$$E_1(x_i \mid y_i) = 1 - \delta(y_i - x_i)$$

where we reduce the global dependence of the function on the data $\boldsymbol{y}$ to the local dependence on the data y_i. The above unary energy term does not include any model parameters $\boldsymbol{w}$. We set the functional form of the pairwise energy terms in equation (3.5) as

$$E_2(x_i, x_{i'}) = \beta(1 - \delta(x_i - x_{i'}))$$

where we drop the dependence of the function on the data $\boldsymbol{y}$ completely. The right hand side in the above equation is the Potts model and, if the number of states $K = 2$, then it is an equivalent representation of the Ising model. The model parameter vector $\boldsymbol{v}$ reduces to the scalar parameter β that is weighting the pairwise energy terms relative to the unary energy terms. We let the set $\mathcal{I}$ of pixel sites contain two sites 0 and 1. We consider the Potts CRF energy and let the intensities and the class labels both take values it the set $\mathcal{K} = (0, 1)$. Let the two pixel intensities $y_0 = 1$ and $y_1 = 0$ be given and let the associated pixel class labels be denoted by x_0 and x_1. In order to fully specify the energy function in equation (3.5), we set the scalar parameter $\beta = 0.9$ and we let the set $\mathcal{E}$ contain the pixel site pair $(0, 1)$. We can now evaluate the unary energy function terms

$$\begin{gathered} E_1(x_0 = 0 \mid y_0) = 1 \\ E_1(x_0 = 1 \mid y_0) = 0 \\ E_1(x_1 = 0 \mid y_1) = 0 \\ E_1(x_1 = 1 \mid y_1) = 1 \end{gathered}$$

and the pairwise energy function terms

$$\begin{aligned} E_2(x_0 = 0, x_1 = 0) &= 0 \\ E_2(x_0 = 1, x_1 = 0) &= 0.9 \\ E_2(x_0 = 0, x_1 = 1) &= 0.9 \\ E_2(x_0 = 1, x_1 = 1) &= 0 \end{aligned}$$

We add these values according to equation (3.5) and obtain a value of the overall energy function. Overall energy function values for four possible labeling of the CRF are listed in table 3.3. Energy values in table 3.3 allow us to evaluate the partition function in equation (3.3). Here the value of the partition function is $Z(\boldsymbol{y}) = 1.1974$. *Eventually the energy values and the partition function allow us to evaluate the probability mass in equation (3.2). Probability masses of the four possible labeling are listed in table 3.3.* $\ast$

Table 3.3: *Two variable $\mathcal{P}_5$ CRF.* Energy function values and probability masses of four possible labeling of CRF in example 3.2. The energy values are computed using equation (3.5) and the probability masses are computed using equation (3.2). The partition function involved in equation (3.2) is computed using equation (3.3). Here the value of the partition function is $Z(\boldsymbol{y}) = 1.1974$.

Labeling $\boldsymbol{x}$		Energy	Probability
x_0	x_1		mass
0	0	1.0	0.3072
1	0	0.9	0.3396
0	1	2.9	0.0460
1	1	1.0	0.3072

In table 3.2 we compare the subclasses of models with data-independent label interaction in terms of the numbers of pairwise parameters. Specifically table 3.2 compares the subclass $\mathcal{P}_3$ of the asymmetric generalized Potts models in equation (3.22), the subclass $\mathcal{P}_4$ of the generalized Potts models in equation (3.25) and the subclass $\mathcal{P}_5$ of the Potts models in equation (3.29).

We have derived, as the subclasses of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1), the models with the data-independent label interactions reviewed in Section 1.2.2. Our next step is to derive models without any label interaction.

3.2.3 Local Classifiers

At last we derive local independent classifiers as a subclass of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1). We impose linear equality constraints

$$\boldsymbol{v}_{kk'} = \boldsymbol{0} \qquad \forall k, \forall k' \tag{3.30}$$

on pairwise parameters $\boldsymbol{v}$ and obtain the subclass $\mathcal{P}_6$,

$$\mathcal{P}_6 = \{p \mid p \in \mathcal{P} \text{ s.t. } (3.30)\} \tag{3.31}$$

of the local classifiers with no pairwise label interaction that take the form of logistic regression model for classification. The subclass $\mathcal{P}_6$ of the local classifiers in equation (3.31) is parametrized by points forming a trivial subspace in the pairwise parameter vector spaces of all the previous subclasses. $K^2 D_v$ linear equality constraints in equation (3.30) form a 0-dimensional subspace in the pairwise parameter vector space $\mathbb{R}^{K^2 D_v}$ of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models.

In table 3.2 we compare the subclass $\mathcal{P}_6$ of the local classifiers in equation (3.31) with no label interaction with the subclasses of models with label interaction in terms of the numbers of pairwise parameters.

We derived, as the subclasses of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1), the models with no label interactions. In the following we give illustrative examples of the model subclasses.

3.2.4 Model Subclass Examples

In previous sections we have derived the models, that we reviewed in Section 1.2.2, as the subclasses of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1). We have shown that models from the respective subclasses can be seen as being parametrized by points forming subspaces in the parameter vector space of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models. In table 3.2 we compare the dimensions of the respective subclass subspaces. In this section we illuminate the subclass parametrization using the two semantic variable $\mathcal{P}$ CRF from example 3.1.

Example 3.3 (Two variable $\boldsymbol{\mathcal{P}}$ CRF: Subclasses)
We give model subclass examples of the two variable $\mathcal{P}$ CRF described in example 3.1. The parameter vector $\boldsymbol{u}^\mathsf{T} = [\boldsymbol{w}^\mathsf{T}, \boldsymbol{v}^\mathsf{T}]$, where

$$\boldsymbol{u}^\mathsf{T} = [[w_{0;0}, w_{0;1}, w_{0;0}, w_{0;1}], [v_{00;0}, v_{00;1}, v_{10;0}, v_{10;1}, v_{01;0}, v_{01;1}, v_{11;0}, v_{11;1}]]$$

For illustration purposes we make use of a particular parameter arrangement in the first row from the top in table 3.4. To ease the notation we rename the parameters from the first row and include them in the second row from the top in table 3.4 using a simple running index. The arrangement means that the parameter vector

$$\boldsymbol{u}^\mathsf{T} = [u_0, u_2, u_1, u_3, u_4, u_8, u_5, u_9, u_6, u_{10}, u_7, u_{11}]$$

Table 3.4: *Two variable $\mathcal{P}$ CRF: Subclass parameters.* (a)(b) Unary and (c)(d) pairwise parameters of the model subclasses.

	$\begin{bmatrix}w_{0;0} & w_{1;0}\end{bmatrix}$	$\begin{bmatrix}w_{0;1} & w_{1;1}\end{bmatrix}$	$\begin{bmatrix}v_{00;0} & v_{01;0}\\ v_{10;0} & v_{11;0}\end{bmatrix}$	$\begin{bmatrix}v_{00;1} & v_{01;1}\\ v_{10;1} & v_{11;1}\end{bmatrix}$
$(\mathcal{P})$	$\begin{bmatrix}u_0 & u_1\end{bmatrix}$	$\begin{bmatrix}u_2 & u_3\end{bmatrix}$	$\begin{bmatrix}u_4 & u_6\\ u_5 & u_7\end{bmatrix}$	$\begin{bmatrix}u_8 & u_{10}\\ u_9 & u_{11}\end{bmatrix}$
$(\mathcal{P}_1)$	$\begin{bmatrix}u_0 & u_1\end{bmatrix}$	$\begin{bmatrix}u_2 & u_3\end{bmatrix}$	$\begin{bmatrix}u_4 & u_5\\ u_5 & u_6\end{bmatrix}$	$\begin{bmatrix}u_7 & u_8\\ u_8 & u_9\end{bmatrix}$
$(\mathcal{P}_2)$	$\begin{bmatrix}u_0 & u_1\end{bmatrix}$	$\begin{bmatrix}u_2 & u_3\end{bmatrix}$	$\begin{bmatrix}u_4 & u_5\\ u_5 & u_4\end{bmatrix}$	$\begin{bmatrix}u_6 & u_7\\ u_7 & u_6\end{bmatrix}$
$(\mathcal{P}_3)$	$\begin{bmatrix}u_0 & u_1\end{bmatrix}$	$\begin{bmatrix}u_2 & u_3\end{bmatrix}$	$\begin{bmatrix}u_4 & u_6\\ u_5 & u_7\end{bmatrix}$	$\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$
$(\mathcal{P}_4)$	$\begin{bmatrix}u_0 & u_1\end{bmatrix}$	$\begin{bmatrix}u_2 & u_3\end{bmatrix}$	$\begin{bmatrix}u_4 & u_5\\ u_5 & u_6\end{bmatrix}$	$\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$
$(\mathcal{P}_5)$	$\begin{bmatrix}u_0 & u_1\end{bmatrix}$	$\begin{bmatrix}u_2 & u_3\end{bmatrix}$	$\begin{bmatrix}u_4 & u_5\\ u_5 & u_4\end{bmatrix}$	$\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$
$(\mathcal{P}_6)$	$\begin{bmatrix}u_0 & u_1\end{bmatrix}$	$\begin{bmatrix}u_2 & u_3\end{bmatrix}$	$\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$
	(a)	(b)	(c)	(d)

In table 3.5 we summarize the subclasses of the models in terms of the numbers of pairwise parameters similar to table 3.2. In table 3.5 we include models with data-dependent label interaction, namely the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1), the subclass $\mathcal{P}_1$ of the data-dependent symmetric interaction models in equation (3.16) *and the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation* (3.20). *The table also includes models with data-independent label interaction, namely the*

Table 3.5: *Two variable $\mathcal{P}$ CRF: Subclass pairwise parameter numbers.* Numbers of pairwise parameters in 2-class CRF model with 2-dimensional pairwise feature vector.

Model class	Parameters space dimension	Linear equality constraints	Parameter subspace dimension
$\mathcal{P}$	8	0	8
$\mathcal{P}_1$	8	2	6
$\mathcal{P}_2$	8	4	4
$\mathcal{P}_3$	8	4	4
$\mathcal{P}_4$	8	5	3
$\mathcal{P}_5$	8	6	2
$\mathcal{P}_6$	8	8	0

subclass $\mathcal{P}_3$ of the asymmetric generalized Potts models in equation (3.22)*, the subclass $\mathcal{P}_4$ of the generalized Potts models in equation* (3.25) *and the subclass $\mathcal{P}_5$ of the Potts models in equation* (3.29)*. Eventually the table includes models with no label interaction, namely the subclass $\mathcal{P}_6$ of the local classifiers in equation* (3.31)*.* $*$

We gave illustrative examples of the model subclasses that were meant to illuminate the subclass parametrization. The contribution of Section 3.2 is the derivation of the models, that we reviewed in Section 1.2.2, as the subclasses of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1). We have shown that models from the respective subclasses can be seen as being parametrized by points forming subspaces in the parameter vector space of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models. We will make use of this perspective to arrive at an extension of the learning of the parameters of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models that confines the learning to the learning of the parameter of the subclasses derived in this section.

Chapter 4

A Tractable Learning for a Class of Global Discriminative Models

This chapter deals with the core computational problem of how should the trainable function for automatic image interpretation, described in Chapter 2, learn the relation between images and semantic descriptions from examples. More specifically we describe how should the function with the use of the training examples learn parameters of an internal global statistical model of the class of models proposed in Chapter 3. In Section 4.1 we translate the training of the function for automatic image interpretation to the problem of finding the unique minimum of a strongly convex objective function, where the minimum is related to a mode of the likelihood function in the problem (2.16). In Section 4.2 we explain why in general training the function for automatic image interpretation in this manner is computationally intractable. In Section 4.3 we formulate the training of the function for automatic image interpretation as a consistent tractable strongly convex approximation of the intractable learning problem in Section 4.1. In Section 4.4 we propose to train the function for automatic image interpretation using the rapidly convergent algorithms for convex optimization. Eventually in Section 4.6 we propose a way to compare performance of the trainable function for automatic image interpretation, based on the class of the global statistical models described in Chapter 3, with the functions, based on the subclasses of the models described in Section 3.2. Specifically we describe how learning of the parameters of the global statistical models of the class described in Chapter 3, can in a simple way be confined to learning the parameters of the models of the subclasses described in Section 3.2.

4.1 An Exact Parameter Learning Principle

This chapter describes how in principle the trainable function f in equation (2.9) for automatic image interpretation learns from the training set $\mathcal{D}_{\mathcal{M}} = \{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m\in\mathcal{M}}$ in equation (2.7) the relation between images $\boldsymbol{y}^n$ and the semantic descriptions $\boldsymbol{x}^n$ from the test set $\mathcal{D}_{\mathcal{N}} = \{\boldsymbol{x}^n, \boldsymbol{y}^n\}_{n\in\mathcal{N}}$ in equation (2.8) without having access to the test set during the training. The function f achieves this by according to the maximum likelihood principle automatically learning parameters $\boldsymbol{u}$ in equation (3.6) of an internal global conditional probability mass function p in equation (3.2) of the class $\mathcal{P}$ of models in equation (3.1) that maps images $\boldsymbol{y}^m$ and the semantic descriptions $\boldsymbol{x}^m$ from the training set on a high probability value. The computational challenge of training the function f translates in current section to the problem of finding unique minimum of a strongly convex function that results from combining the convex negative log likelihood in the problem (2.16) with the strongly convex negative log parameter prior distribution.

4.1.1 An Exact Convex Learning Problem

In this section the problem of parameter learning takes the form of an unconstrained convex optimization problem. The input to the learning problem is the training set $\mathcal{D}_{\mathcal{M}} = \{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m\in\mathcal{M}}$ in equation (2.7) that we treat as as a set of samples of the random vector $\underline{\boldsymbol{y}}$ that takes values in the continuous space $\mathbb{R}^{IC}$ of image data vectors and of samples of a random vector $\underline{\boldsymbol{x}}$ that takes values in the discrete space $\mathcal{K}^I$ of image label configurations. We treat these samples as having been sampled independently from an identical distribution and that the conditional probability mass $p(\{\boldsymbol{x}^m\}_{m\in\mathcal{M}} \mid \{\boldsymbol{y}^m\}_{m\in\mathcal{M}}, \boldsymbol{u})$ can thus be factorized as the product $\prod_{m\in\mathcal{M}} p(\boldsymbol{x}^m \mid \boldsymbol{y}^m, \boldsymbol{u})$.

The maximum likelihood (ML) parameter vector estimate $\boldsymbol{u}^{\mathrm{ML}}$,

$$\boldsymbol{u}^{\mathrm{ML}} \in \arg\max_{\boldsymbol{u}} L(\boldsymbol{u}) \tag{4.1}$$

maximizes the likelihood $L : \mathbb{R}^{D_u} \to (0, 1)$,

$$L(\boldsymbol{u}) = \prod_{m\in\mathcal{M}} p(\boldsymbol{x}^m \mid \boldsymbol{y}^m, \boldsymbol{u}) \tag{4.2}$$

of the parameter vector $\boldsymbol{u}$, given a fixed training set $\{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m\in\mathcal{M}}$. The likelihood L is a function that maps a parameter vector $\boldsymbol{u} \in \mathbb{R}^{D_u}$ on a real number in the open interval from 0 to 1. The interval being open is a consequence of the class of models described in Chapter 3 that we assume to adopt. The class of conditional probability mass functions described in

Chapter 3 will always place a strictly positive mass on each of the conceivable configurations. As a result there will be no label configuration that is assigned the value zero and there will be no label configuration that is assigned the value one. Equivalently the parameter estimate $\boldsymbol{u}^{\mathrm{ML}}$,

$$\boldsymbol{u}^{\mathrm{ML}} \in \arg\min_{\boldsymbol{u}} l(\boldsymbol{u}) \tag{4.3}$$

minimizes the negative log likelihood $l : \mathbb{R}^{D_u} \to \mathbb{R}_{++}$,

$$l(\boldsymbol{u}) = \sum_{m \in \mathcal{M}} - \log p(\boldsymbol{x}^m \mid \boldsymbol{y}^m, \boldsymbol{u}) \tag{4.4}$$

of the parameter vector $\boldsymbol{u}$, given a fixed training set $\{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m \in \mathcal{M}}$. Negative log likelihood l is a function that maps a parameter vector $\boldsymbol{u}$ on a strictly positive real number. It can be shown that the negative log likelihood function l in equation (4.4) is a convex function [Winkler, 2006, Wainwright and Jordan, 2008].

We evaluate an example of the negative log likelihood function in equation (4.4) and we plot the graph of this function to illustrate its convexity.

Example 4.1 (Two variable $\mathcal{P}_5$ CRF: Likelihood)
We adopt the two variable $\mathcal{P}_5$ CRF from example 3.2 and consider the negative log likelihood $l(\beta)$ of the scalar parameter β given a labeled training image $\{\boldsymbol{x}^1, \boldsymbol{y}^1\}$. Let the image $\boldsymbol{y}^1$ be given in example 3.2 and let the labeling $\boldsymbol{x}^1 = [1, 0]^\mathsf{T}$. We use probability mass from table 3.3 to evaluate the likelihood $L(0.9)$,

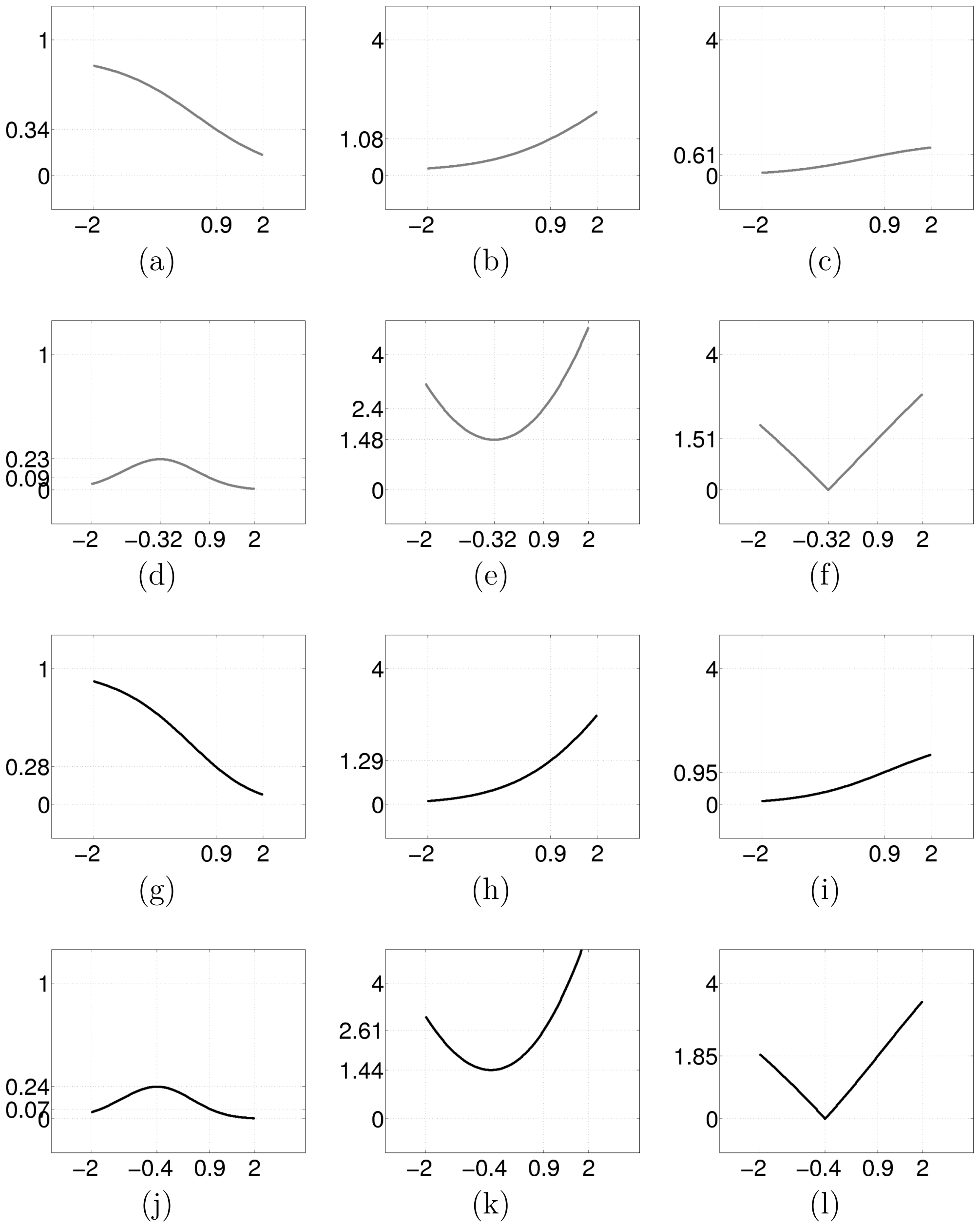
$$L(0.9) = 0.3396$$

in equation (4.2) at the parameter value 0.9. Afterwards we use the likelihood value to evaluate the negative log likelihood $l(0.9)$,

$$l(0.9) = 1.0800$$

in equation (4.4) at the parameter value 0.9. Values of the above two functions for parameter value β being varied between -2 and 2 are respectively shown in figure 4.1(a) and in figure 4.1(b). Indeed in figure 4.1(b) we observe a graph of a convex function. $\ast$

We evaluate another example of the negative log likelihood in equation (4.4) and again we plot the graph of this function to illustrate its convexity.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.1: *Two variable $\mathcal{P}_5$ CRF: Learning.* Parameter β from example 4.1 and its (a) likelihood $L(\beta)$, (b) negative log likelihood $l(\beta)$ and (c) gradient magnitude $||\nabla l(\beta)||_2$. (d) Posterior $L_{\mathrm{MAP}}(\beta)$, (e) negative log posterior $l_{\mathrm{MAP}}(\beta)$ and (f) gradient magnitude $||\nabla l_{\mathrm{MAP}}(\beta)||_2$. Later in section we describe (g) pseudolikelihood $\hat{L}(\beta)$, (h) negative log pseudolikelihood $\hat{l}(\beta)$ and (i) gradient magnitude $||\nabla\hat{l}(\beta)||_2$. (j) Pseudoposterior $\hat{L}_{\mathrm{MAP}}(\beta)$, (k) negative log pseudoposterior $\hat{l}_{\mathrm{MAP}}(\beta)$ and (l) gradient magnitude $||\nabla\hat{l}_{\mathrm{MAP}}(\beta)||_2$.

Example 4.2 (Two variable $\mathcal{P}$ CRF: Likelihood)
We adopt the two variable $\mathcal{P}$ CRF from example 3.1 and consider the negative log likelihood $l(\boldsymbol{u})$ of the parameter vector $\boldsymbol{u}$, specified in equation (3.14) in example 3.1, given a labeled training image $\{\boldsymbol{x}^1, \boldsymbol{y}^1\}$. Let the image $\boldsymbol{y}^1$ be given in example 3.1 and let the labeling $\boldsymbol{x}^1 = [1, 0]^\intercal$. We use a probability mass from table 3.1 to evaluate the likelihood $L(\boldsymbol{u})$,

$$L(\boldsymbol{u}) = 0.7112$$

in equation (4.2) of the parameters $\boldsymbol{u}$. Afterwards we use the likelihood value to evaluate the negative log likelihood $l(\boldsymbol{u})$,

$$l(\boldsymbol{u}) = 0.3408$$

in equation (4.4) of the parameters $\boldsymbol{u}$. For illustration we vary the value of the first component $w_{0;0}$ and the value of the fifth component $v_{00;0}$ of the parameter vector $\boldsymbol{u}$,

$$\boldsymbol{u} = [[w_{0;0}, 1], [0.5, -1], [v_{00;0}, 1], [0.5, -1], [0.5, -1], [-0.5, 1]]^\intercal$$

between -1 and 1. Level curves of the above two functions for these points are respectively shown in figure 4.2(a) and in figure 4.2(b). $*$

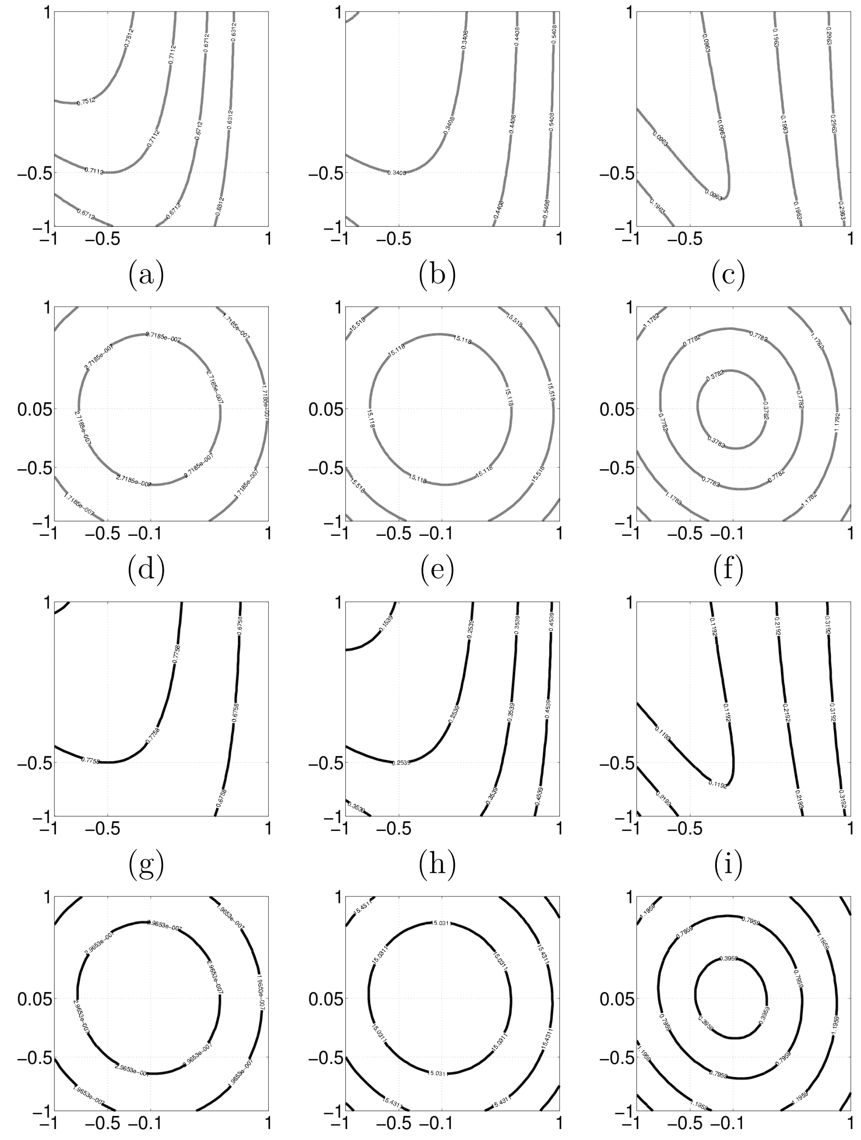
Throughout this chapter figure 4.1 and figure 4.2 are meant as extensions of our examples that visually illustrate the properties of the involved functions. The first column in figure 4.1 and figure 4.2 shows graphs of log-concave functions, the second column in figure 4.1 and figure 4.2 shows graphs of convex functions and the last column in figure 4.1 and figure 4.2 shows the first partial derivatives of the convex functions in the second column in figure 4.1 and figure 4.2.

We have described the desired model parameter vector as a solution of the unconstrained convex optimization problem (4.3). In the following section we turn the problem (4.3) into an unconstrained strongly convex optimization problem by combining the convex negative log likelihood in equation (4.4) with a strongly convex negative log parameter prior distribution.

4.1.2 An Exact Strongly Convex Learning Problem

In this section the problem of parameter learning takes the form of an unconstrained strongly convex optimization problem. The maximum a posteriori (MAP) parameter vector estimate $\boldsymbol{u}^{\mathrm{MAP}}$,

$$\boldsymbol{u}^{\mathrm{MAP}} \in \arg\max_{\boldsymbol{u}} \prod_{m \in \mathcal{M}} p(\boldsymbol{u} \mid \boldsymbol{x}^m, \boldsymbol{y}^m)$$



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.2: *Two variable $\mathcal{P}$ CRF: Learning.* Parameter vector $\boldsymbol{u}$ that is confined to its first component $w_{0;0}$ (horizontal axis) and its fifth component $v_{00;0}$ (vertical axis) that are varied between -1 and 1, and the level curves of its (a) likelihood $L(\boldsymbol{u})$, (b) negative log likelihood $l(\boldsymbol{u})$ and (c) gradient magnitude $||\nabla l(\boldsymbol{u})||_2$. (d) Posterior $L_{\mathrm{MAP}}(\boldsymbol{u})$, (e) negative log posterior $l_{\mathrm{MAP}}(\boldsymbol{u})$ and (f) gradient magnitude $||\nabla l_{\mathrm{MAP}}(\boldsymbol{u})||_2$. (g) Pseudolikelihood $\hat{L}(\boldsymbol{u})$, (h) negative log pseudolikelihood $\hat{l}(\boldsymbol{u})$ and (i) gradient magnitude $||\nabla \hat{l}(\boldsymbol{u})||_2$. (j) Pseudoposterior $\hat{L}_{\mathrm{MAP}}(\boldsymbol{u})$, (k) negative log pseudoposterior $\hat{l}_{\mathrm{MAP}}(\boldsymbol{u})$ and (l) gradient magnitude $||\nabla \hat{l}_{\mathrm{MAP}}(\boldsymbol{u})||_2$.

maximizes the product of posterior probability densities $p : \mathbb{R}^{D_u} \times \mathcal{K}^I \times \mathbb{R}^{IC} \to \mathbb{R}_{++}$, where $\int_{\mathbb{R}^{D_u}} p(\boldsymbol{u} \mid \boldsymbol{x}^m, \boldsymbol{y}^m) d\boldsymbol{u} = 1$, of a parameter vector $\boldsymbol{u}$, given a fixed training set $\{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m \in \mathcal{M}}$. Given a fixed training set $\{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m \in \mathcal{M}}$ and given a prior probability density function $p(\boldsymbol{u})$, the above product of posterior densities,

$$\prod_{m \in \mathcal{M}} p(\boldsymbol{u} \mid \boldsymbol{x}^m, \boldsymbol{y}^m) \propto \prod_{m \in \mathcal{M}} \Bigg(p(\boldsymbol{x}^m \mid \boldsymbol{y}^m, \boldsymbol{u}) p(\boldsymbol{u}) \Bigg) \tag{4.5}$$

can be written as being proportional to the right hand side of the above equation. Equivalently the parameter estimate $\boldsymbol{u}^{\text{MAP}}$,

$$\boldsymbol{u}^{\text{MAP}} \in \arg\max_{\boldsymbol{u}} L_{\text{MAP}}(\boldsymbol{u}) \tag{4.6}$$

maximizes the function $L_{\text{MAP}} : \mathbb{R}^{D_u} \to \mathbb{R}_{++}$,

$$L_{\text{MAP}}(\boldsymbol{u}) = \prod_{m \in \mathcal{M}} \Bigg(p(\boldsymbol{x}^m \mid \boldsymbol{y}^m, \boldsymbol{u}) p(\boldsymbol{u}) \Bigg) \tag{4.7}$$

of the parameter vector $\boldsymbol{u}$, given a fixed training set $\{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m \in \mathcal{M}}$. Function value $L_{\text{MAP}}(\boldsymbol{u})$ is up to a constant factor equal to the product of posterior densities on the left hand side in equation (4.5). With a slight abuse of terminology we will refer to the function $L_{\text{MAP}}(\boldsymbol{u})$ as to the posterior of the parameter vector $\boldsymbol{u}$. Equivalently the parameter estimate $\boldsymbol{u}^{\text{MAP}}$,

$$\boldsymbol{u}^{\text{MAP}} \in \arg\min_{\boldsymbol{u}} l_{\text{MAP}}(\boldsymbol{u}) \tag{4.8}$$

minimizes the negative log posterior $l_{\text{MAP}} : \mathbb{R}^{D_u} \to \mathbb{R}$,

$$l_{\text{MAP}}(\boldsymbol{u}) = \Bigg(l(\boldsymbol{u}) - |\mathcal{M}| \log p(\boldsymbol{u}) \Bigg) \tag{4.9}$$

of the parameter vector $\boldsymbol{u}$, given a fixed training set $\{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m \in \mathcal{M}}$. The scalar $|\mathcal{M}|$ denotes the number of labeled images in the training set. We adopt a prior probability density $p(\boldsymbol{u} \mid \tau)$ forming the normal (or Gaussian) distribution $\mathcal{N}(\boldsymbol{u} \mid \boldsymbol{0}, \tau^2 \boldsymbol{I})$,

$$p(\boldsymbol{u} \mid \tau) = \mathcal{N}(\boldsymbol{u} \mid \boldsymbol{0}, \tau^2 \boldsymbol{I}) = \frac{1}{(2\pi)^{D_u/2} \mid \tau^2 \boldsymbol{I} \mid^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \boldsymbol{u}^\intercal (\tau^2 \boldsymbol{I})^{-1} \boldsymbol{u} \right) \tag{4.10}$$

with the mean vector $\boldsymbol{0}$ and the variance τ^2. To show that the negative log posterior function l_{MAP} in equation (4.9), where the prior probability density function $p(\boldsymbol{u})$ takes the form in equation (4.10), is a convex function, we need to show that the negative log prior probability density in equation (4.8) is convex. Computing the Hessian yields for the finite variance τ^2 the positive definite matrix $(\tau^2 \boldsymbol{I})^{-1}$ that not only guarantees strict convexity but also that the function is strongly convex [Boyd and Vandenberghe, 2004]. Summing the convex likelihood function with the strongly convex negative log prior probability density function yields the strongly convex negative log posterior in equation (4.9). Let us note that the problem (4.8) is equivalent to the problem (4.3) if prior information is unavailable and an uninformative prior probability density function $p(\boldsymbol{u})$ is employed. In case of the Gaussian prior $p(\boldsymbol{u} \mid \tau)$ this would mean that the standard deviation $\tau = \infty$.

We evaluate an example of the negative log posterior density function in equation (4.9). We plot a graph of the negative log posterior density function from the example to illustrate the retained convexity and the shift in location of the optimum as compared to the analogous negative log likelihood function.

Example 4.3 (Two variable $\mathcal{P}_5$ CRF: Posterior)
We adopt example 4.1 and employ a prior probability density function over the parameter β in the form of the Gaussian density function $p(\beta \mid \tau)$, where we set the mean to 0 and the variance $\tau^2 = 1$. We use a likelihood value from example 4.1 to evaluate the posterior $L_{\text{MAP}}(0.9)$,

$$L_{\text{MAP}}(0.9) = 0.0904$$

in equation (4.7) of the parameter value 0.9. Afterwards we use the posterior value to evaluate the negative log posterior density function $l_{\text{MAP}}(0.9)$,

$$l_{\text{MAP}}(0.9) = 2.4039$$

in equation (4.9) at the parameter value 0.9. Values of the above two functions for parameter value β being varied between -2 and 2 are respectively shown in figure 4.1(d) and in figure 4.1(e). We observe that the negative log posterior density function in figure 4.1(e) retains convexity and shifts the location of the optimum as compared to the analogous negative log likelihood function in figure 4.1(b). $*$

We evaluate another example of the negative log posterior density in equation (4.9). Again we plot a graph of the negative log posterior density function from the example to illustrate the retained convexity and the shift in location of the optimum as compared to the analogous negative log likelihood function.

Example 4.4 (Two variable $\mathcal{P}$ CRF: Posterior)
We continue example 4.2 by employing a prior probability density function over the parameter vector $\boldsymbol{u}$ in the form of the Gaussian probability density function in equation (4.10), where we set the variance $\tau^2 = 1$. We first use the likelihood value from example 4.2 to evaluate the posterior $L_{\mathrm{MAP}}(\boldsymbol{u})$,

$$L_{\mathrm{MAP}}(\boldsymbol{u}) \approx 0$$

in equation (4.7) of the parameter vector $\boldsymbol{u}$. This is a very small number. Afterwards we use the posterior value to evaluate the negative log posterior $l_{\mathrm{MAP}}(\boldsymbol{u})$,

$$l_{\mathrm{MAP}}(\boldsymbol{u}) = 15.1180$$

in equation (4.9) of the parameter vector $\boldsymbol{u}$. We vary the value of the first component $w_{0;0}$ and the value of the fifth component $v_{00;0}$ of the parameter vector $\boldsymbol{u}$ between -1 and 1. Level curves of the above two functions for these points are respectively shown in figure 4.2(d) and in figure 4.2(e). We observe that the negative log posterior density function in figure 4.2(e) retains convexity and shifts the location of the optimum as compared to the analogous negative log likelihood function in figure 4.2(b). $*$

We have described the desired model parameter vector as a solution of the unconstrained strongly convex optimization problem (4.8). And as we explain in Chapter 1 an iterative descent method is theoretically guaranteed to solve a strongly convex optimization problem up to a precision in a finite number of iterations. In the following we show however that performing one such iteration is in general computationally intractable.

4.2 Intractable Exact Parameter Learning

In this section we explain what it means to solve the unconstrained strongly convex optimization problem (4.8) and why in general training the function f in equation (2.9) in this manner is computationally *intractable*. The computational challenge of training the function f translates to the problem of iterative evaluation of the likelihood and of marginal probabilities in an iterative algorithm for convex optimization. Since convex optimization algorithms including gradient descent methods, conjugate gradient methods and quasi-Newton methods are detailed in many textbooks we omit the description of the algorithms themselves. See for instance the gradient descent method in Algorithm 9.3 in [Boyd and Vandenberghe, 2004].

4.2.1 Solving the Exact Convex Learning Problem

To minimize the negative log likelihood $l(\boldsymbol{u})$ in equation (4.3) with a gradient descent method, we need to be able to iteratively evaluate the negative log likelihood $l(\boldsymbol{u})$ and the gradient $\nabla l(\boldsymbol{u})$, or, more specifically, its components, the partial derivatives. While in principle it is possible to evaluate gradient vector components numerically by evaluating the function $l(\boldsymbol{u})$ in the neighborhood of the vector $\boldsymbol{u}$, it is often computationally advantageous to evaluate explicit form of the partial derivatives. We begin by considering the partial derivatives of the the likelihood for the pairwise homogeneous isotropic conditional random field in equation (3.5), where we note that this model can also be shown to be a member of the exponential family. Partial derivatives with respect to the unary parameters $w_{k;j}$ are given by

$$\frac{\partial l(\boldsymbol{u})}{\partial w_{k;j}} = \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{I}^m} \left(\frac{\partial E_1(x_i^m, \boldsymbol{w})}{\partial w_{k;j}} - \left\langle \frac{\partial E_1(x_i'^m, \boldsymbol{w})}{\partial w_{k;j}} \right\rangle_{m,u} \right) \tag{4.11}$$

and partial derivatives with respect to the pairwise parameters $v_{kk';j}$ are given by

$$\frac{\partial l(\boldsymbol{u})}{\partial v_{kk';j}} = \sum_{m \in \mathcal{M}} \sum_{(i,i') \in \mathcal{E}^m} \left(\frac{\partial E_2(x_i^m, x_{i'}^m, \boldsymbol{v})}{\partial v_{kk';j}} - \left\langle \frac{\partial E_2(x_i'^m, x_{i'}'^m, \boldsymbol{v})}{\partial v_{kk';j}} \right\rangle_{m,u} \right) \tag{4.12}$$

For compactness in the above two equations we drop the dependence on image data $\boldsymbol{y}^m$ in both the unary and the pairwise terms. The terms involving the angle brackets $\langle\rangle_{m,u}$,

$$\langle g(x_i'^m) \rangle_{m,u} = \sum_{\boldsymbol{x}'^m \in \mathcal{K}^{I^m}} g(x_i'^m) p(\boldsymbol{x}'^m \mid \boldsymbol{y}^m, \boldsymbol{u})$$

denote expectations with the global conditional probability mass function $p(\boldsymbol{x}'^m \mid \boldsymbol{y}^m, \boldsymbol{u})$. Let us note that in the above equation we vary the value $k \in \mathcal{K}$ of the variable $x_i'^m \in \mathcal{K}$ that is indexed with the fixed site index i and with the fixed training sample index m. The terms involving the expectation result from differentiating the log partition function. Proofs of the likelihood gradient form for the general Markov random field from the exponential family without conditioning are given in [Winkler, 2006, Wainwright and Jordan, 2008].

We evaluate an example of the partial derivative in equation (4.12) with respect to the pairwise parameter of the two semantic variable $\mathcal{P}_5$ CRF. The gradient function is the central component of a learning algorithm. We recommend the interested reader to both verify the form of the gradient and to verify its value at the location suggested in the following example using a pocket calculator.

Example 4.5 (Two variable $\mathcal{P}_5$ CRF: Likelihood gradient)
We extend example 4.1. We use probability masses in table 3.3 to evaluate the gradient $\nabla l(0.9)$*,*

$$\nabla l(0.9) = 1 - \delta(x_0^1 - x_1^1) - \langle 1 - \delta(x_0'^1 - x_1'^1)\rangle_{1,0.9} = 0.6144$$

in equation (4.12) of the negative log likelihood at the pairwise parameter value 0.9*. We leave our pocket calculator aside and identify the computed value in figure 4.1(c), where we show the gradient magnitude* $||\nabla l(\beta)||_2$ *for values of the parameter* β *varied between* -2 *and* 2 *in figure 4.1(c).* $*$

Partial derivatives in equation (4.11) and in equation (4.12) are partial derivative equations for the pairwise homogeneous isotropic conditional random field in equation (3.5). To minimize the negative log likelihood $l(\boldsymbol{u})$ in equation (4.3) of parameters $\boldsymbol{u}$ of a CRF of the class $\mathcal{P}$ of models in equation (3.1) with a gradient descent method, we need to specialize equation (4.11) and equation (4.12) with respect to equation (3.7). The partial derivatives with respect to the unary parameters $w_{k;j}$ are given by

$$\frac{\partial l(\boldsymbol{u})}{\partial w_{k;j}} = \sum_{m\in\mathcal{M}} \sum_{i\in\mathcal{I}^m} \Big(\delta(x_i^m - k) - p(x_i^m = k \mid \boldsymbol{y}^m, \boldsymbol{u})\Big) h_{i;j}(\boldsymbol{y}^m) \tag{4.13}$$

and the partial derivatives with respect to the pairwise parameters $v_{kk';j}$ are given by

$$\begin{aligned}\frac{\partial l(\boldsymbol{u})}{\partial v_{kk';j}} = \sum_{m\in\mathcal{M}} \sum_{(i,i')\in\mathcal{E}^m} &\Big(\delta(x_i^m - k)\delta(x_{i'}^m - k') \\ &- p(x_i^m = k, x_{i'}^m = k' \mid \boldsymbol{y}^m, \boldsymbol{u})\Big)\mu_{ii';j}(\boldsymbol{y})\end{aligned} \tag{4.14}$$

To show that equation (4.13) is the correct specialization of equation (4.11), we need the expectation $\langle \delta(x_i'^m - k)\rangle_{m,u}$. It is given by

$$\langle \delta(x_i'^m - k)\rangle_{m,u} = \sum_{\boldsymbol{x}'^m\in\mathcal{K}^I | x_i'^m = k} p(\boldsymbol{x}'^m \mid \boldsymbol{y}^m, \boldsymbol{u})$$

which yields the unary marginal conditional probability mass $p(x_i^m = k \mid \boldsymbol{y}^m, \boldsymbol{u})$ in equation (4.13). To further show that equation (4.14) is the correct specialization of equation (4.12), we need the expectation $\langle \delta(x_i'^m - k)\delta(x_{i'}'^m - k')\rangle_{m,u}$. It is given by

$$\langle \delta(x_i'^m - k)\delta(x_{i'}'^m - k')\rangle_{m,u} = \sum_{\boldsymbol{x}'^m\in\mathcal{K}^I | x_i'^m = k, x_{i'}'^m = k'} p(\boldsymbol{x}'^m \mid \boldsymbol{y}^m, \boldsymbol{u})$$

which yields the pairwise marginal conditional probability mass $p(x_i^m = k, x_{i'}^m = k' \mid \boldsymbol{y}^m, \boldsymbol{u})$ in equation (4.14).

We evaluate an example of the partial derivative in equation (4.13) and of the partial derivative in equation (4.14). Again we recommend the interested reader to both verify the form of the gradient and to verify its value at the location suggested in the following example using a pocket calculator.

Example 4.6 (Two variable $\mathcal{P}$ CRF: Likelihood gradient)
Let us extend example 4.2 and evaluate the gradient $\nabla l(\boldsymbol{u})$ of the negative log likelihood at the point $\boldsymbol{u}$. It is a vector with 12 components. We use probability masses in table 3.1 to evaluate the first component $\nabla_{w_{0;0}} l(\boldsymbol{u})$,

$$\frac{\partial l(\boldsymbol{u})}{\partial w_{0;0}} = \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} p(x_0^1 = 0 \mid \boldsymbol{y}^1, \boldsymbol{u}) \\ p(x_1^1 = 0 \mid \boldsymbol{y}^1, \boldsymbol{u}) \end{bmatrix} \right)^{\mathsf{T}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0$$

of the gradient in equation (4.13) at the parameter vector $\boldsymbol{u}$. We use probability masses in table 3.1 to evaluate the fifth component $\nabla_{v_{00;0}} l(\boldsymbol{u})^{\mathsf{T}}$,

$$\frac{\partial l(\boldsymbol{u})}{\partial v_{00;0}} = \left(0 - p(x_0^1 = 0, x_1^1 = 0 \mid \boldsymbol{y}^1, \boldsymbol{u})\right) 1 = -0.0963$$

of the gradient in equation (4.14) at the parameter vector $\boldsymbol{u}$. As in example 4.2 we vary the value of the first component $w_{0;0}$ and the value of the fifth component $v_{00;0}$ of the parameter vector $\boldsymbol{u}$ between -1 and 1. Level curves of the magnitude $||[\nabla_{w_{0;0}} l(\boldsymbol{u}), \nabla_{v_{00;0}} l(\boldsymbol{u})]^{\mathsf{T}}||_2$ for these points are shown in figure 4.2(c). $*$

Solving the unconstrained convex problem (4.3) with the gradient descent method involves iterative evaluation of the likelihood function in equation (4.4) and iterative evaluation of the marginal conditional probability masses in equation (4.13) and in equation (4.14). We now extend this procedure and show what it means to solve the strongly convex optimization problem (4.8).

4.2.2 Solving the Exact Strongly Convex Learning

To minimize the negative log posterior $l_{\mathrm{MAP}}(\boldsymbol{u})$ in equation (4.8) with a gradient descent method, we need to be able to iteratively evaluate the negative log posterior $l_{\mathrm{MAP}}(\boldsymbol{u})$ and its gradient $\nabla l_{\mathrm{MAP}}(\boldsymbol{u})$. We include the equation of the partial derivatives

$$\frac{\partial l_{\mathrm{MAP}}(\boldsymbol{u})}{\partial w_{k;j}} = \frac{\partial l(\boldsymbol{u})}{\partial w_{k;j}} - |\mathcal{M}| \frac{\partial \log p(\boldsymbol{u})}{\partial w_{k;j}} \tag{4.15}$$

with respect to the unary parameters $w_{k;j}$ and equation of the partial derivatives

$$\frac{\partial l_{\text{MAP}}(\boldsymbol{u})}{\partial v_{kk';l}} = \frac{\partial l(\boldsymbol{u})}{\partial v_{kk';l}} - |\mathcal{M}| \frac{\partial \log p(\boldsymbol{u})}{\partial v_{kk';l}} \tag{4.16}$$

with respect to the pairwise parameters $v_{kk';j}$.

We evaluate an example of the partial derivative in equation (4.16). The example is a minor computational extension of the gradient in example 4.5 and illustrates how to combine the gradient of the negative log likelihood with the gradient of the prior distribution.

Example 4.7 (Two variable $\mathcal{P}_5$ CRF: Posterior gradient)
We extend previous example 4.3. We use equation (4.16) and result obtained in example 4.5 to evaluate the gradient $\nabla l_{\text{MAP}}(\beta)$*,*

$$\nabla l_{\text{MAP}}(0.9) = \nabla l(0.9) + 0.9 = 1.5144$$

of the negative log posterior density at the point 0.9*. We show the gradient magnitude* $||\nabla l_{\text{MAP}}(\beta)||_2$ *for values of the parameter* β *varied between* -2 *and* 2 *in figure 4.1(f).* $*$

In particular to minimize negative log posterior $l_{\text{MAP}}(\boldsymbol{u})$ in equation (4.8) with Gaussian prior distribution in equation (4.10), we include equation of the partial derivatives

$$\frac{\partial l_{\text{MAP}}(\boldsymbol{u})}{\partial w_{k;j}} = \frac{\partial l(\boldsymbol{u})}{\partial w_{k;j}} + |\mathcal{M}| \frac{w_{k;j}}{\tau^2} \tag{4.17}$$

with respect to the unary parameters $w_{k;j}$ and equation of the partial derivatives

$$\frac{\partial l_{\text{MAP}}(\boldsymbol{u})}{\partial v_{kk';j}} = \frac{\partial l(\boldsymbol{u})}{\partial v_{kk';j}} + |\mathcal{M}| \frac{v_{kk';j}}{\tau^2} \tag{4.18}$$

with respect to the pairwise parameters $v_{kk';j}$.

We evaluate an example of the partial derivative in equation (4.17) and of the partial derivative in equation (4.18). Again the example is minor computational extension of the gradient in example 4.6 and illustrates how to combine the gradient of the negative log likelihood with the gradient of the prior distribution.

Example 4.8 (Two variable $\mathcal{P}$ CRF: Posterior gradient)
Let us continue from example 4.4 and evaluate the gradient $\nabla l_{\mathrm{MAP}}(\boldsymbol{u})$ of the negative log posterior at the point $\boldsymbol{u}$. We use result from example 4.6 to evaluate the first component $\nabla_{w_{0;0}} l_{\mathrm{MAP}}(\boldsymbol{u})$,

$$\frac{\partial l_{\mathrm{MAP}}(\boldsymbol{u})}{\partial w_{0;0}} = \frac{\partial l(\boldsymbol{u})}{\partial w_{0;0}} + w_{0;0} = -0.5$$

of the gradient in equation (4.17) at the parameter vector $\boldsymbol{u}$. We use result from example 4.6 to evaluate the fifth component $\nabla_{v_{00;0}} l_{\mathrm{MAP}}(\boldsymbol{u})$,

$$\frac{\partial l_{\mathrm{MAP}}(\boldsymbol{u})}{\partial v_{00;0}} = \frac{\partial l(\boldsymbol{u})}{\partial v_{00;0}} + v_{00;0} = -0.5963$$

of the gradient in equation (4.18) at the parameter vector $\boldsymbol{u}$. As in example 4.2 we vary the value of the first component $w_{0;0}$ and the value of the fifth component $v_{00;0}$ of the parameter vector $\boldsymbol{u}$ between -1 and 1. In figure 4.2(f) we show for these points the level curves of the magnitude $||[\nabla_{w_{0;0}} l_{\mathrm{MAP}}(\boldsymbol{u}), \nabla_{v_{00;0}} l_{\mathrm{MAP}}(\boldsymbol{u})]^{\mathsf{T}}||_2$. $*$

Solving the strongly convex problem (4.8) with gradient descent method involves iterative evaluation of the posterior function in equation (4.9) and iterative evaluation of the partial derivatives in equation (4.17) and in equation (4.18). We observe that computationally this is only a minor extension of the procedure, described in Section 4.2.1, where we concluded that solving the learning problem involves iterative evaluation of the likelihood function in equation (4.4) and iterative evaluation of the marginal conditional probability masses in equation (4.13) and in equation (4.14). However both evaluation of the likelihood function in equation (4.4) and the marginal conditional probability masses in equation (4.13) and in equation (4.14) involve the evaluation of the conditional probability mass function in equation (3.2), which, as we described, is intractable due to the number of summands in the partition function in equation (3.3) that rises exponentially in the number of sites.

We have shown that each iteration of an iterative descent method, that attempts to solve the exact strongly convex learning problem (4.8), is in general intractable and thus we conclude that solving the exact strongly convex learning problem (4.8) is in general intractable. Hence it is appropriate to adopt an approximate method, which is our next step.

4.3 An Approximate Parameter Learning Principle

In this section we formulate the training of the function f in equation (2.9) as a consistent tractable strongly convex approximation of the intractable problem (4.8). The computational challenge of training the function f translates to the problem of finding the unique minimum of the tractable strongly convex function that results from combining the convex negative log pseudolikelihood [Besag, 1975] with a strongly convex negative log parameter prior distribution.

4.3.1 An Approximate Convex Learning Problem

In this section the problem of approximate parameter learning takes the form of an unconstrained convex optimization problem. The approximate maximum likelihood parameter vector estimate or the maximum pseudolikelihood parameter vector estimate $\hat{\boldsymbol{u}}^{\mathrm{ML}}$,

$$\hat{\boldsymbol{u}}^{\mathrm{ML}} \in \arg\max_{\boldsymbol{u}} \hat{L}(\boldsymbol{u})$$

maximizes what is known in the literature as the pseudolikelihood [Besag, 1975] $\hat{L} : \mathbb{R}^{D_u} \to (0, 1)$,

$$\hat{L}(\boldsymbol{u}) = \prod_{m \in \mathcal{M}} \prod_{i \in \mathcal{I}^m} p(x_i^m \mid \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \boldsymbol{u}) \tag{4.19}$$

of the parameter vector $\boldsymbol{u}$, given a fixed training set $\{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m \in \mathcal{M}}$. In equation (4.19) we define a subvector $\boldsymbol{x}_{\mathcal{N}_i}^m = [\ldots, x_{i'}^m, \ldots]^\mathsf{T}$, $i' \in \mathcal{N}_i$ for a set $\mathcal{N}_i = \{i' \in \mathcal{I}^m \mid (i, i') \in \mathcal{E}^m\}$ of sites neighboring to the site $i \in \mathcal{I}^m$ in the m-th training image. In equation (4.19) local conditional likelihood function $p(x_i^m \mid \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \boldsymbol{u})$,

$$p(x_i^m \mid \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \boldsymbol{u}) = \frac{1}{z_i^m(\boldsymbol{y}^m, \boldsymbol{u})} \exp\Big(-E(x_i^m \mid \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \boldsymbol{u})\Big) \tag{4.20}$$

of parameters $\boldsymbol{u}$ is normalized by the local partition function $z_i^m(\boldsymbol{y}^m, \boldsymbol{u})$,

$$z_i^m(\boldsymbol{y}^m, \boldsymbol{u}) = \sum_{x_i^m \in \mathcal{K}} \exp\Big(-E(x_i^m \mid \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \boldsymbol{u})\Big) \tag{4.21}$$

such that the condition $\sum_{x_i^m \in \mathcal{K}} p(x_i^m \mid \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \boldsymbol{u}) = 1$ is fulfilled. The local conditional likelihood function is governed by the local energy function $E(x_i^m \mid \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \boldsymbol{u})$,

$$E(x_i^m \mid \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \boldsymbol{u}) = E_1(x_i^m \mid \boldsymbol{y}^m, \boldsymbol{w}) + \sum_{i' \in \mathcal{N}_i} E_2(x_i^m, x_{i'}^m \mid \boldsymbol{y}^m, \boldsymbol{v}) \tag{4.22}$$

formed by a subset of unary and pairwise terms in equation (3.5) related to the site i.

The pseudolikelihood function $\hat{L}(\boldsymbol{u})$ in equation (4.19) approximates the likelihood function $L(\boldsymbol{u})$ in equation (4.2). Let us point out that there are K summands in the local partition function in equation (4.21) and that the number of summands in the pseudolikelihood function in equation (4.19) rises linearly in the number of semantic variables in the training set. This is in sharp contrast with the exponential rise for the likelihood function in equation (4.2). It is a known fact that in the infinite data limit both the likelihood estimator and the pseudolikelihood estimator are consistent, that is they yield a parameter estimate that parametrizes a distribution that equals the true underlying distribution, in the case where the model is well-specified [Winkler, 2006]. In the well-specified case the difference between the two estimators lies in the performance on the finite training set. In [Liang and Jordan, 2008] the authors prove for particular risk (expected log-loss) that on a finite training set the above maximum likelihood estimate leads on average to lower risk than the above maximum pseudolikelihood estimate. The consistency can be described in terms of the approximation error and the performance on the finite training set can be characterized in terms of the estimation error, see for instance [Liang and Jordan, 2008]. In [Liang and Jordan, 2008] the authors prove that in the general case, where the model is misspecified, the above likelihood estimator has both lower approximation error and lower asymptotic estimation error as compared to the pseudolikelihood estimator. The authors provide empirical validation of the theoretical analysis.

Equivalently the parameter estimate $\hat{\boldsymbol{u}}^{\mathrm{ML}}$,

$$\hat{\boldsymbol{u}}^{\mathrm{ML}} \in \arg\min_{\boldsymbol{u}} \hat{l}(\boldsymbol{u}) \tag{4.23}$$

minimizes the negative log pseudolikelihood $\hat{l} : \mathbb{R}^{D_u} \to \mathbb{R}_{++}$,

$$\hat{l}(\boldsymbol{u}) = \sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{I}^m} - \log p(x_i^m \mid \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \boldsymbol{u}) \tag{4.24}$$

of the parameter vector $\boldsymbol{u}$, given a fixed training set $\{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m \in \mathcal{M}}$. It can be shown that the negative log pseudolikelihood function $\hat{l}$ in equation (4.24), where the function is determined by the energy function in equation (3.7), is a convex function [Winkler, 2006]. We refer to the problem (4.23) as to the pseudolikelihood (PL) learning.

We evaluate an example of the negative log likelihood $\hat{l}(\boldsymbol{u})$ in equation (4.24) and we plot the graph of this function to illustrate its convexity.

Example 4.9 (Two variable $\mathcal{P}_5$ CRF: Pseudolikelihood)
We continue example 4.1 and consider the negative log pseudolikelihood $\hat{l}(\beta)$ of the scalar parameter β. We use values of the unary and of the pairwise terms of the energy function in example 3.2 to evaluate local energy functions $E(x_i^1 = k \mid x_{\mathcal{N}_i}^1, \boldsymbol{y}^1, 0.9)$,

$$\begin{aligned} E(x_0^1 = 0 \mid x_1^1, \boldsymbol{y}^1, 0.9) &= 1 \\ E(x_0^1 = 1 \mid x_1^1, \boldsymbol{y}^1, 0.9) &= 0.9 \\ E(x_1^1 = 0 \mid x_0^1, \boldsymbol{y}^1, 0.9) &= 0.9 \\ E(x_1^1 = 1 \mid x_0^1, \boldsymbol{y}^1, 0.9) &= 1 \end{aligned}$$

in equation (4.22) at the parameter value 0.9*, partition functions $z_i^1(\boldsymbol{y}^1, 0.9)$,*

$$\begin{aligned} z_0^1(\boldsymbol{y}^1, 0.9) &= 0.7744 \\ z_1^1(\boldsymbol{y}^1, 0.9) &= 0.7744 \end{aligned}$$

in equation (4.21) at the parameter value 0.9*, and local conditional likelihood functions $p(x_i^1 \mid x_{\mathcal{N}_i}^1, \boldsymbol{y}^1, 0.9)$,*

$$\begin{aligned} p(x_0^1 \mid x_1^1, \boldsymbol{y}^1, 0.9) &= 0.5250 \\ p(x_1^1 \mid x_0^1, \boldsymbol{y}^1, 0.9) &= 0.5250 \end{aligned}$$

in equation (4.20) at the parameter value 0.9*. We use the local conditional likelihood values to evaluate the pseudolikelihood $\hat{L}(0.9)$,*

$$\hat{L}(0.9) = 0.2756$$

in equation (4.19) of the parameter value 0.9*. At last we use the pseudolikelihood value to evaluate the negative log pseudolikelihood $\hat{l}(0.9)$,*

$$\hat{l}(0.9) = 1.2888$$

in equation (4.24) of the parameter value 0.9*. Values of the above two functions for parameter value β being varied between -2 and 2 are respectively shown in figure 4.1(g) and in figure 4.1(h). We observe that figure 4.1(h) indeed shows a graph of a convex function.* $\ast$

We evaluate another example of the negative log pseudolikelihood $\hat{l}(\boldsymbol{u})$ in equation (4.24) and again we plot the graph of the function to illustrate its convexity.

Example 4.10 (Two variable $\mathcal{P}$ CRF: Pseudolikelihood)
We continue example 4.2 and consider the negative log pseudolikelihood $\hat{l}(\boldsymbol{u})$ of the parameter vector $\boldsymbol{u}$ specified in equation (3.14) in example 3.1. We use values of the unary terms and of the pairwise terms of the energy function in example 3.1 to evaluate local energy functions $E(x_i^1 \mid x_{i'}^1, \boldsymbol{y}^1, \boldsymbol{u})$,

$$\begin{array}{rcr} E(x_0^1 = 0 \mid x_1^1, \boldsymbol{y}^1, \boldsymbol{u}) & = & 1 \\ E(x_0^1 = 1 \mid x_1^1, \boldsymbol{y}^1, \boldsymbol{u}) & = & -1 \\ E(x_1^1 = 0 \mid x_0^1, \boldsymbol{y}^1, \boldsymbol{u}) & = & -1 \\ E(x_1^1 = 1 \mid x_0^1, \boldsymbol{y}^1, \boldsymbol{u}) & = & 1 \end{array}$$

in equation (4.22) at the parameter vector $\boldsymbol{u}$, partition functions $z_i^1(\boldsymbol{y}^1, \boldsymbol{u})$,

$$z_0^1(\boldsymbol{y}^1, \boldsymbol{u}) = 3.0862$$
$$z_1^1(\boldsymbol{y}^1, \boldsymbol{u}) = 3.0862$$

in equation (4.21) at the parameter vector $\boldsymbol{u}$, and local conditional likelihood functions $p(x_i^1 \mid x_{i'}^1, \boldsymbol{y}^1, \boldsymbol{u})$,

$$p(x_0^1 \mid x_1^1, \boldsymbol{y}^1, \boldsymbol{u}) = 0.8808$$
$$p(x_1^1 \mid x_0^1, \boldsymbol{y}^1, \boldsymbol{u}) = 0.8808$$

in equation (4.20) at the parameter vector $\boldsymbol{u}$. We use the local conditional likelihood function values to evaluate the pseudolikelihood $\hat{L}(\boldsymbol{u})$,

$$\hat{L}(\boldsymbol{u}) = 0.7758$$

in equation (4.19) of the parameter vector $\boldsymbol{u}$. At last we use the pseudolikelihood value to evaluate the negative log pseudolikelihood $\hat{l}(\boldsymbol{u})$,

$$\hat{l}(\boldsymbol{u}) = 0.2539$$

in equation (4.24) of the parameter vector $\boldsymbol{u}$. We vary the value of the first component $w_{0;0}$ and the value of the fifth component $v_{00;0}$ of the parameter vector $\boldsymbol{u}$ between -1 and 1. Level curves of the pseudolikelihood function at these points and level curves of the negative log pseudolikelihood function at these points are respectively shown in figure 4.2(g) and in figure 4.2(h). Figure 4.2(h) illustrates the convexity of the negative log pseudolikelihood function. $*$

We have described the desired approximate model parameter vector as a solution of the unconstrained convex optimization problem (4.23). In the following section we turn the problem (4.23) into an unconstrained strongly convex optimization problem by combining the convex negative log pseudolikelihood in equation (4.24) with a strongly convex negative log parameter prior distribution.

4.3.2 An Approximate Strongly Convex Learning Problem

In this section the problem of approximate parameter learning takes the form of an unconstrained strongly convex optimization problem. The approximate maximum a posteriori parameter vector estimate or the maximum pseudo a posteriori parameter estimate $\hat{\boldsymbol{u}}^{\mathrm{MAP}}$,

$$\hat{\boldsymbol{u}}^{\mathrm{MAP}} \in \arg\max_{\boldsymbol{u}} \hat{L}_{\mathrm{MAP}}(\boldsymbol{u})$$

maximizes the pseudoposterior density $\hat{L}_{\mathrm{MAP}} : \mathbb{R}^{D_u} \to \mathbb{R}_{++}$,

$$\hat{L}_{\mathrm{MAP}}(\boldsymbol{u}) = \prod_{m \in \mathcal{M}} \left(\prod_{i \in \mathcal{I}^m} p(x_i^m \mid \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \boldsymbol{u}) p(\boldsymbol{u}) \right) \tag{4.25}$$

of the parameter vector $\boldsymbol{u}$, given a fixed training set $\{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m \in \mathcal{M}}$. The pseudoposterior density function in equation (4.25) approximates the posterior density function in equation (4.7). Equivalently the parameter vector estimate $\hat{\boldsymbol{u}}^{\mathrm{MAP}}$,

$$\hat{\boldsymbol{u}}^{\mathrm{MAP}} \in \arg\min_{\boldsymbol{u}} \hat{l}_{\mathrm{MAP}}(\boldsymbol{u}) \tag{4.26}$$

minimizes the negative log pseudoposterior density $\hat{l}_{\mathrm{MAP}} : \mathbb{R}^{D_u} \to \mathbb{R}$,

$$\hat{l}_{\mathrm{MAP}}(\boldsymbol{u}) = \left(\hat{l}(\boldsymbol{u}) - |\mathcal{M}| \log p(\boldsymbol{u}) \right) \tag{4.27}$$

of the parameter vector $\boldsymbol{u}$, given a fixed training set $\{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m \in \mathcal{M}}$. We adopt a prior probability density $p(\boldsymbol{u})$ forming the normal (or Gaussian) distribution $p(\boldsymbol{u} \mid \tau) = \mathcal{N}(\boldsymbol{u} \mid \boldsymbol{0}, \tau^2 \boldsymbol{I})$ in equation (4.10) with the mean vector $\boldsymbol{0}$ and the variance τ^2. We use the argument that we used to show that the negative log posterior in equation (4.9) is strongly convex, also to establish the strong convexity of the negative log pseudoposterior in equation (4.27). Let us note that conceptually the problem (4.26) is equivalent to the problem (4.23) if prior information is unavailable and an uninformative prior probability density function $p(\boldsymbol{u})$ is employed. Specifically in case of the adopted Gaussian prior $p(\boldsymbol{u} \mid \tau)$ the problem (4.23) is equivalent to the problem (4.26), when an uninformative Gaussian prior $p_{\mathrm{PL}}(\boldsymbol{u} \mid \infty)$,

$$p_{\mathrm{PL}}(\boldsymbol{u} \mid \infty) = \mathcal{N}(\boldsymbol{u} \mid \boldsymbol{0}, \mathbf{Diag}(\infty \mathbf{1}_{D_u})) \tag{4.28}$$

with infinite standard deviation is employed. Hence we refer both to the problem (4.23) with no prior and to the conceptually equivalent problem (4.26) with prior in equation (4.28) as to the pseudolikelihood (PL) learning.

Let us evaluate an example of the negative log pseudoposterior density in equation (4.27). We plot the graph of the negative log pseudoposterior density to illustrate the shift in location as compared to the analogous negative log pseudolikelihood function.

Example 4.11 (Two variable $\mathcal{P}_5$ CRF: Pseudoposterior)
Let us continue example 4.9 and employ a prior probability density function over the parameter β in the form of the Gaussian distribution $p(\beta \mid \tau)$, where we set the mean to 0 *and the variance $\tau^2 = 1$. We use the pseudolikelihood value $\hat{L}(0.9)$ from example 4.9 to evaluate the pseudoposterior $\hat{L}_{\mathrm{MAP}}(\beta)$,*

$$\hat{L}_{\mathrm{MAP}}(0.9) = 0.0733$$

in equation (4.25) at the parameter value 0.9*. Afterwards we use the pseudoposterior value to evaluate the negative log pseudoposterior $\hat{l}_{\mathrm{MAP}}(\beta)$,*

$$\hat{l}_{\mathrm{MAP}}(0.9) = 2.6127$$

in equation (4.27) at the parameter value 0.9*. Values of the pseudoposterior function and values of the negative log pseudoposterior function for parameter value β being varied between -2 and 2 are respectively shown in figure 4.1(j) and in figure 4.1(k). Figure 4.1(k) illustrates the shift in location of the optimum as compared to the analogous negative log pseudolikelihood in figure 4.1(h).* $\ast$

We evaluate another example of the negative log pseudoposterior density in equation (4.27) and as in the previous example we plot the graph of the function to illustrate the shift in location of the optimum.

Example 4.12 (Two variable $\mathcal{P}$ CRF: Pseudoposterior)
We continue example 4.10 by employing a prior probability density function over the parameter vector $\boldsymbol{u}$ in the form of the Gaussian distribution in equation (4.10), where we set the variance $\tau^2 = 1$. We use the pseudolikelihood $\hat{L}(\boldsymbol{u})$ value from example 4.10 to evaluate the pseudoposterior $\hat{L}_{\mathrm{MAP}}(\boldsymbol{u})$,

$$\hat{L}_{\mathrm{MAP}}(\boldsymbol{u}) \approx 0$$

in equation (4.25) at the parameter vector $\boldsymbol{u}$. This is a very small number. Afterwards we use the pseudoposterior value to evaluate the negative log pseudoposterior $\hat{l}_{\mathrm{MAP}}(\boldsymbol{u})$,

$$\hat{l}_{\mathrm{MAP}}(\boldsymbol{u}) = 15.0311$$

in equation (4.27) at the parameter vector $\boldsymbol{u}$. As in example 4.2 we vary the value of the first component $w_{0;0}$ and the value of the fifth component $v_{00;0}$ of the parameter vector $\boldsymbol{u}$ between -1 and 1. Level curves of the pseudoposterior function at these points and level curves of the negative log pseudoposterior function at these points are respectively shown in figure 4.2(j) and in figure 4.2(k). Figure 4.2(k) illustrates the shift in location of the optimum as compared to the analogous negative log pseudolikelihood in figure 4.2(h). $*$

We have described the desired approximate parameter vector estimate as a solution of the unconstrained strongly convex optimization problem (4.26). Again as we explain in Chapter 1 an iterative descent method is theoretically guaranteed to solve a strongly convex optimization problem up to a precision in a finite number of iterations. In the following we show that performing one such iteration is now computationally tractable.

4.4 A Tractable Approximate Learning

In this section we explain what it means to efficiently solve the unconstrained strongly convex optimization problem (4.26) and why training the function f in equation (2.9) in this manner in practice becomes computationally *tractable*. The computational challenge of training the function f translates to the problem of evaluating local conditional likelihoods and local conditional distributions. Specifically solving the problem (4.26) involves iterative evaluation of a product of local conditional likelihoods and iterative evaluation of expectations with the local conditional distributions.

Our contribution is to provide the first partial derivative equations of the objective needed to compute the solution with gradient descent methods or with faster converging conjugate gradient methods or with quasi-Newton methods. To the best of our knowledge the partial derivative equations have not been provided in the literature reporting the pseudolikelihood based learning experiments. We are only aware of the pseudolikelihood gradient for the general Markov random field from the exponential family without conditioning that is provided in [Winkler, 2006].

Let us point out that it would be also possible to optimize the pseudolikelihood based learning objective with second-order methods for convex optimization. By second-order methods we mean the rapidly convergent Newton's method that work locally with the second-order Taylor approximation. See for instance the Newton's method in Algorithm 9.5 in [Boyd and Vandenberghe, 2004]. This is possible due to the fact that the strongly convex learning objective in the problem (4.26) is twice differentiable and, hence, object function can be evaluated together with both the gradient and the Hessian. We do not include the second partial derivative equations of the objective. Interested reader shall find hints in [Winkler, 2006, Wainwright and Jordan, 2008]. In [Bottou and Bousquet, 2008] the authors point out that the generalization performance of a learning system is not only dependent on the convergence rate of an optimization algorithm, which results in an optimization error, however also on the statistical properties of the objective function, which result in the approximation and the estimation error. Definition of the approximation and the estimation error can be found for instance in [Bottou and Bousquet, 2008, Liang and Jordan, 2008]. In [Bottou and Bousquet, 2008] the authors note that when the approximation and the estimation errors are dominant, there is no need for accurate optimization. In the setting of large-scale learning, where the computation time is limited, they argue in favor of approximate and first order optimization methods that may eventually lead to better generalization as in the limited time more training examples can be processed.

4.4.1 Solving the Approximate Convex Learning Problem

To minimize the negative log pseudolikelihood $\hat{l}(\boldsymbol{u})$ in equation (4.23) with gradient descent methods, conjugate gradient methods or with quasi-Newton methods, we need to be able to iteratively evaluate the negative log pseudolikelihood $\hat{l}(\boldsymbol{u})$ and its gradient $\nabla\hat{l}(\boldsymbol{u})$ or, more specifically, its components, the partial derivatives. We begin by considering the partial derivatives of the negative log pseudolikelihood for the pairwise homogeneous isotropic conditional random field in equation (3.5). The partial derivatives with respect to the unary parameters $w_{k;j}$ are given by

$$\frac{\partial\hat{l}(\boldsymbol{u})}{\partial w_{k;j}} = \sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{I}^m}\left(\frac{\partial E_i(x_i^m,\boldsymbol{w})}{\partial w_{k;j}} - \left\langle\frac{\partial E_i(x_i'^m,\boldsymbol{w})}{\partial w_{k;j}}\right\rangle_{i,m,u}\right) \tag{4.29}$$

and the partial derivatives with respect to the pairwise parameters $v_{kk';j}$ are given by

$$\begin{aligned}\frac{\partial\hat{l}(\boldsymbol{u})}{\partial v_{kk';j}} = \sum_{m\in\mathcal{M}}\sum_{(i,i')\in\mathcal{E}^m}\Bigg(2&\frac{\partial E_{ii'}(x_i^m,x_{i'}^m,\boldsymbol{v})}{\partial v_{kk';j}}\\ &-\left\langle\frac{\partial E_{ii'}(x_i'^m,x_{i'}^m,\boldsymbol{v})}{\partial v_{kk';j}}\right\rangle_{i,m,u}\\ &-\left\langle\frac{\partial E_{ii'}(x_i^m,x_{i'}'^m,\boldsymbol{v})}{\partial v_{kk';j}}\right\rangle_{i',m,u}\Bigg)\mu_{ii';j}\end{aligned} \tag{4.30}$$

For compactness in the above two equations we drop the dependence on image data $\boldsymbol{y}^m$ in both the unary and the pairwise terms. The terms involving the angle brackets $\langle\rangle_{i,m,u}$,

$$\langle g(x_i'^m)\rangle_{i,m,u} = \sum_{x_i'^m \in \mathcal{K}} g(x_i'^m) p(x_i'^m \mid \boldsymbol{x}_{\mathcal{N}_i}'^m, \boldsymbol{y}^m, \boldsymbol{u})$$

denote expectations with the local conditional probability mass function $p(x_i'^m \mid \boldsymbol{x}_{\mathcal{N}_i}'^m, \boldsymbol{y}^m, \boldsymbol{u})$. These terms result from differentiating the local log partition function. Proof of the log pseudolikelihood gradient form for the general Markov random field from the exponential family without conditioning is given in [Winkler, 2006].

We evaluate an example of the partial derivative in equation (4.30). We encourage the interested reader to verify the form of the gradient and subsequently to verify the value of the gradient at the location suggested in the example using a pocket calculator.

Example 4.13 (Two variable $\boldsymbol{\mathcal{P}}_5$ CRF: Pseudolikelihood grad.) *We continue from example 4.9. We use local conditional probability masses in example 4.9 to evaluate the gradient $\nabla\hat{l}(\beta)$,*

$$\nabla\hat{l}(0.9) = 2-2\delta(x_0^1-x_1^1)-\langle 1-\delta(x_0'^1-x_1^1)\rangle_{0,1,0.9}-\langle 1-\delta(x_0^1-x_1'^1)\rangle_{1,1,0.9} = 0.9500$$

in equation (4.30) of the negative log pseudolikelihood at the parameter value 0.9*. We put our pocket calculator aside and identify the computed value in figure 4.1(i), where we show the gradient magnitude $||\nabla\hat{l}(\beta)||_2$ for values of the parameter β varied between* -2 *and* 2*.* $*$

Partial derivatives in equation (4.29) and in equation (4.30) are partial derivative equations for the pairwise homogeneous isotropic conditional random field in equation (3.5). To minimize the negative log pseudolikelihood $\hat{l}(\boldsymbol{u})$ in equation (4.23) of parameters $\boldsymbol{u}$ of a CRF of the class $\mathcal{P}$ of models in equation (3.1) with a gradient descent method, conjugate gradient methods or with quasi-Newton methods, we need to specialize equation (4.29) and equation (4.30) with respect to equation (3.7). The partial derivatives with respect to the unary parameters $w_{k;j}$ are given by

$$\frac{\partial\hat{l}(\boldsymbol{u})}{\partial w_{k;j}} = \sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{I}^m}\Big(\delta(x_i^m-k) - p(x_i^m = k \mid \boldsymbol{x}_{\mathcal{N}_i}^m, \boldsymbol{y}^m, \boldsymbol{u})\Big)h_{i;j}(\boldsymbol{y}^m) \qquad (4.31)$$

and the partial derivatives with respect to the pairwise parameters $v_{kk';j}$ are given by

$$\begin{aligned}\frac{\partial \hat{l}(\boldsymbol{u})}{\partial v_{kk';j}} = \sum_{m\in\mathcal{M}} \sum_{(i,i')\in\mathcal{E}^m} \Big(& 2\cdot\delta(x_i^m - k)\delta(x_{i'}^m - k') \\ & - \delta(x_{i'}^m - k')p(x_i^m = k \mid \boldsymbol{x}_{N_i}^m, \boldsymbol{y}^m, \boldsymbol{u}) \\ & - \delta(x_i^m - k)p(x_{i'}^m = k' \mid \boldsymbol{x}_{N_{i'}}^m, \boldsymbol{y}^m, \boldsymbol{u})\Big)\mu_{ii';j}\end{aligned} \tag{4.32}$$

To show that equation (4.31) and equation (4.32) are the correct specializations of equation (4.29) and equation (4.30), we use the argument we used to show that equation (4.13) and equation (4.14) are the correct specializations of equation (4.11) and equation (4.12), where we replace the expectations using the global conditional probability mass function with the expectations using the local conditional probability mass functions.

We evaluate an example of the key equations that need to be implemented in order to train the function f for automatic image interpretation. It is the partial derivative in equation (4.31) and the partial derivative in equation (4.32). Ready verification of both the form of the equations in the following example and the values therein using our pocket calculator is a necessary prerequisite for a successful implementation of the trainable function for automatic image interpretation described in this thesis.

Example 4.14 (Two variable $\mathcal{P}$ CRF: Pseudolikelihood gradient) *Let us extend example 4.10 and evaluate the gradient $\nabla\hat{l}(\boldsymbol{u})$ of the negative log pseudolikelihood at the point $\boldsymbol{u}$ specified in equation (3.14) in example 3.1. It is a vector with* 12 *components. We first use local conditional probability masses in example 4.10 to evaluate the first component $\nabla_{w_{0;0}}\hat{l}(\boldsymbol{u})$,*

$$\frac{\partial \hat{l}(\boldsymbol{u})}{\partial w_{0;0}} = \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} p(x_0^1 = 0 \mid x_1^1, \boldsymbol{y}^1, \boldsymbol{u}) \\ p(x_1^1 = 0 \mid x_0^1, \boldsymbol{y}^1, \boldsymbol{u}) \end{bmatrix}\right)^{\mathsf{T}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 0$$

of the gradient in equation (4.31) at the parameter vector $\boldsymbol{u}$. We use local conditional probability masses in example 4.10 to evaluate the fifth component $\nabla_{v_{00;0}}\hat{l}(\boldsymbol{u})^{\mathsf{T}}$,

$$\frac{\partial \hat{l}(\boldsymbol{u})}{\partial v_{00;0}} = \Big(2\cdot 0 - 1p(x_0^1 = 0 \mid x_1^1, \boldsymbol{y}^1, \boldsymbol{u}) - 0p(x_1^1 = 0 \mid x_0^1, \boldsymbol{y}^1, \boldsymbol{u})\Big)\,1 = -0.1192$$

of the gradient in equation (4.32) at the parameter vector $\boldsymbol{u}$. We leave our pocket calculator aside and vary the value of the first component $w_{0;0}$ and the *value of the fifth component* $v_{00;0}$ *of the parameter vector* $\boldsymbol{u}$ *between* -1 *and* 1*. Level curves of the magnitude* $||[\nabla_{w_{0;0}}\hat{l}(\boldsymbol{u}), \nabla_{v_{00;0}}\hat{l}(\boldsymbol{u})]^{\mathsf{T}}||_2$ *for these points are shown in figure 4.2(i).* $*$

Solving the unconstrained convex problem (4.23) with gradient descent method involves iterative evaluation of the pseudolikelihood function in equation (4.24) and iterative evaluation of the local conditional probability masses in equation (4.31) and in equation (4.32). We now extend this procedure and show what it means to solve the strongly convex optimization problem (4.26).

4.4.2 Solving the Approximate Strongly Convex Learning

To minimize the negative log pseudoposterior $\hat{l}_{\text{MAP}}(\boldsymbol{u})$ in equation (4.26) with gradient descent methods, conjugate gradient methods or with quasi-Newton methods, we again need to be able to iteratively evaluate the negative log pseudoposterior $\hat{l}_{\text{MAP}}(\boldsymbol{u})$ and its gradient $\nabla\hat{l}_{\text{MAP}}(\boldsymbol{u})$. We include equation of the partial derivatives

$$\frac{\partial \hat{l}_{\text{MAP}}(\boldsymbol{u})}{\partial w_{k;j}} = \frac{\partial \hat{l}(\boldsymbol{u})}{\partial w_{k;j}} - |\mathcal{M}|\frac{\partial \log p(\boldsymbol{u})}{\partial w_{k;j}} \tag{4.33}$$

with respect to the unary parameters $w_{k;j}$ and equation of the partial derivatives

$$\frac{\partial \hat{l}_{\text{MAP}}(\boldsymbol{u})}{\partial v_{kk';l}} = \frac{\partial \hat{l}(\boldsymbol{u})}{\partial v_{kk';l}} - |\mathcal{M}|\frac{\partial \log p(\boldsymbol{u})}{\partial v_{kk';l}} \tag{4.34}$$

with respect to the pairwise parameters $v_{kk';j}$.

We evaluate an example of the partial derivative in equation (4.34). The following example extends the gradient of the negative log likelihood function in example 4.13 with the gradient of the parameter prior distribution.

Example 4.15 (Two variable $\mathcal{P}_5$ CRF: Pseudoposterior grad.) *We continue previous example 4.11. We use gradient value* $\nabla\hat{l}(0.9)$ *of the negative log pseudolikelihood at the parameter value* 0.9 *from example 4.13 to evaluate the gradient* $\nabla\hat{l}_{\text{MAP}}(0.9)$*,*

$$\nabla\hat{l}_{\text{MAP}}(0.9) = \nabla\hat{l}(0.9) + 0.9 = 1.8500$$

of the negative log pseudoposterior density in equation (4.34) at the parameter value 0.9*. We show the gradient magnitude* $||\nabla\hat{l}_{\text{MAP}}(\beta)||_2$ *for values of the parameter* β *varied between* -2 *and* 2 *in figure 4.1(l).* $*$

In particular to minimize negative log pseudoposterior $\hat{l}_{\mathrm{MAP}}(\boldsymbol{u})$ in equation (4.26) with Gaussian prior distribution in equation (4.10), we derive equation of the partial derivatives

$$\frac{\partial \hat{l}_{\mathrm{MAP}}(\boldsymbol{u})}{\partial w_{k;j}} = \frac{\partial \hat{l}(\boldsymbol{u})}{\partial w_{k;j}} + |\mathcal{M}| \frac{w_{k;j}}{\tau^2} \tag{4.35}$$

with respect to the unary parameters $w_{k;j}$ and equation of the partial derivatives

$$\frac{\partial \hat{l}_{\mathrm{MAP}}(\boldsymbol{u})}{\partial v_{kk';j}} = \frac{\partial \hat{l}(\boldsymbol{u})}{\partial v_{kk';j}} + |\mathcal{M}| \frac{v_{kk';j}}{\tau^2} \tag{4.36}$$

with respect to the pairwise parameters $v_{kk';j}$.

We evaluate an example of the partial derivative in equation (4.35) and in equation (4.36). Again the following example extends the gradient of the negative log likelihood function in example 4.14 with the gradient of the parameter prior distribution. This computationally minor extension completes the equations of the gradient that we need to implement when realizing the trainable function for automatic image interpretation described in this thesis.

Example 4.16 (Two variable $\mathcal{P}$ CRF: Pseudoposterior gradient) *We continue from example 4.12 and evaluate the gradient $\nabla \hat{l}_{\mathrm{MAP}}(\boldsymbol{u})$ of the negative log pseudoposterior at the parameter value $\boldsymbol{u}$ specified in equation (3.14) in example 3.1. Let us use partial derivative value $\frac{\partial \hat{l}(\boldsymbol{u})}{\partial w_{0;0}}$ at the parameter vector $\boldsymbol{u}$ in example 4.14 to evaluate the first component $\nabla_{w_{0;0}} \hat{l}_{\mathrm{MAP}}(\boldsymbol{u})$,*

$$\frac{\partial \hat{l}_{\mathrm{MAP}}(\boldsymbol{u})}{\partial w_{0;0}} = \frac{\partial \hat{l}(\boldsymbol{u})}{\partial w_{0;0}} + w_{0;0} = -0.5$$

of the gradient in equation (4.35) at the parameter vector $\boldsymbol{u}$. We use partial derivative value $\frac{\partial \hat{l}(\boldsymbol{u})}{\partial v_{00;0}}$ at the parameter vector $\boldsymbol{u}$ from example 4.14 to evaluate the fifth component $\nabla_{v_{00;0}} \hat{l}_{\mathrm{MAP}}(\boldsymbol{u})$,

$$\frac{\partial \hat{l}_{\mathrm{MAP}}(\boldsymbol{u})}{\partial v_{00;0}} = \frac{\partial \hat{l}(\boldsymbol{u})}{\partial v_{00;0}} + v_{00;0} = -0.6192$$

of the gradient in equation (4.36) at the parameter vector $\boldsymbol{u}$. As in example 4.2 we vary the value of the first component $w_{0;0}$ and the value of the fifth component $v_{00;0}$ of the parameter vector $\boldsymbol{u}$ between -1 and 1. We show the level curves of the magnitude $||[\nabla_{w_{0;0}} \hat{l}_{\mathrm{MAP}}(\boldsymbol{u}), \nabla_{v_{00;0}} \hat{l}_{\mathrm{MAP}}(\boldsymbol{u})]^{\mathsf{T}}||_2$ for these points in figure 4.2(l). $*$

Solving the strongly convex optimization problem (4.26) with the gradient descent method involves iterative evaluation of the pseudoposterior function in equation (4.27) and iterative evaluation of the partial derivatives in equation (4.35) and in equation (4.36). Again we observe that computationally this is only a minor extension of the procedure, that we have described Section 4.4.1, where we concluded that solving the learning problem involves iterative evaluation of the pseudolikelihood function in equation (4.24) and iterative evaluation of the local conditional probability masses in equation (4.31) and in equation (4.32). Both evaluation of the pseudolikelihood function in equation (4.24) and of the local conditional probability masses in equation (4.31) and in equation (4.32) involve the evaluation of the local conditional probability mass function in equation (4.20), which is tractable due to only K summands in the partition function in equation (4.21). The number of summands in the pseudolikelihood function rises linearly in the number of semantic variables. As already mentioned above this is in sharp contrast to the exponential rise in case of the original likelihood function. We have shown that each iteration of an iterative descent method that attempts to solve the approximate strongly convex learning problem (4.26), is now tractable and hence we conclude that solving the approximate strongly convex learning problem (4.26) is tractable.

In summary we have formulated the training of the function f in equation (2.9) as a tractable strongly convex approximation of the intractable problem (4.8). A natural question is how good the approximation is as compared to the approximated objective. In Chapter 1 we point out that it has been theoretically proved that under certain conditions the pseudolikelihood approximation is consistent. In the next section we illustrate the relation between the likelihood function and its approximation on an example.

4.5 A Likelihood vs. Pseudolikelihood Example

Below we evaluate an example of the negative log pseudolikelihood in equation (4.24) together with an example of the negative log likelihood in equation (4.4). We adopt a problem that is small enough to allow computation of the full likelihood involving complete labeling enumeration. To minimize these two objective functions with gradient descent methods, conjugate gradient methods or with quasi-Newton methods, we need to compute their gradients. Hence we evaluate an example of the partial derivative in equation (4.30) together with an example of the partial derivative in equation (4.12).

We begin by specifying the pixel site level image representation. We specify the set $\mathcal{I}$ in equation (2.2), the associated vector $\boldsymbol{y}$ in equation (2.1) and the associated vector $\boldsymbol{x}$ in equation (2.3). Let us consider an image with 4 × 4 pixel sites. The set $\mathcal{I} = \{0, \dots, 15\}$ contains 16 pixel sites. The associated vector $\boldsymbol{y} = [\boldsymbol{y}_0^\mathsf{T}, \dots, \boldsymbol{y}_{15}^\mathsf{T}]^\mathsf{T}$ is composed of 16 pixel color vectors. The associated vector $\boldsymbol{x} = [x_0, \dots, x_{15}]^\mathsf{T}$ contains 16 pixel class labels. In figure 4.3 we show the pixel sites, the pixel color vectors and the pixel class labels in a 4 × 4 grid.

$$\underset{\text{(a)}}{\begin{bmatrix} 0 & 4 & 8 & 12 \\ 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \end{bmatrix}} \quad \underset{\text{(b)}}{\begin{bmatrix} \boldsymbol{y}_0 & \boldsymbol{y}_4 & \boldsymbol{y}_8 & \boldsymbol{y}_{12} \\ \boldsymbol{y}_1 & \boldsymbol{y}_5 & \boldsymbol{y}_9 & \boldsymbol{y}_{13} \\ \boldsymbol{y}_2 & \boldsymbol{y}_6 & \boldsymbol{y}_{10} & \boldsymbol{y}_{14} \\ \boldsymbol{y}_3 & \boldsymbol{y}_7 & \boldsymbol{y}_{11} & \boldsymbol{y}_{15} \end{bmatrix}} \quad \underset{\text{(c)}}{\begin{bmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{bmatrix}}$$

Figure 4.3: *Pixel level image representation.* Image with 4 × 4 pixels: (a) Pixel sites $\mathcal{I} = \{0, \dots, 15\}$, (b) image data $\boldsymbol{y} = [\boldsymbol{y}_0^\mathsf{T}, \dots, \boldsymbol{y}_{15}^\mathsf{T}]^\mathsf{T}$ and (c) image labeling $\boldsymbol{x} = [x_0, \dots, x_{15}]^\mathsf{T}$.

In our example and later in the experiments we make a particular choice of the directed graph $\mathcal{G}$ and work with the four nearest-neighbor lattice graph defined over the set $\mathcal{I}$ of pixel sites. Let us note that concepts developed in this work do not rely on the particular choice of the underlying graph made here. We illustrate this particular choice in the following. We define the four nearest-neighbor lattice graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ over the pixel sites $\mathcal{I}$. The set $\mathcal{E} = \{(0,1), \dots, (14,15)\}$ contains 24 directed edges and each edge defines an oriented pair of pixel class label variables that are indexed by the pixel sites included in the edge. Pixel class labels in figure 4.3(c) are depicted as circles in figure 4.4. Pairs of pixel class label variables are in figure 4.4 depicted with lines.

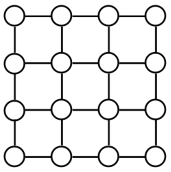
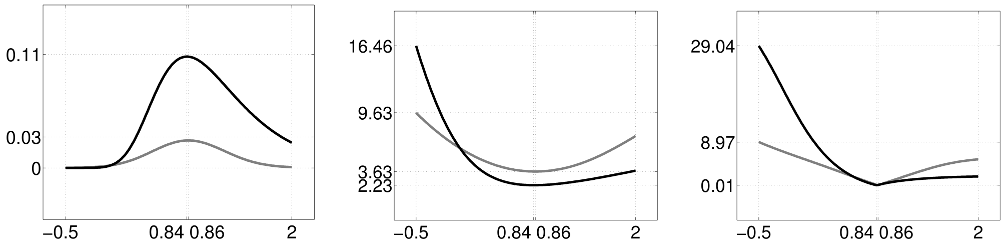


Figure 4.4: *Lattice graph.* Image with 4 × 4 pixels: Circles denote pixel class label variables and lines denote pairs of pixel class label variables.

We shall consider the functional form of the conditional random field energy specified in example 3.2. We consider the negative log likelihood and the negative log pseudolikelihood of the scalar parameter β given the labeled training image $\{\boldsymbol{x}^1, \boldsymbol{y}^1\}$ in figure 4.5. The set of pixel sites $\mathcal{I}^1$ contains our 16 sites located in the 4 × 4 grid and the set $\mathcal{E}^1$ of ordered pairs of neighboring sites contains our 24 pairs of sites. The likelihood in equation (4.2) and the pseudolikelihood in equation (4.19) are displayed in figure 4.6(a). The negative log likelihood from equation (4.4) is shown in gray in figure 4.6(b). The negative log pseudolikelihood from equation (4.24) is shown in figure 4.6(b) in black. All functions are evaluated for values of the parameter β varied between -0.5 and 2.

$$\underset{\text{(a)}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}} \qquad \underset{\text{(b)}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}$$

Figure 4.5: *Labeled training image.* (a) Image $\boldsymbol{y}^1 = [y_0^1, .., y_{15}^1]^\mathsf{T}$ and (b) image labeling $\boldsymbol{x}^1 = [x_0^1, .., x_{15}^1]^\mathsf{T}$.



(a) (b) (c)

Figure 4.6: *Likelihood and pseudolikelihood.* Parameter β and its (a) likelihood $L(\beta)$ (gray) and pseudolikelihood $\hat{L}(\beta)$ (black), given a labeled training image in figure 4.5. (b) Negative log likelihood $l(\beta)$ (gray) and negative log pseudolikelihood $\hat{l}(\beta)$ (black). (c) Gradient magnitude $||\nabla l(\beta)||_2$ (gray) of the negative log likelihood and gradient magnitude $||\nabla \hat{l}(\beta)||_2$ (black) of the negative log pseudolikelihood. The meaning of the values 0.84 and 0.86 is explained in the text.

Let us continue by forming the gradient $\nabla l(\beta)$,

$$\nabla l(\beta) = \sum_{(i,i')\in\mathcal{E}^1} \Big(1 - \delta(x_i^1 - x_{i'}^1) - \langle 1 - \delta(x_i'^1 - x_{i'}'^1)\rangle_{1,\beta}\Big)$$

of the likelihood in equation (4.12) and the gradient $\nabla\hat{l}(\beta)$,

$$\nabla\hat{l}(\beta) = \sum_{(i,i')\in\mathcal{E}^1} \Big(2 - 2\delta(x_i^1 - x_{i'}^1) - \langle 1 - \delta(x_i'^1 - x_{i'}^1)\rangle_{i,1,\beta} - \langle 1 - \delta(x_i^1 - x_{i'}'^1)\rangle_{i',1,\beta}\Big)$$

of the pseudolikelihood in equation (4.30). We show the magnitude $||\nabla l(\beta)||_2$ of the likelihood gradient and the magnitude $||\nabla\hat{l}(\beta)||_2$ of the pseudolikelihood gradient for parameter values β varied between -0.5 and 2 in respectively gray and black in figure 4.6(c). The minimum $l(\beta^{\mathrm{ML}}) = 3.63$ of the likelihood function is attained at the parameter value $\beta^{\mathrm{ML}} = 0.86$ and the minimum $\hat{l}(\hat{\beta}^{\mathrm{ML}}) = 2.23$ of the pseudolikelihood function is attained at the parameter value $\hat{\beta}^{\mathrm{ML}} = 0.84$.

This comparison illustrates on one particular example the intimate relation between the mode of the likelihood function and the mode of its approximation in the form of the pseudolikelihood function.

4.6 A Learning within Subclasses of Models

The contribution of this section is a convenient way to compare the performance of the function f in equation (2.9) for automatic image interpretation that is based on different formulations of the global statistical models. Specifically we describe how learning parameters of the internal global conditional probability mass function p in equation (3.2) of the class $\mathcal{P}$ in equation (3.1) can be reduced to learning existing models like the Potts model by confining the learning to the subclasses of models described in Section 3.2.

Let us motivate this part by describing what it would usually mean to compare the performance of the functions f in practice. To test one trainable function for automatic interpretation based on one of the subclasses of the models in practice would more or less mean to formulate and to implement a model function, to formulate and to implement an objective function for the learning and eventually to derive and to implement a gradient function and a Hessian function for an iterative convex optimization method. In Chapter 1.2.2 we have organized models, that the trainable functions for automatic image interpretation are based on, in the literature according to their ability to capture semantic context into seven groups of models. Comparing the functions based on one of the seven groups of models would mean to repeat the sequence of steps described above for each of the groups. Here we propose to do this only once for the class of the most general models, described in Chapter 3, and than for each of the subclasses, described in Section 3.2, of the models to only simply extend the learning problem. Parameters of the models from the subclasses of models described in Section 3.2, like the

Potts model, can be learned by simply adding linear equality constraints of the respective subclasses to the unconstrained strongly convex optimization problem (4.26). We then express the desired model parameter vector $\hat{\boldsymbol{u}}^{\mathrm{MAP}}$,

$$\hat{\boldsymbol{u}}^{\mathrm{MAP}} \in \arg\min_{\boldsymbol{u}} \left\{ \hat{l}_{\mathrm{MAP}}(\boldsymbol{u}) \mid \text{s.t. } \boldsymbol{A}_i \boldsymbol{u} = \mathbf{0} \right\} \tag{4.37}$$

as a solution of the equality constrained strongly convex optimization problem (4.37). In problem (4.37) we express the constraints of the class $\mathcal{P}_i$ from Section 3.2 in the matrix form $\boldsymbol{A}_i \boldsymbol{u} = \mathbf{0}$. Learning parameters of a model from a subclass $\mathcal{P}_i$ then means minimizing the tractable strongly convex objective $\hat{l}_{\mathrm{MAP}}(\boldsymbol{u})$ in the nullspace of the matrix $\boldsymbol{A}_i$, which is the parameter subspace of the model subclass $\mathcal{P}_i$. Convergence rate of the optimization methods proposed in Section 4.4 to solve the problem (4.26) is generally not affected by adding equality constraints in the problem (4.37).

In summary we have described how learning parameters of the internal global statistical model can simply be confined to learning models from the subclasses of models described in Section 3.2. This facilitates the comparison of the performance of the trainable functions for automatic image interpretation that are based on different formulations of the global statistical models.

Chapter 5

Applications of Context Sensitive Image Models

This chapter has two goals. The first goal is to investigate the change in the performance of the trainable function for automatic image interpretation, described in Chapter 2, that is varied first in terms of the model, it is based on, and second in terms of the approximate learning method, it is based on. The second goal is to investigate the potential relevance of this work in the field of 3D city modeling, precision agriculture and medical imaging.

Regarding the first goal we compare the performance of the function that adopts different methods for learning from examples the relation between images and semantic descriptions. More specifically we compare the performance of the function that is based on different methods for learning the parameters of the internal global statistical model that should map images and their semantic descriptions on a high probability value.

Before we approach both of our goals we address in Section 5.1 the computational problem of how should the trainable function for automatic image interpretation, described in Chapter 2, infer semantic descriptions from images. In Section 5.1 this computational challenge translates to the problem (2.17) of computing a mode of a global statistical model from Chapter 3, that the function is based on, and learning of which is described in Chapter 4. In Section 5.1 we describe how in practice the trainable function for automatic image interpretation infers semantic descriptions from images using approximate tractable methods of probabilistic inference forming tractable convex optimization problems.

In Section 5.2 we empirically compare the performance of the function that is based on the pseudolikelihood based learning from Chapter 4 with a function that is based on learning methods reviewed in Section 1.2.3. Regarding further the first goal we compare the performance of the function that is based on different models for relating images and semantic descriptions. More specifically we compare the performance of the function that is based on different global statistical models.

In Section 5.3 we compare the performance of the function that is based on a global statistical model of the class of models proposed in Section 3.1 with a function that is based on a global statistical model from a subclass of models described in Section 3.2. Regarding the second goal we present applications of the trainable function for automatic image interpretation that adopts internally global statistical models from a subclasses of models described in Section 3.2.

In Section 5.4 we present application examples of image patch level object class segmentation for interpreting images of street scenes, pixel level object class segmentation for interpreting multi-spectral images of diseased plant leafs and of voxel level object class segmentation for volumetric human knee images from magnetic resonance.

5.1 Computing Context Sensitive Interpretation

In this section the problem of semantic description inference takes the form of an integer programming problem. For a conditional probability mass function p in equation (3.2) of the class $\mathcal{P}$ in equation (3.1), for a parameter vector $\hat{\boldsymbol{u}}^{\mathrm{MAP}}$ in equation (4.26) estimated from the training set $\mathcal{D}_{\mathcal{M}} = \{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m\in\mathcal{M}}$ in equation (2.7), for an image $\boldsymbol{y}^n$ from the test set $\mathcal{D}_{\mathcal{N}} = \{\boldsymbol{x}^n, \boldsymbol{y}^n\}_{n\in\mathcal{N}}$ in equation (2.8) we are interested in finding a label configuration $\hat{\boldsymbol{x}}^n$ in equation (2.17) that maximizes the conditional probability mass function p or equivalently a label configuration $\hat{\boldsymbol{x}}^n$,

$$\hat{\boldsymbol{x}}^n \in \arg\min_{\boldsymbol{x}\in\mathcal{K}^I} E(\boldsymbol{x} \mid \boldsymbol{y}^n, \hat{\boldsymbol{u}}^{\mathrm{MAP}}) \tag{5.1}$$

that minimizes the energy function in equation (3.7). All of the optimization variables in problem (5.1) are restricted to be integers and we are faced with an instance of the integer programming problems. When referring to the label configuration $\hat{\boldsymbol{x}}^n$, we view the conditional probability mass function p as a posterior probability mass function and also refer to the label configuration $\hat{\boldsymbol{x}}^n$ as a maximum a posteriori probability label configuration.

We solve two simple examples of the above optimization problem. The two following examples border on triviality and we include them for the sake of completeness. They conclude the two series of our examples that throughout this thesis have illuminated the core conceptual and computational problems.

Example 5.1 (Two variable $\mathcal{P}_5$ CRF: MAP labeling)
We now identify the mode of the CRF in example 3.2 by observation. In table 3.3 we observe that the minimum energy value 0.9 *and the maximum probability mass* 0.3396 *correspond to the labeling* $\hat{\boldsymbol{x}}$*, where* $\hat{x}_0 = 1$ *and* $\hat{x}_1 = 0$*, which yields the solution of this simple MAP labeling problem.* $*$

We solve another simple example of the above optimization problem.

Example 5.2 (Two variable $\mathcal{P}$ CRF: MAP labeling)
We find the mode of the CRF in example 3.1 by observation. In table 3.1 we observe that the minimum energy value -1.0 *and the maximum probability mass* 0.7112 *correspond to the labeling* $\hat{\boldsymbol{x}}$*, where* $\hat{x}_0 = 1$ *and* $\hat{x}_1 = 0$*, which yields the solution of this simple MAP labeling problem.* $*$

We have described the problem of inferring the semantic description from an observed image as an integer programming problem. In general, solving the problem (5.1) exactly is intractable. It is thus admissible to adopt an approximate approach, which is our next step.

In this section the problem of semantic description inference takes the form of a linear inequality and equality constrained convex optimization problem. Specifically the problem takes the form of a linear programming problem. Recently, algorithms based on graph cuts, loopy belief propagation and convex relaxation have proven to be efficient in finding good approximate solutions to the problem (5.1). In our experiments we adopt an approach based on convex relaxation. In this approach the original difficult integer programming problem (5.1) is approximated by a convex problem that can be solved efficiently. Convex relaxation approach has proven to be a powerful alternative to the previously mentioned algorithms. Specifically, in our experiments we implement the linear programming relaxation proposed by [Schlesinger, 1976] for a special case and independently by [Chekuri et al., 2001, Koster et al., 1998, Wainwright et al., 2005a] for the general case. In [Wainwright and Jordan, 2008, Kumar et al., 2009] the authors prove that the adopted linear programming relaxation provides better approximation than other convex relaxations proposed recently. For models from the subclass $\mathcal{P}_5$ of the Potts models in equation (3.29) with only two class labels the linear programming relaxation provides an exact solution [Wainwright and Jordan, 2008]. For models from the subclass $\mathcal{P}_5$ of the Potts models in equation (3.29) the linear programming relaxation provides the so called multiplicative bound of two [Chekuri et al., 2001]. To our knowledge the bound for the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1) is unknown. As we explain in Section 1.2.4 linear programs can be solved with the barrier method, a variant of the interior point methods, for which convergence analysis for self-concordant convex functions applies and hence rigorous upper bound on the complexity of obtaining a solution can be evaluated [Boyd and Vandenberghe, 2004]. It is a known fact that linear programming problems can be solved numerically efficiently [Boyd and Vandenberghe, 2004] and we can thus say that the approximation takes the form of a tractable convex optimization problem.

We have described the desired semantic description of an image as an approximate solution of a problem forming a tractable convex optimization problem. This is our third and last component of the trainable function for automatic image interpretation needed the test the function empirically. In the next sections we first investigate the changing performance of the trainable function for automatic image interpretation that is varied in terms of its internal components and afterwards we illustrate the potential relevance of these concepts in applications.

5.2 Comparison of Learning Methods

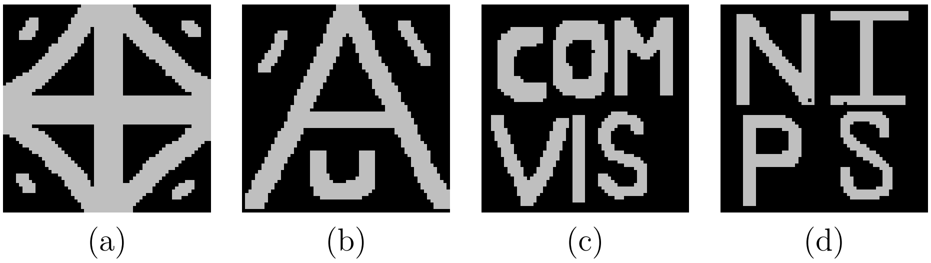
This sections compares the performance of the trainable function f for automatic image interpretation in equation (2.9) that adopts different approximate methods for learning from the training set $\mathcal{D}_{\mathcal{M}} = \{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m \in \mathcal{M}}$ in equation (2.7) the relation between images and semantic descriptions in the test set $\mathcal{D}_{\mathcal{N}} = \{\boldsymbol{x}^n, \boldsymbol{y}^n\}_{n \in \mathcal{N}}$ in equation (2.8) without having access to the test set during the training. More specifically it is our contribution to compare the performance of the function that is based on different approximate methods for learning the parameters $\boldsymbol{u}$ in equation (3.6) of the internal global conditional probability mass function p in equation (3.2) that maps the training images and their semantic descriptions on a high probability value. We empirically compare the performance of the function that attempts to compute the parameters $\boldsymbol{u}$ by approximately solving the problem (4.8). We compare the performance of the pseudolikelihood based approximate learning from Chapter 4 with the performance based on approximate learning methods reviewed in Section 1.2.3.

We repeat learning experiments performed in [Kumar and Hebert, 2004a, 2006, Kumar et al., 2005, Vishwanathan et al., 2006, Korč and Förstner, 2008a,b, Pletscher et al., 2009, Hazan and Urtasun, 2010a,b] and compare our results with results published in these works. In the experiment the function f, that is based on the global conditional probability mass function from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20), is applied to an image restoration task. The aim of these experiments is to compare the overall performance of the function f in terms of a criterion forming the class error $e_{\mathcal{N}}$ in equation (2.10).

We first describe the used dataset and the image features. Afterwards we describe our experiments and state the results that we obtain. Eventually we discuss our results and finish with concluding remarks.

Image Datasets

We consider two synthetic data sets, namely the Gaussian synthetic data set and the bimodal synthetic data set. Each synthetic data set $\mathcal{D} = \left\{\boldsymbol{x}^l, \boldsymbol{y}^l\right\}_{l \in \mathcal{L}}$ in equation (2.5) comprises 200 labeled images $(\boldsymbol{x}^l, \boldsymbol{y}^l)$ 64×64 pixels each that are indexed with the index set $\mathcal{L} = \{1, \ldots, l, \ldots, 200\}$ in equation (2.6). Each synthetic data set $\mathcal{D}$ contains 50 copies $\boldsymbol{x}^l$ of each of the 4 ground truth label configurations shown in figure 5.1(a-d). Labels in the original dataset employed in [Kumar and Hebert, 2006] take values in the set $\{-1, +1\}$[1]. Images $\boldsymbol{y}^l$ in each of the data sets have been artificially generated by copying the ground truth label configurations $\boldsymbol{x}^l$ and adding random noise to each pixel value $x_i^l \in \{-1, 1\} \subset \mathbb{R}$ independently. This resulted in pixel values $y_i^l \in \mathbb{R}$ in a larger range. Each data set originates from adding different type of random noise to the pixel values. Gaussian synthetic data set originates from adding a Gaussian noise, see figure 5.2, that is the resulting observations are real values with potentially large values. In figure 5.3(a) we show the



(a) (b) (c) (d)

Figure 5.1: *Synthetic data sets: Ground truth.* (a-d) Ground truth label configurations present in synthetic data sets, where single pixel class label takes on the value 1 or the value 0. We show 1s in gray and 0s in black.

[1] Since we in this work are generally interested in the multi-class set $\{0, \ldots, K-1\}$ we let the label variables in figure 5.1 and in our experiments take values in the set $\{0, 1\}$. Label -1 in the original dataset has the value 1 in our dataset.

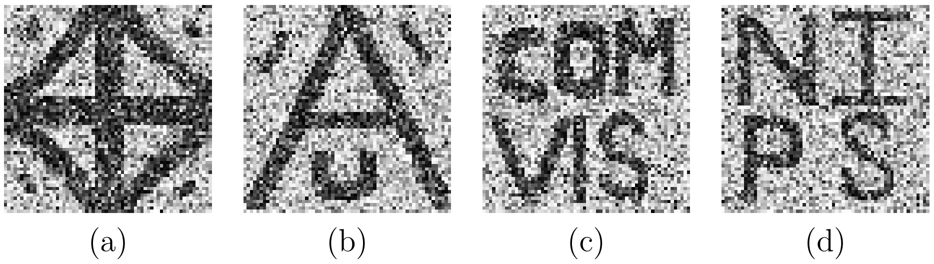
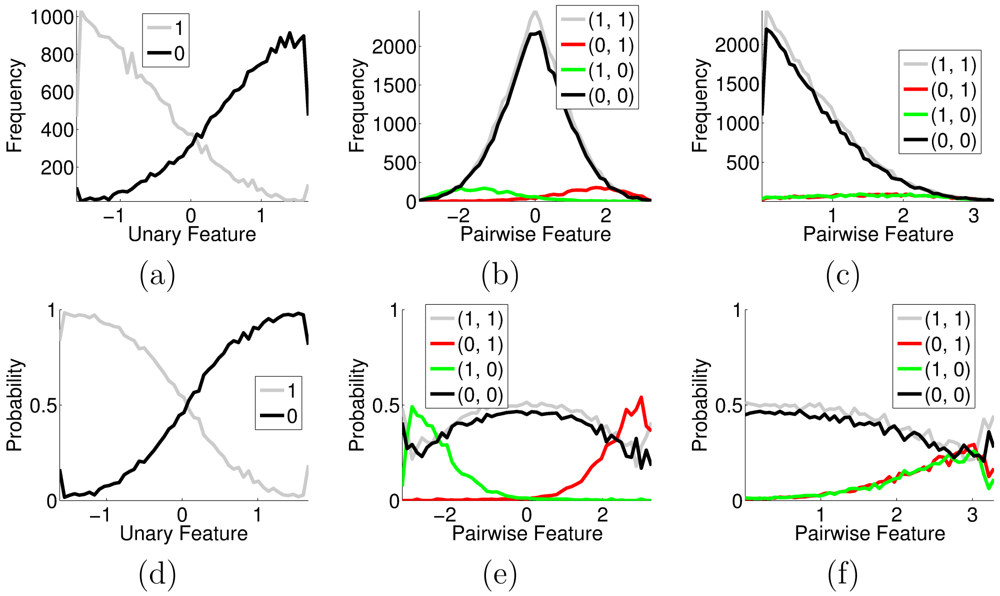

(a) (b) (c) (d)

Figure 5.2: *Gaussian synthetic data set: Image data.* (a-d) Images corresponding respectively to label configurations in figure 5.1(a-d). In each image (a-d) we show the minimum pixel value in black and the maximum pixel value in white. This means that single color can represent significantly different values in the four cases. The visualization has no effect on the experiment of course.



(a) (b) (c)
(d) (e) (f)

Figure 5.3: *Gaussian synthetic data set: Feature frequencies.* Single point on a function graph shows: (a) Class-specific frequency of an unary feature value in the training data, (b) class-pair-specific frequency of a pairwise feature value in the training data, (c) class-pair-specific frequency of a pairwise feature absolute value in the training data, (d) unary feature value conditional class probability computed from the frequency in (a), (e) pairwise feature value conditional class pair probability computed from the frequency in (b) and (f) pairwise feature absolute value conditional class pair probability computed from the frequency in (c).

class-specific frequencies of the pixel intensity values. The bimodal synthetic data set originates from adding a label dependent noise in the form of a mixture of two Gaussian distributions, see figure 5.4. In figure 5.5(a) we show

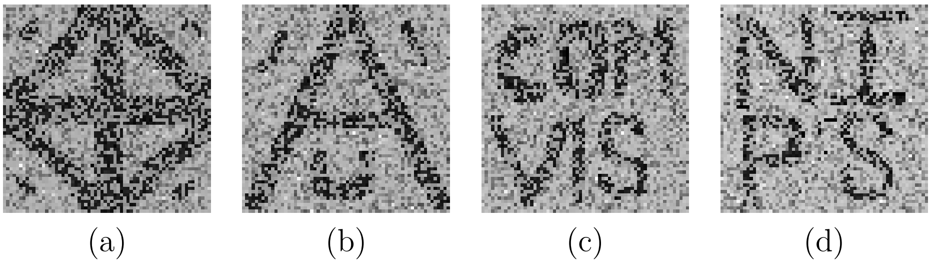


Figure 5.4: *Bimodal synthetic data set: Image data.* (a-d) Images corresponding respectively to label configurations in figure 5.1(a-d). In each image (a-d) we show the minimum pixel value in black and the maximum pixel value in white.

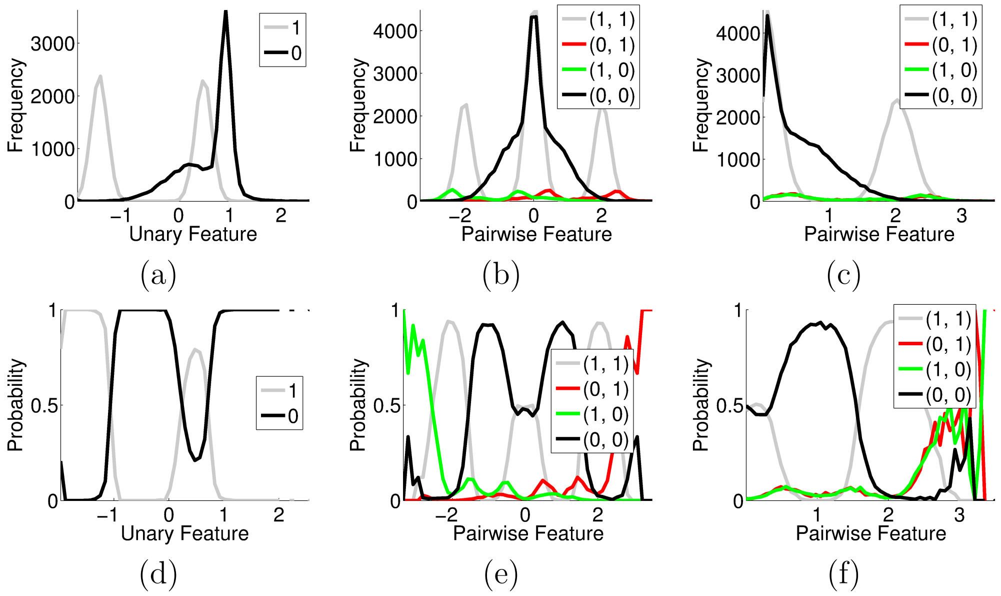
the class-specific frequencies of the pixel intensity values. figure 5.3(a) and figure 5.5(a) illustrate how informative are the single pixel intensity values for a local classifier. We include the noise model parameters given in [Kumar and Hebert, 2006]. For the Gaussian synthetic dataset the authors report an independent Gaussian noise of standard deviation 0.3. For the bimodal synthetic data set the authors report that the mixture model parameters (mean, std) for the two classes were chosen to be $[(0.08, 0.03), (0.46, 0.03)]$, and $[(0.55, 0.02), (0.42, 0.10)]$.

Image Features

Here we describe the unary and pairwise image features. We want to illustrate how informative these features are for the unary and pairwise energy terms. For each image $\boldsymbol{y}^l$ from each dataset $\mathcal{D}$ in Section 5.2 we compute the following image features. We set unary feature values $\mathbf{h}_i^l(\mathbf{y})$ in equation (3.10),

$$\mathbf{h}_i^l(\mathbf{y}) = [1, y_i^l]^T \tag{5.2}$$

to pixel values y_i^l. Unary features values for the Gaussian synthetic data set are shown in figure 5.2 and in figure 5.3(a)(d). Unary feature values for the bimodal synthetic data set are shown in figure 5.4 and in figure 5.5(a)(d). Figure 5.3(a)(d) and figure 5.5(a)(d) show the class-specific frequencies of the unary feature values in the training data, which illustrates how informative the unary features are for the unary terms. We compute horizontal and



(a) (b) (c)

(d) (e) (f)

Figure 5.5: *Bimodal synthetic data set: Feature frequencies.* Single point on a function graph shows: (a) Class-specific frequency of an unary feature value in the training data, (b) class-pair-specific frequency of a pairwise feature value in the training data, (c) class-pair-specific frequency of a pairwise feature absolute value in the training data, (d) unary feature value conditional class probability computed from frequency in (a), (e) pairwise feature value conditional class pair probability computed from frequency in (b) and (f) pairwise feature absolute value conditional class pair probability computed from frequency in (c).

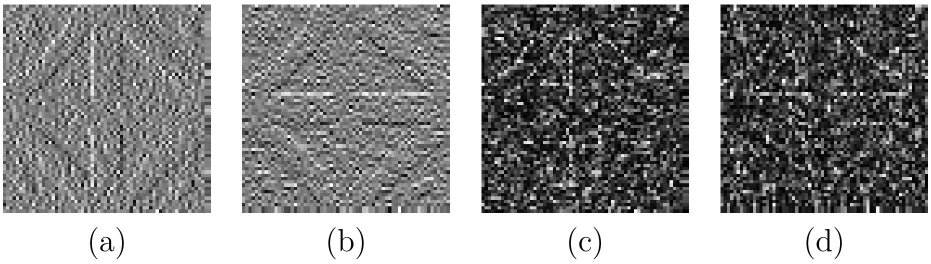
vertical pairwise image feature values $\boldsymbol{\mu}^l_{ii'}(\mathbf{y})$ in equation (3.13),

$$\boldsymbol{\mu}^l_{ii'}(\mathbf{y}) = [1, |y^l_i - y^l_{i'}|]^T \tag{5.3}$$

as absolute value of difference of two neighboring pixel values y^l_i and $y^l_{i'}$. Pairwise feature values for the Gaussian synthetic data set are shown in figure 5.6(c)(d) and in figure 5.3(c)(f). Pairwise feature values for the bimodal synthetic data set are shown in figure 5.7(c)(d) and in figure 5.5(c)(f). Figure 5.3(c)(f) and figure 5.5(c)(f) show class-pair-specific frequencies of the pairwise feature values in the training data, which illustrates how informative the pairwise features are for the pairwise terms.

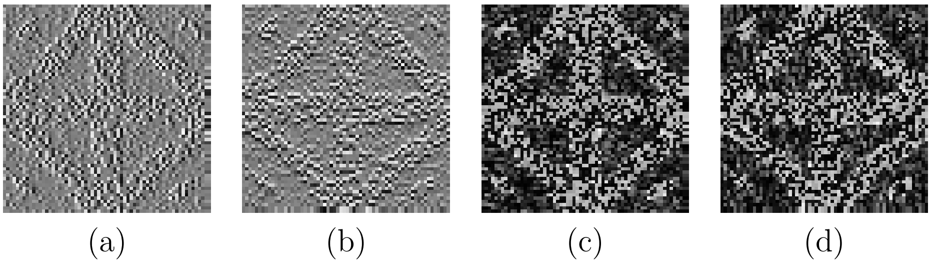
Experiment

We now describe the training data, the test data and subsequently we detail the learning and inference procedure in our experiment. In each of the



(a) (b) (c) (d)

Figure 5.6: *Gaussian synthetic data set: Pairwise image features.* (a) Horizontal and (b) vertical pairwise feature values. (c) Horizontal and (d) vertical pairwise feature absolute values. In each image (a-d) we show the minimum pixel value in black and the maximum pixel value in white.



(a) (b) (c) (d)

Figure 5.7: *Bimodal synthetic data set: Pairwise image features.* (a) Horizontal and (b) vertical pairwise feature values. (c) Horizontal and (d) vertical pairwise feature absolute values. In each image (a-d) we show the minimum pixel value in black and the maximum pixel value in white.

10 runs of the experiment we partition the data set $\mathcal{D}$ described in Section 5.2 anew into a training set $\mathcal{D}_{\mathcal{M}}$ in equation (2.7) and a test set $\mathcal{D}_{\mathcal{N}}$ in equation (2.8). The training set $\mathcal{D}_{\mathcal{M}}$, $\mathcal{D}_{\mathcal{M}} = \{\boldsymbol{x}^m, \boldsymbol{y}^m\}_{m \in \mathcal{M}}$, comprises 10 labeled images $(\boldsymbol{x}^m, \boldsymbol{y}^m)$ of the type shown in figure 5.1(a). The test set $\mathcal{D}_{\mathcal{N}}$, $\mathcal{D}_{\mathcal{N}} = \{\boldsymbol{x}^n, \boldsymbol{y}^n\}_{n \in \mathcal{N}}$, comprises the rest 190 labeled images $(\boldsymbol{x}^n, \boldsymbol{y}^n)$ of the types shown shown in figure 5.1(a-d).

In our experiments we in accordance with the findings in our previous work [Korč and Förstner, 2008a] do not impose any prior on the unary parameters $\boldsymbol{w}$ in equation (3.8) and adopt Gaussian prior on the pairwise parameters $\boldsymbol{v}$ in equation (3.11). We refer to the problem (4.26) with such prior as to the penalized pseudolikelihood (PPL) learning. In each run of the experiment the function f in equation (2.9) employs the pseudolikelihood (PL) learning from Section 4.3.1 and the penalized pseudolikelihood (PPL) learning to internally estimate parameters of a global conditional probability mass function p from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20). The function f then as described in Section 5.1 infers the maximum a posteriori (MAP) label configurations $\hat{\boldsymbol{x}}^n$ from the test images $\boldsymbol{y}^n$ in the test set $\mathcal{D}_{\mathcal{N}} = \{\boldsymbol{x}^n, \boldsymbol{y}^n\}_{n \in \mathcal{N}}$. We evaluate the test pixel class error $e_{\mathcal{N}}$ in equation (2.10) with respect to the test label configurations $\boldsymbol{x}^n$ in the test set in each run and report the mean and the standard deviation across the 10 runs.

Results

We summarize the results of our comparison of the learning methods in table 5.1. In this experiment we consider models from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20) and hence we consider rows 1-18 in table 5.1.

In table 5.1 we provide results of the learning methods described in Section 1.2.3. We provide results of a learning method based on a stochastic approximation of the likelihood gradient, namely

- the contrastive divergence (CD) method [Hinton, 2002],

results of learning methods based on deterministic approximations of the likelihood gradient that include

- the saddle point approximation (SPA) method [Geiger and Girosi, 1991],
- the maximum marginal approximation (MMA) method [Kumar et al., 2005] and
- the pseudo-marginal approximation (PMA) method [McCallum et al., 2003]

and in particular we provide results of learning methods based on convex surrogate likelihood including

- the pseudolikelihood (PL) method [Besag, 1975],
- the penalized pseudolikelihood (PPL) method [Kumar et al., 2005, Korč and Förstner, 2008a] and
- the spanning tree approximation (STA) method [Pletscher et al., 2009].

Next to these approximations we provide results of

- a mean field (MF) approximation based learning method [Wainwright and Jordan, 2008]

Table 5.1: *Comparison of models and learning methods.* Pixel class errors (%) on 190 test images for our results. Rows show combinations of a model class and a learning method. See text in Section 5.2 for an explanation. Columns show two synthetic image data sets. Learning time is reported where available. KF08 stands for our results published in [Korč and Förstner, 2008b]. KH04 stands for the results published in [Kumar and Hebert, 2004a]. K05 stands for the results published in [Kumar et al., 2005]. KH06 stands for the results published in [Kumar and Hebert, 2006]. V06 stands for the results published in [Vishwanathan et al., 2006]. P09 stands for the results published in [Pletscher et al., 2009]. HU10 stands for the results published in [Hazan and Urtasun, 2010b,a]. Means ± standard deviations over 10 experiments are given for our results. Rows are sorted according to the error for the bimodal synthetic data set.

Row	Model class	Learn. approx.	Gaussian synthetic data set	Bimodal synthetic data set	Learn. time (sec)	Source
1	$\mathcal{P}_2$	PPL	–	$5 < e_{\mathcal{N}} < 10$	–	P09
2	$(\mathcal{P}_2)$	PL	“failed”	“failed”	–	HU10
3	$\mathcal{P}_2$	PL	“failed”	–	–	V06
4	$\mathcal{P}_2$	PL	29.49	29.49	–	KH04
5	$\mathcal{P}_2$	PL	3.82	17.69	–	KH06
6	$\mathcal{P}_2$	PL	–	10 (∼)	–	P09
7	$\mathcal{P}_2$	PL	3.10	7.31	300	K05
8	$\mathcal{P}_2$	MF	3 (∼)	–	–	V06
9	$\mathcal{P}_2$	CD	2.82	6.29	207	K05
10	$\mathcal{P}_2$	PPL	2.30	6.21	–	KH04
11	$\mathcal{P}_2$	PPL	2.30	6.21	–	KH06
12	$\mathcal{P}_2$	SPA	2.49	5.82	82	K05
13	$\mathcal{P}_2$	MMA	2.96	5.70	636	K05
14	$\mathcal{P}_2$	LBP	2.7 (∼)	–	–	V06
15	$\mathcal{P}_2$	PL	2.55 ± 0.02	5.68 ± 0.05	42 ± 7	KF08
16	$\mathcal{P}_2$	PPL	2.54 ± 0.02	5.64 ± 0.04	45 ± 8	KF08
17	$\mathcal{P}_2$	PMA	2.51	5.48	1183	K05
18	$\mathcal{P}_2$	STA	$\mathbf{2.33} \pm 0.01$	$\mathbf{5.23} \pm 0.01$	–	P09
19	$\mathcal{P}$	PL	4.81 ± 2.59	1.98 ± 0.43	< 300	our
20	$\mathcal{P}$	PPL	2.43 ± 0.05	$\mathbf{1.58} \pm 0.07$	< 300	our

and a learning method based on

- the loopy belief propagation (LBP) approximation of the Bethe problem [Ganapathi et al., 2008].

Results in table 5.1 are based on inference criteria that include the maximum a posterior (MAP) probability and the maximum posterior marginal (MPM) probability. We report results based on the MAP inference criterion for our results and the better performing inference criterion otherwise.

In table 5.1 we report the test pixel class error on the Gaussian synthetic data set and the bimodal synthetic data set, both described in Section 5.2. Rows in table 5.1 are sorted according to the error for the bimodal synthetic data set. For our results we report the mean and the standard deviation of the test pixel class error over 10 runs. Results published in [Pletscher et al., 2009] include the mean and the standard deviation over 5 runs.

In table 5.1 we report the learning time when available. For our results published in [Korč and Förstner, 2008b] we report the mean and the standard deviation over 10 runs.

KF08 in table 5.1 stands for our results published in [Korč and Förstner, 2008b], which is a revised version of our work published in [Korč and Förstner, 2008a]. KH04 stands for the results published in [Kumar and Hebert, 2004a]. K05 stands for the results published in [Kumar et al., 2005]. KH06 stands for the results published in [Kumar and Hebert, 2006]. V06 stands for the results published in [Vishwanathan et al., 2006]. In [Vishwanathan et al., 2006] the authors only report results for the Gaussian synthetic data set. The authors also only report a qualitative result for PL method, where the authors phrase it as a complete failure. For the LBP method and the MF method they only provide the test pixel class error curves, from which we estimate the error values that we include in table 5.1. P09 stands for the results published in [Pletscher et al., 2009]. In [Pletscher et al., 2009] for the PL method and the PPL method the authors only report error curves, from which we estimate the error values that we include in table 5.1. Their test pixel class error of the PPL method is slightly above 5%, however more precise value is not obvious to us from their log scale axes and hence in table 5.1 we only provide rough estimate of the value. HU10 stands for the results published in [Hazan and Urtasun, 2010b,a]. In [Hazan and Urtasun, 2010b,a] the authors only report a qualitative result for a distinct model though with pairwise interaction of a model from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models.

Discussion

Let us first restrict ourselves to the PL method and consider the rows 2-7,15 in table 5.1. In table 5.1 we observe that our results published in the revised version [Korč and Förstner, 2008b] of our work in [Korč and Förstner, 2008a] (row 15 in table 5.1) yield the lowest test pixel class error among the published results that employ the PL method. Here we also point to the low standard deviation of our test pixel class error over 10 runs of the experiment. Our results are in contrast to the results of the PL learning published in [Kumar and Hebert, 2004a, 2006, Kumar et al., 2005, Vishwanathan et al., 2006, Pletscher et al., 2009, Hazan and Urtasun, 2010a,b], where the reported test pixel class errors are significantly higher (row 2-7 in table 5.1).

Let us now restrict ourselves to the PPL method and consider the rows 1, 10-11, 16 in table 5.1. In table 5.1 we observe that our results published in [Korč and Förstner, 2008b] (row 16) yield the lowest test pixel class error. Here we again point to the low standard deviation of our test pixel class error over 10 runs of the experiment. Again results are in contrast to the results of the PPL learning published in [Kumar and Hebert, 2004a, 2006] and reporting higher class errors (rows 1, 10-11).

Let us now compare the PL learning and the PPL learning with the other methods and consider the rows 1-18 in table 5.1. In table 5.1 we observe that the STA method published in [Pletscher et al., 2009] (row 18 in table 5.1) yields the lowest test pixel class error and that with low standard deviation over 5 runs. This result is followed by the PMA method, for which the authors in [Kumar et al., 2005] (row 17 in table 5.1) only provide a result of a single run. These two results are followed by our results with the PPL learning (row 16) and with the PL learning (row 15) that we report to have low standard deviation. In table 5.1 we observe that both the PL learning and the PPL learning yield competitive results with respect to the other methods and that while providing low complexity and advantages to formulating parameter learning as a (strongly) convex optimization problem. Let us note that what the STA method and the PL and the PPL methods have in common is that they both are based on a convex surrogate likelihood. In these cases, the learning problem can be solved, reliably and efficiently, drawing upon the benefits of readily available methods for convex optimization. Our findings are in contrast with the comparison published in [Kumar et al., 2005], where the PL learning is claimed to be the worst performing learning method as compared to the SPA, PMA, MMA, and CD methods.

Conclusion

We conclude that the pseudolikelihood based learning, discussed in the comparison in the previous section, yields competitive results. Our findings counterbalance the in the literature widely adopted view that the pseudolikelihood based learning performs poorly or fails completely. From a broader perspective we conclude that it is in our comparison the wider class of approximate learning methods, based on tractable convex surrogate likelihoods and including the pseudolikelihood and the spanning tree approximation [Pletscher et al., 2009], that yields competitive results.

Let us now consider the comparison of the class of the approximate learning methods based on the tractable convex surrogate likelihoods and the class of the approximate learning methods based on the deterministic approximations of the likelihood gradient. Based on the comparison published in [Kumar et al., 2005] it would be sensible to draw a conclusion that learning methods based on deterministic approximations of the likelihood gradient yield more accurate results. In terms of the results that we obtain in our comparison we however consider both approaches to be competitive and thus to represent valid avenues when approaching new better approximations. Taking into account the heuristic nature and lack of convergence guarantees of the class of the approximate learning methods based on the deterministic approximations of the likelihood gradient, we are inclined to favor the class of the approximate learning methods based on the tractable convex surrogate likelihoods.

Let us maintain that we have drawn our conclusions from the provided and still rather limited experiments. Further experiments are needed to further support our statements.

5.3 Comparison of Global Models

This section compares performance of the trainable function f for automatic image interpretation in equation (2.9) that is varied in terms of models, that it is based on, to relate images and semantic descriptions. More specifically it is our contribution to provide a comparison of the performance of the function f, that is based on the global statistical models from the in the literature rarely adopted class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1), with the performance of the function f, that is based on models from the in the literature widely adopted subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20).

Image Datasets

In this experiment we use the Gaussian synthetic data set and the bimodal synthetic data set, both described in Section 5.2.

Image Features

For each image $\boldsymbol{y}^l$ from each dataset $\mathcal{D}$ in Section 5.2 we compute the following image features. We first compute unary feature values $\mathbf{h}_i^l(\mathbf{y})$ in equation (5.2). Unary features values for the Gaussian synthetic data set are shown in figure 5.2 and in figure 5.3(a)(d). Unary feature values for the bimodal synthetic data set are shown in figure 5.4 and in figure 5.5(a)(d). We compute horizontal and vertical pairwise image feature values $\boldsymbol{\mu}_{ii'}^{\prime l}(\mathbf{y})$ in equation (3.13),

$$\boldsymbol{\mu}_{ii'}^{\prime l}(\mathbf{y}) = [1, y_i^l - y_{i'}^l]^T \tag{5.4}$$

as the difference of two neighboring pixel values y_i^l and $y_{i'}^l$. We point out the difference between the pairwise image features in equation (5.4) and the pairwise image features in equation (5.3). Pairwise feature values for the Gaussian synthetic data set are shown in figure 5.6(a)(b) and in figure 5.3(b)(e). We observe that the pairwise features capture the orientation of edges in figure 5.6(a)(b), which can be exploited by a model with asymmetric label interactions. On the contrary the pairwise features in figure 5.6(c)(d) do not retain the edge orientation in data, which anyhow cannot be exploited by a model with symmetric label interactions. Pairwise feature values for the bimodal synthetic data set are shown in figure 5.7(a)(b) and in figure 5.5(b)(e). figure 5.3(b)(e) and figure 5.5(b)(e) show class-pair-specific frequencies of the pairwise feature values in the training data, which illustrates how informative the pairwise features are for the pairwise terms. We use quadratic (i.e. non-linear) feature mapping to project the unary feature vectors and the pairwise feature vectors $\boldsymbol{\mu}_{ii'}^{\prime l}(\mathbf{y})$ in higher dimension, which in this particular case results simply in unary image feature vectors $\mathbf{h}_i^l(\mathbf{y})$,

$$\mathbf{h}_i^l(\mathbf{y}) = [1, y_i^l, (y_i^l)^2, (y_i^l)^3, (y_i^l)^4]^T$$

and in pairwise image feature vectors $\boldsymbol{\mu}_{ii'}^l(\mathbf{y})$,

$$\boldsymbol{\mu}_{ii'}^l(\mathbf{y}) = [1, y_i^l - y_{i'}^l, (y_i^l - y_{i'}^l)^2, (y_i^l - y_{i'}^l)^3, (y_i^l - y_{i'}^l)^4]^T$$

that we employ in our experiment.

Experiment

We repeat the experiment in Section 5.2 with the only difference that in each run of the experiment the function f in equation (2.9) now learns internally parameters of a global conditional probability mass function of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1).

Results

We summarize our comparison of the models in table 5.1. In this experiment we consider both the models of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1) and the models from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20) and hence we now consider all rows 1-20 in table 5.1. The rows are denoted by the class $\mathcal{P}$ and by the subclass $\mathcal{P}_2$.

In table 5.1 we report the test pixel class error on the Gaussian synthetic data set and the bimodal synthetic data set, described in Section 5.2. We report the mean and the standard deviation of the test pixel class error over 10 runs. We do not optimize our implementation of the optimization algorithms for speed and do not report learning time in this experiment. The optimization takes up to 300 seconds with non-optimized code.

Discussion

We now discuss the comparison of the performance of the function f in equation (2.9) that is based on a model of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1) with the performance of the function f that is based on a model from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20) that is also employed in [Kumar and Hebert, 2004a, 2006, Kumar et al., 2005, Vishwanathan et al., 2006, Korč and Förstner, 2008a,b, Pletscher et al., 2009, Hazan and Urtasun, 2010a,b].

Let us first restrict our comparison to the pseudolikelihood based (PL, PPL) learning and consider rows 1-7,10-11,15-16,19-20 in table 5.1. In table 5.1 we observe that the model of the class $\mathcal{P}$ in the rows 19-20 significantly lowers the test pixel class error in this case. Further for the class $\mathcal{P}$ in the rows 19-20 there is a notable improvement in the case of the PPL method in the row 20 as compared to the PL method in the row 19. For the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models the difference between the PL method in the row 15 and the PPL method in the row 16 is insignificant.

Let us now compare the performance of the function f, that is based on a model of the class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1) with the pseudolikelihood based learning (rows 19-20 in table 5.1), with the performance of the function f that is based on a model from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models with the best performing learning (row 18 in table 5.1). In table 5.1 we observe that a model of the class $\mathcal{P}$ with the pseudolikelihood based learning notably improves the performance of a simpler model from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models even in case where the model from the subclass $\mathcal{P}_2$ is employed with the best performing learning method.

Conclusion

We find in our comparison that the performance of the trainable function f for automatic image interpretation in equation (2.9) can notably be improved by being based on the global conditional probability mass functions from the in the literature rarely adopted class $\mathcal{P}$ of the data-dependent asymmetric interaction models in equation (3.1) as opposed to being based on a model from the in the literature widely adopted subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20).

5.4 Applications

The goal of this section is to illustrate the relevance of trainable functions for automatic image interpretation in applications. Here no attempt is made to empirically prove or disprove the chosen models. Our only aim at this place is to demonstrate the sole applicability. We present application examples of image patch level object class segmentation for interpreting images of street scenes, of pixel level object class segmentation for interpreting multi-spectral images of diseased plant leafs and of voxel level object class segmentation for interpreting volumetric human knee images from magnetic resonance.

5.4.1 Detecting Man-Made Objects in Natural Images

In our work in [Korč and Förstner, 2008c] we apply the function f in equation (2.9) to the task of detecting man-made objects in urban street scene images. These are RGB images taken with a consumer camera, 8 bit/color.

As we report in [Korč and Förstner, 2008c] we divide our test images, each of size 384×256 pixels, into non-overlapping blocks, each of size 16×16 pixels, that we form our image sites. For each image site, a 2-dimensional single-site gradient magnitude and orientation based feature vector is computed. Similar features are employed in [Yang et al., 2010]. In [Korč and Förstner, 2008c] we use linear discriminant and quadratic feature mapping to design the potential functions of the conditional random field.

Here we vary the function f in terms of being based on a model from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20), the subclass $\mathcal{P}_5$ of the Potts models in equation (3.29) and the subclass $\mathcal{P}_6$ of the local classifiers in equation (3.31). We train the function f with the PPL learning from Section 4.3.2 and we infer the MAP object class label configurations as explained in [Korč and Förstner, 2008c]. Models from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models provide a theoretically well founded approach for combining local discriminative classifiers that allow the use of arbitrary overlapping features, with adaptive data-dependent smoothing over the label field. Figure 5.8 illustrates improved detection rate and reduced false positive rate of the function f based on a model from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models as compared to the function f that is based on a model from the subclass $\mathcal{P}_5$ of the Potts models or from the subclass $\mathcal{P}_6$ of the local classifiers.

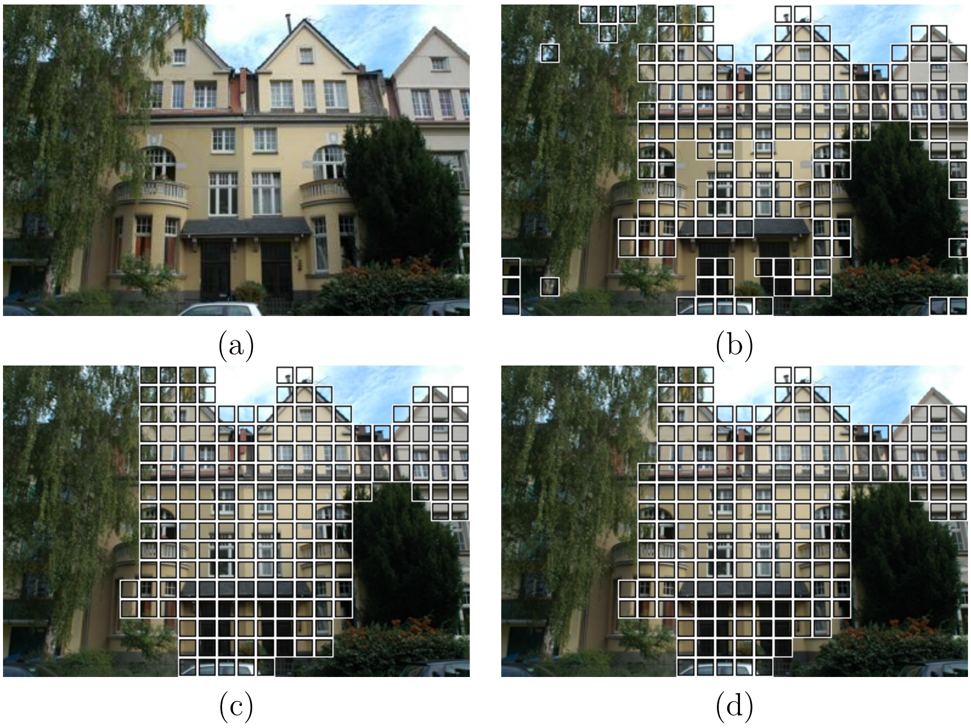


Figure 5.8: *Interpreted image of a street scene.* (a) Input image of a street scene. (b) Automatic image patch level man-made structure class segmentation based on a model from the subclass $\mathcal{P}_6$ of the local classifiers in equation (3.31). (c) Automatic image patch level man-made structure class segmentation based on a model from the subclass $\mathcal{P}_5$ of the Potts models in equation (3.29). (d) Automatic image patch level man-made structure class segmentation based on a model from the subclass $\mathcal{P}_2$ of the contrast sensitive Potts models in equation (3.20). Man-made structure class labels are denoted by bounding boxes superimposed on the input image.

5.4.2 Interpreting Images of Diseased Plant Leafs

In our work in [Bauer, Korč, and Förstner, 2011] we apply the function f in equation (2.9) to the task of interpreting images of diseased plant leafs.

To construct image feature vectors in our experiments we combined measurements from RGB and multi-spectral images. In the first set of our experiments we combined intensity values from the three channels of RGB images with the intensity values from the infrared channel of multi-spectral images into 4-dimensional image feature vectors that we associated with each pixel location. In the second set of our experiments we used image feature vectors that we extract from overlapping neighborhoods. For each pixel location we concatenated the previously described feature vector with the feature vectors from the four neighboring pixels and obtained thus for each pixel location a 20-dimensional feature vector.

In this work we vary the function f in terms of being based on models from the subclass $\mathcal{P}_4$ of the generalized Potts models in equation (3.25), the subclass $\mathcal{P}_5$ of the Potts models in equation (3.29) and the subclass $\mathcal{P}_6$ of the local classifiers in equation (3.31). We train the function f with the PPL learning from Section 4.3.2 and we infer object class label configurations using the linear programming relaxation described in Section 5.1. In this work we show that the function f that is based on a model from the subclass $\mathcal{P}_4$ of the generalized Potts models learns from the training data a label pair specific interaction that improves results of the function f that is based on a model from the subclass $\mathcal{P}_5$ of the Potts models or a model from the subclass $\mathcal{P}_6$ of the local classifiers. Figure 5.9 illustrates the improved performance of the function f that is based on a model from the subclass $\mathcal{P}_4$ of the generalized Potts models as compared to the performance of the function f that is based on a model from the subclass $\mathcal{P}_6$ of the local classifiers.

As we summarized in [Bauer, Korč, and Förstner, 2011], the enhancement of the feature vector with neighborhood information had a beneficial effect on the classification rates. Our results show the typical failures of a pixelwise classifier, that is isolated pixels are misclassified and neighboring pixels have been allocated to different classes, although they belong to the same class. Our limited experiments with CRFs suggest that modeling class neighborhood in a global probabilistic model is a feasible approach to eliminate the artifacts of pixelwise classification.

5.4.3 Segmenting Cartilage and Bone in MRI Data

In our work [Korč, Schneider, and Förstner, 2010] we presented a fast automatic method for three-dimensional voxel level object class segmentation of

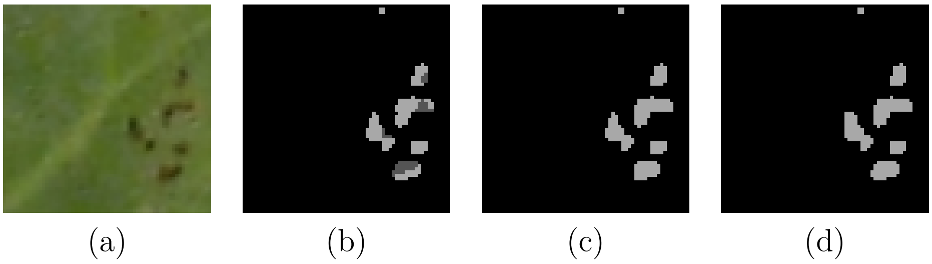
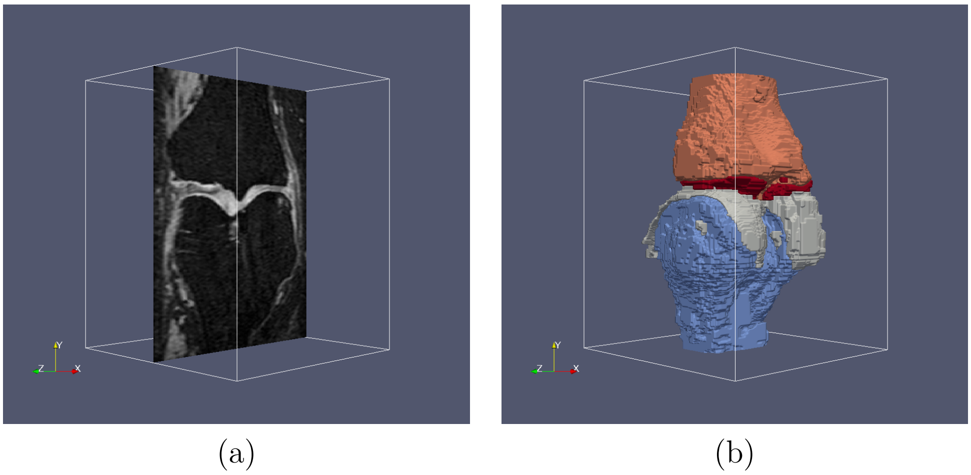


Figure 5.9: *Interpreted multi-spectral image of a diseased sugar beet leaf.* (a) Input image of a diseased sugar beet leaf. (b) Automatic pixel level object class segmentation based on a model from the subclass $\mathcal{P}_6$ of the local classifiers in equation (3.31), showing *Cercospora beticola, Uromyces betae* and *healthy leaf* class labels. (c) Automatic pixel level object class segmentation based on a model from the subclass $\mathcal{P}_4$ of the generalized Potts models in equation (3.25). Parameters of both models have been learned automatically from the training data. (d) Manual pixel level object class segmentation. Example of results published in [Bauer, Korč, and Förstner, 2011].

magnetic resonance images of the knee.

As we explain in [Korč, Schneider, and Förstner, 2010], the training data comprised 60 magnetic resonance images and 60 corresponding manual segmentations. Test data consisted of 40 magnetic resonance images. Image size ranged from 8 to 16 Mio voxels, where typical size was 11 Mio voxels. Typical image resolution was roughly 300 × 350 × 100 voxels. Image spatial resolution was 0.4 × 0.4 × 1 mm. Hence, typical image volume size was roughly $120 \times 140 \times 100$ mm. In our experiments, we normalized the intensity values, in each image individually to $[0, 1]$ by finding the maximum intensity and by dividing each intensity by this value. We did not correct the bias by subtracting the minimum intensity from each intensity value. We used intensity values as our image features.

We formulated a simple global statistical model equivalent to a model from the subclass $\mathcal{P}_5$ of the Potts models in equation (3.29) that allows to jointly segment all classes. The model estimation was performed automatically, though in a different way than described in this work. The adopted linear programming based inference was approximate version of the linear programming inference adopted in experiments in Section 5.2. The voxel level object class segmentation of a magnetic resonance image with 11 Mio voxels took approximately one minute. Figure 5.10(a) shows two-dimensional slice of a three-dimensional (volumetric) test magnetic resonance image shown in the outlined three-dimensional magnetic resonance image volume. Fig-



(a) (b)

Figure 5.10: *Interpreted magnetic resonance image.* (a) two-dimensional slice of a three-dimensional (volumetric) test magnetic resonance image shown in the outlined three-dimensional magnetic resonance image volume. (b) Automatic voxel level object class segmentation of the test magnetic resonance image in (a), where we show the corresponding three-dimensional view, showing *femur, femoral cartilage, tibia, tibial cartilage* and transparent *background* class labels. Example from [Korč, Schneider, and Förstner, 2010], data from [Heimann et al., 2010].

ure 5.10(b) shows automatic voxel level object class segmentation of the test magnetic resonance image in figure 5.10(a), where we show the corresponding three-dimensional view. We note that context insensitive approach based on local classifiers lead to complete failure in these experiments.

Our three application examples illustrate broad applicability of the presented trainable functions for automatic image interpretation. We conclude that in principle modeling context using the global statistical models in the presented application leads to improvement as compared to context insensitive local classifiers. Further we conclude that more sophisticated ways of modeling context lead to further improvement. We note that the success of the methods presented here depends to a large extent on the chosen image features. Finding expressive features for a specific application may be in practice as important as formulating an effective global model.

Chapter 6

Conclusion

We developed a trainable function for automatic image interpretation. The function is capable of learning functional dependencies between irregular arrangements of measurements in space and irregular configurations of semantic labels. The function is based on state-of-the-art models that possess general conceptual properties of being statistical, discriminative, data context sensitive, semantic context sensitive in the data dependent manner and trainable. The function is based on learning and inference mechanisms that possess the general computational properties of being tractable in the sense of being based on tractable convex optimization problems. Next to the general properties the function is further based on in the literature rarely adopted models that possess the specific conceptual property of being semantic context sensitive in an asymmetric manner.

In the structured prediction literature we have identified the conditional Markov random fields as a family of models that posses the desired conceptual properties. In the following we summarize the contributions of this thesis.

- We summarize the state-of-the-art conditional random field models in a class of the in the literature rarely adopted data-dependent label-pair-specific and asymmetric interaction models. We explain how the in the literature widely adopted simpler state-of-the-art models can be obtained as its subclasses.

- We show that conditional Markov random field models with log-potential functions parametrized as affine functions are a simpler equivalent form of the models with log-potential functions parametrized as popular multi-class logistic regression models for classification.

- We identify a consistent approximation of the intractable maximum likelihood learning and that in the form of a tractable strongly convex optimization problem. The approximation is based on the pseudolikelihood function [Besag, 1975], a special case of the the composite likelihood [Lindsay, 1988]. We provide the first partial derivative equations of the pseudolikelihood based learning objective needed to compute the solution with efficient algorithms of convex optimization. To the best of our knowledge the partial derivative equations have not been previously provided in the literature reporting the pseudolikelihood based learning experiments with specific image models.

- We counterbalance reported statements that pseudolikelihood based learning approaches yield unsatisfactory results by providing a comparison with state-of-the-art methods, where the pseudolikelihood based learning yields competitive results. Specifically we compare the performance of the developed trainable function for automatic image interpretation that is varied in terms of its internal approximate learning method. Our competitive results with the pseudolikelihood based learning counterbalance the in the literature widely adopted view that the pseudolikelihood based learning performs poorly or fails completely. On a more abstract level we observe in our comparison that it is the class of approximate learning methods based on tractable convex surrogate likelihoods that yields state-of-the-art results and that in terms of convergence properties represents an appealing alternative to the class of the approximate learning methods based on the deterministic approximations of the likelihood gradient that yields competitive results however generally lacks convergence guarantees. Our conclusion counterbalances the previously published comparison that was in favor of the class of the approximate learning methods based on the deterministic approximations of the likelihood gradient. In terms of the results that we obtain in our comparison we however consider both approaches to represent valid avenues when approaching novel competitive approximations. Even more broadly the idea of convex surrogate likelihood based learning can bee seen in the context of the variational inference approach, as developed in [Wainwright and Jordan, 2008], that appears to be a powerful tool complementing the broad class of Monte Carlo simulation based methods. The theory of the variational inference has now been brought to some degree of maturity, many important properties have recently been proved and this advancement is further reflected in emerging empirical results.

- We provide a way to compare performance of the developed trainable function for automatic image interpretation that is based on different

> formulations of the global statistical models. Specifically our contribution is to describe how learning parameters of the global statistical model of the class described in Section 3.1 can be reduced to learning parameters of the global statistical models like the Potts model, by confining the learning to the subclasses of models described in Section 3.2. We provide a comparison of the performance of the trainable function for automatic image interpretation based on the in the literature rarely adopted class of the data-dependent asymmetric interaction models with the performance of the function based on the in the literature widely adopted subclass of the contrast sensitive Potts models. In our experiments we observe that the performance of the trainable function for automatic image interpretation can notably be improved by letting the function be based on the global statistical models of the class of the data-dependent asymmetric interaction models as opposed to letting the function being based on a model from the subclass of the contrast sensitive Potts models. We point out that this notable improvement is in our experiments achieved in combination with the pseudolikelihood based approximate learning.

In the context of applications we conclude that modeling context using the global statistical models leads to improvement as compared to the conceptually simpler context insensitive classifiers. The development of the more sophisticated global statistical models is in our view conditioned on the prior development of more powerful approaches to tractable approximate learning in this context. In practice, it appears that there are good enough approximations that can be solved efficiently. However novel competitive model formulations and more experiments are needed to further support the statement.

Many problems remain unresolved. The problem of simultaneous three-dimensional reconstruction and interpretation from images taken from multiple views has been attracting growing interest as information about the geometry of a scene provides strong cues for the interpretation task. This topic poses the problem of combining interpretation with the matching of images from the multiple views. Structured prediction models have been applied for instance to the problem of matching images of sparsely textured scenes [Dickscheid, 2011] and to the problem of learning graph matchings [Caetano et al., 2009]. One of the limitations of the probabilistic graphical models is that they have not proved to be a natural way to express shapes of objects. A natural question is then how to extend global structured prediction models to also account for global topological properties [Chen et al., 2011]. We feel that there is also a need to further illuminate the connections to other sources of structured prediction models, in particular to the structured support vector machines [Bakir et al., 2007] that have lately proved to be competitive counterparts of the graphical models. Learning of structured support vector machines is closely related to the learning of discriminative graphical models and identifying the similarities in the foundations of the probabilistic learning and in the foundations of the maximum margin learning may in this context be an insightful undertaking. Facing the difficulty of learning the model structure, a question arises as of what model structures and inference approximations to adopt when facing a task in a particular application domain where specific knowledge is available. In this context it would be beneficial to characterize the tradeoffs between the employment of simple tractable structures and intractable expressive structures that in general can only be approximated. In our view on the practical side an important part of this endeavor should become the production of readily available implementations that would allow the comparison of the methods across application domains and their integration and testing in larger systems. Further large scale learning using approximate stochastic descent methods [Bottou and Bousquet, 2008] has been gaining more attention in the context of applications. Dealing with large amounts of data often amounts to sequential learning which is related to the question of how should a system learn incrementally over possibly long periods of time. Eventually pursuing the answer to the question of how to represent human common sense and expert knowledge about a particular domain and how to incorporate the knowledge in the interpretation process, should become a part of the endeavor of developing systems that learn over time.

Appendix A

Equivalence of Logistic and Linear Terms

We now show that CRF model in equation (3.2) that involves energy function in equation (3.17) is equivalent to the CRF model that involves energy function in equation (3.7). As we describe in Section 1.2.1, the posterior probability masses in equation (3.17) are modeled using multi-class logistic regression model for classification, which is a form of generalized linear model. We show that CRF model that involves energy function, where the unary terms and the pairwise terms are modeled as generalized linear models in the form of multi-class logistic regression models, is equivalent to a CRF model that involves energy function with only the linear models themselves.

We combine the CRF model $p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u})$ in equation (3.2) with the energy function $E(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{u})$ in equation (3.17), where we replace both the conditional probability mass $p_1(x_i|\boldsymbol{y}, \boldsymbol{w})$ in the unary energy term and the conditional probability mass $p_2(x_i, x_{i'}|\boldsymbol{y}, \boldsymbol{v})$ in the pairwise energy term with the respective multi-class logistic regression models for classification. We then write the CRF model $p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u})$ as

$$\frac{1}{Z(\boldsymbol{y}, \boldsymbol{u})} \exp\left(\sum_{i} \log \frac{\exp(\boldsymbol{w}_{x_i}^T \boldsymbol{h}_i)}{\sum_{k=1}^{K} \exp(\boldsymbol{w}_k^T \boldsymbol{h}_i)} + \sum_{(i,i')} \log \frac{\exp(\boldsymbol{v}_{x_i x_{i'}}^T \boldsymbol{\mu}_{ii'})}{\sum_{k,k'=1}^{K} \exp(\boldsymbol{v}_{kk'}^T \boldsymbol{\mu}_{ii'})}\right) \tag{A.1}$$

We show the equivalence by performing couple of algebraic manipulations.

We start by considering pairwise energy terms in equation (A.1) and for the moment neglect unary energy terms in equation (A.1) and only denote them by dots in the subsequent steps. We express the logarithm in the pairwise energy term in equation (A.1) as a sum of two terms and write the

CRF model $p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u})$ as

$$\frac{1}{Z(\boldsymbol{y},\boldsymbol{u})} \exp\left(\sum_{(i,i')}\left(\log\left(\sum_{k,k'=1}^{K}\exp(\boldsymbol{v}_{kk'}^{T}\boldsymbol{\mu}_{ii'})\right)^{-1} + \boldsymbol{v}_{x_i x_{i'}}^{T}\boldsymbol{\mu}_{ii'}\right) + \sum_{i}\dots\right)$$

We split the sum of pairwise energy terms in the above equation into two sums,

$$\frac{1}{Z(\boldsymbol{y},\boldsymbol{u})} \exp\left(\sum_{(i,i')}\log\left(\sum_{k,k'=1}^{K}\exp(\boldsymbol{v}_{kk'}^{T}\boldsymbol{\mu}_{ii'})\right)^{-1} + \sum_{(i,i')}\boldsymbol{v}_{x_i x_{i'}}^{T}\boldsymbol{\mu}_{ii'} + \sum_{i}\dots\right)$$

and replace the exponential function with the product

$$\frac{1}{Z(\boldsymbol{y},\boldsymbol{u})} \exp\left(\sum_{(i,i')}\log\left(\sum_{k,k'=1}^{K}\exp(\boldsymbol{v}_{kk'}^{T}\boldsymbol{\mu}_{ii'})\right)^{-1}\right) \exp\left(\sum_{(i,i')}\boldsymbol{v}_{x_i x_{i'}}^{T}\boldsymbol{\mu}_{ii'} + \dots\right)$$

of two exponentials containing the sums. We express the exponential function on the left in the above equation as a product of exponentials and cancel the exponentials by combining them with the logarithms. We write

$$\frac{1}{Z(\boldsymbol{y},\boldsymbol{u})}\frac{1}{\prod_{(i,i')}\left(\sum_{k,k'=1}^{K}\exp(\boldsymbol{v}_{kk'}^{T}\boldsymbol{\mu}_{ii'})\right)} \exp\left(\sum_{(i,i')}\boldsymbol{v}_{x_i x_{i'}}^{T}\boldsymbol{\mu}_{ii'} + \sum_{i}\dots\right)$$

We note that if an image and a parameter vector are given then the denominator in the second fraction is a constant. We combine the denominator with the normalization constant $Z(\boldsymbol{y},\boldsymbol{u})$ and obtain new normalization constant $Z'(\boldsymbol{y},\boldsymbol{u})$. In the following we again include the unary energy terms in equation (A.1) that we neglected in the previous steps. Thus we arrive at the model

$$\frac{1}{Z'(\boldsymbol{y},\boldsymbol{u})} \exp\left(\sum_{i}\log\frac{\exp(\boldsymbol{w}_{x_i}^{T}\boldsymbol{h}_i)}{\sum_{k=1}^{K}\exp(\boldsymbol{w}_{k}^{T}\boldsymbol{h}_i)} + \sum_{(i,i')}\boldsymbol{v}_{x_i x_{i'}}^{T}\boldsymbol{\mu}_{ii'}\right) \tag{A.2}$$

Let us note that the CRF model in equation (A.2) is equivalent to the CRF model in equation (A.1).

We continue by repeating the above steps for unary energy terms in equation (A.2) and for the moment neglect pairwise energy terms and only denote them by dots in the subsequent steps. We express the logarithm in equation (A.2) as a sum of two terms and write the CRF model $p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{u})$ as

$$\frac{1}{Z'(\boldsymbol{y},\boldsymbol{u})} \exp\left(\sum_{i}\left(\log\left(\sum_{k=1}^{K}\exp(\boldsymbol{w}_{k}^{T}\boldsymbol{h}_i)\right)^{-1} + \boldsymbol{w}_{x_i}^{T}\boldsymbol{h}_i\right) + \sum_{(i,i')}\dots\right)$$

We split the sum of unary energy terms in the above equation into two sums,

$$\frac{1}{Z'(\boldsymbol{y},\boldsymbol{u})}\exp\left(\sum_i \log\left(\sum_{k=1}^{K}\exp(\boldsymbol{w}_k^T\boldsymbol{h}_i)\right)^{-1} + \sum_i \boldsymbol{w}_{x_i}^T\boldsymbol{h}_i + \sum_{(i,i')}\dots\right)$$

and replace the exponential function with the product

$$\frac{1}{Z'(\boldsymbol{y},\boldsymbol{u})}\exp\left(\sum_i \log\left(\sum_{k=1}^{K}\exp(\boldsymbol{w}_k^T\boldsymbol{h}_i)\right)^{-1}\right)\exp\left(\sum_i \boldsymbol{w}_{x_i}^T\boldsymbol{h}_i + \sum_{(i,i')}\dots\right)$$

of two exponentials containing the sums. We express the exponential function on the left in the above equation as a product of exponentials and cancel the exponentials by combining them with the logarithms. We write

$$\frac{1}{Z'(\boldsymbol{y},\boldsymbol{u})}\frac{1}{\prod_i \sum_{k=1}^{K}\exp(\boldsymbol{w}_k^T\boldsymbol{h}_i)}\exp\left(\sum_i \boldsymbol{w}_{x_i}^T\boldsymbol{h}_i + \sum_{(i,i')}\dots\right)$$

We again note that if an image and a parameter vector are given then the denominator in the second fraction is a constant. We combine the denominator with the normalization constant $Z'(\boldsymbol{y},\boldsymbol{u})$ and obtain new normalization constant $Z''(\boldsymbol{y},\boldsymbol{u})$. Thus we obtain an equivalent model

$$\frac{1}{Z''(\boldsymbol{y},\boldsymbol{u})}\exp\left(\sum_i \boldsymbol{w}_{x_i}^T\boldsymbol{h}_i + \sum_{(i,i')}\boldsymbol{v}_{x_ix_{i'}}^T\boldsymbol{\mu}_{ii'}\right)$$

involving energy terms only in the form of linear functions in equation (3.7).

Appendix B

List of Examples

Bibliography

D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of markov random fields for segmentation of 3d scan data. In *Computer Vision and Pattern Recognition*, 2005.

G. H. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. V. N. Vishwanathan, editors. *Predicting Structured Data.* The MIT Press, 2007.

A. Barth, J. Siegemund, A. Meißner, U. Franke, and W. Förstner. Probabilistic multi-class scene flow segmentation for traffic scenes. In *Pattern Recognition*, 2010.

S. D. Bauer, F. Korč, and W. Förstner. The potential of automatic methods of classification to identify leaf diseases from multispectral images. *Precision Agriculture*, 12:361, 2011.

A. C. Berg, F. Grabler, and J. Malik. Parsing images of architectural scenes. In *International Conference on Computer Vision*, 2007.

J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3): 179–195, September 1975.

J. Besag. Efficiency of pseudo-likelihood estimation for simple gaussian fields. *Biometrika*, 64:616–618, 1977.

C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *European Conference on Computer Vision*, 2004.

M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *European Conference on Computer Vision*, 2008.

L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Neural Information Processing Systems*, 2008.

S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d image. In *International Conference on Computer Vision*, 2001.

Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Computer Vision and Pattern Recognition*, 1998.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. *Pattern Analysis Machine Intelligence*, 31:1048–1058, 2009.

C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labelling problem via a new linear programming formulation. In *Symposium on Discrete Algorithms*, 2001.

C. Chen, D. Freedman, and C. H. Lampert. Enforcing topological constraints in random field image segmentation. In *Computer Vision and Pattern Recognition*, 2011.

N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. 3d urban scene modeling integrating recognition and reconstruction. *Int. J. Comput. Vision*, 78: 121–141, 2008.

D. Cremers, T. Pock, K. Kolev, and A. Chambolle. Convex relaxation techniques for segmentation, stereo and multiview reconstruction. In *Advances in Markov Random Fields for Vision and Image Processing.* MIT Press, 2011.

T. Dickscheid. *Robust Wide-Baseline Stereo Matching for Sparsely Textured Scenes.* PhD thesis, University of Bonn, 2011.

J. V. Dillon and G. Lebanon. Statistical and computational tradeoffs in stochastic composite likelihood. In *International Conference on Artificial Intelligence and Statistics*, 2009.

J. V. Dillon and G. Lebanon. Statistical and computational tradeoffs in stochastic composite likelihood. *CoRR*, abs/1003.0691, 2010a.

J. V. Dillon and G. Lebanon. Stochastic composite likelihood. *Journal of Machine Learning Research*, 11:2597–2633, 2010b.

W. D. Ellis. *A Source Book of Gestalt Psychology*. Humanities Press, 1950.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88:303–338, 2010.

V. Franc and B. Savchynskyy. Discriminative learning of max-sum classifiers. *Journal of Machine Learning Research*, 9:67–104, 2008.

B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *International Conference on Computer Vision*, 2009.

V. Ganapathi, D. Vickrey, J. Duchi, and D. Koller. Constrained approximate maximum entropy learning of markov random fields. In *Uncertainty in Artificial Intelligence*, 2008.

D. Geiger and F. Girosi. Parallel and deterministic algorithms from MRFs: Surface reconstruction. *Pattern Analysis and Machine Intelligence*, 13(5): 401–412, 1991.

B. Gidas. Consistency of maximum likelihood and pseudo-likelihood estimators for gibbsian distributions. In *Stochastic differential systems, stochastic control theory and applications*, 1988.

G. Gimel'farb and A. Zalesny. Markov random fields with short- and long-range interaction for modelling gray-scale textured images. In *Computer Analysis of Images and Patterns*, 1993.

G. L. Gimel'farb. Texture modeling by multiple pairwise pixel interactions. *Pattern Analysis and Machine Intelligence*, 18:1110–1114, 1996.

G. L. Gimel'farb. *Image Textures and Gibbs Random Fields*. Kluwer Academic Publishers, 1999.

D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society*, 51 (2):271–279, 1989.

U. Grenander. *Elements of pattern theory.* Johns Hopkins University Press, 1996.

T. Hazan and R. Urtasun. Approximated structured prediction for learning large scale graphical models. *CoRR*, abs/1006.2899, 2010a.

T. Hazan and R. Urtasun. A primal-dual message-passing algorithm for approximated large scale structured prediction. In *Advances in neural information processing systems*, 2010b.

X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *Computer Vision and Pattern Recognition*, 2004.

X. He, R. S. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *European Conference on Computer Vision*, 2006.

T. Heimann, B.J. Morrison, M.A. Styner, M. Niethammer, and S.K. Warfield. Segmentation of knee images: A grand challenge. In *MICCAI Workshop on Medical Image Analysis for the Clinic*, 2010.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

A. Hyvärinen. Consistency of pseudolikelihood estimation of fully visible boltzmann machines. *Neural Computation*, 18:2283–2292, 2006.

E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31:253–258, 1925.

M. Klodt, T. Schoenemann, K. Kolev, M. Schikora, and D. Cremers. An experimental comparison of discrete and continuous shape optimization methods. In *European Conference on Computer Vision*, 2008.

S. Kluckner and H. Bischof. Image-based building classification and 3D modeling with super-pixels. In *Photogrammetric Computer Vision and Image Analysis*, 2010.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, 2009.

F. Korč and W. Förstner. Approximate parameter learning in Conditional Random Fields: An empirical investigation. In *Pattern Recognition*, 2008a.

F. Korč and W. Förstner. Approximate parameter learning in Conditional Random Fields: An empirical investigation. http://www.ipb.uni-bonn.de/uploads/tx_ikgpublication/korc08.approximate.pdf, June 2008b. Revised version of F. Korč and W. Förstner, Approximate Parameter Learning in Conditional Random Fields: An Empirical Investigation. In Pattern Recognition, 2008.

F. Korč and W. Förstner. Interpreting terrestrial images of urban scenes using Discriminative Random Fields. In *International Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences*, 2008c.

F. Korč and W. Förstner. eTRIMS Image Database for interpreting images of man-made scenes. Technical Report TR-IGG-P-2009-01, Dept. of Photogrammetry, University of Bonn, April 2009. URL `http://www.ipb.uni-bonn.de/projects/etrims_db/`.

F. Korč, D. Schneider, and W. Förstner. On nonparametric Markov random field estimation for fast automatic segmentation of MRI knee data. In *Medical Image Analysis for the Clinic - A Grand Challenge workshop, MICCAI*, 2010.

A. M. C. A. Koster, S. P. M. van Hoesel, and A. W. J. Kolen. The partial constraint satisfaction problem: Facets and lifting theorems. *Operations Research Letters*, 23(3-5):89–97, 1998.

A. Kulesza and F. Pereira. Structured learning with approximate inference. In *Advances in neural information processing systems*, 2008.

P. M. Kumar, V. Kolmogorov, and P. H. S. Torr. An analysis of convex relaxations for MAP estimation of discrete MRFs. *Journal of Machine Learning Research*, 10:71–106, 2009.

S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proc. of the 9th IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 2003.

S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *Advances in neural information processing systems*. MIT Press, 2004a.

S. Kumar and M. Hebert. Multiclass discriminative fields for parts-based object detection. In *Snowbird Learning Workshop*, 2004b.

S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *IEEE International Conference on Computer Vision*, volume 2, pages 1284–1291. IEEE Computer Society, October 2005.

S. Kumar and M. Hebert. Discriminative random fields. *International Journal of Computer Vision*, 68(2):179–201, June 2006.

S. Kumar, J. August, and M. Hebert. Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. In *5th International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2005.

L. Ladicky, C. Russell, P. Kohli, and P.H.S. Torr. Associative hierarchical CRFs for object class image segmentation. In *International Conference on Computer Vision*, 2009.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289, 2001.

J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In *International Conference on Machine Learning (ICML)*, 2004.

S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

C.-H. Lee, S. Wang, F. Jiao, D. Schuurmans, and R. Greiner. Learning to model spatial dependency: Semi-supervised discriminative random fields. In *Advances in neural information processing systems*, 2007.

S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 3rd edition, 2009.

P. Liang and M. I. Jordan. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *International Conference on Machine Learning*, 2008.

B. G. Lindsay. Composite likelihood methods. *Comtemporary Mathematics*, 80:221—-239, 1988.

A. Lucchi, Y. Li, X. Boix, K.Smith, and P. Fua. Are spatial and global constraints really necessary for segmentation? In *International Conference on Computer Vision*, 2011.

A.-K. Mahlein. *Detection, identification, and quantification of fungal diseases of sugar beet leaves using imaging and non-imaging hyperspectral techniques*. PhD thesis, University of Bonn, 2010.

S. Mase. Consistency of the maximum pseudo-likelihood estimator of continuous state space gibbsian processes. *Applied Probability*, 5:603–612, 1995.

A. McCallum, K. Rohanimanesh, and C. Sutton. Dynamic conditional random fields for jointly labeling multiple sequences. In *NIPS Workshop on Syntax, Semantics, and Statistics*, December 2003.

B. Micusik and J. Kosecka. Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In *Video-Oriented Object and Event Classification*, 2009.

J. W. Modestino and J. Zhang. A markov random field model-based approach to image interpretation. *Pattern Analysis and Machine Intelligence*, 14(6): 606–615, June 1992.

K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Uncertainty in Artificial Intelligence*, 1999.

S. Nowozin and C. H. Lampert. Global interactions in random field models: A potential function ensuring connectedness. *SIAM J. Imaging Sciences*, 3:4, 2010.

S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, To appear., 2011.

S. Nowozin, P. V. Gehler, and C. H. Lampert. On parameter learning in CRF-based approaches to object class image segmentation. In *European conference on computer vision*, 2010.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., 1988.

N. Plath, M. Toussaint, and S. Nakajima. Multi-class image segmentation using conditional random fields and global classification. In *International Conference on Machine Learning*, 2009.

P. Pletscher, C. S. Ong, and J. M. Buhmann. Spanning tree approximations for conditional random fields. In *International Conference on Artificial Intelligence and Statistics*, 2009.

T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *Computer Vision and Pattern Recognition*, 2009.

R. B. Potts. Some generalized order-disorder transformations. In *Cambridge Philosophical Society*, 1952.

A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems*, pages 1521–1527. IEEE Computer Society, 2004.

A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *Pattern Analysis and Machine Intelligence*, 29 (10):1848–1852, October 2007.

A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Proc. of the 11th IEEE International Conference on Computer Vision*, October 2007.

N. Ripperda and C. Brenner. Evaluation of structure recognition using labelled facade images. In *Pattern Recognition*, 2009.

R. Roscher, B. Waske, and W. Förstner. Kernel discriminative random fields for land cover classification. In *IAPR Workshop on Pattern Recognition in Remote Sensing*, 2010.

M. I. Schlesinger. Sintaksicheskiy analiz dvumernykh zritelnikh singnalov v usloviyakh pomekh (Syntactic analysis of two-dimensional visual signals in noisy conditions). *Kibernetika*, 4:113–130, 1976.

M.I. Schlesinger and V. Hlaváč. *Ten Lectures on Statistical and Structural Pattern Recognition.* Kluwer Academic Publishers, 2002.

P. Schnitzspan, M. Fritz, and B. Schiele. Hierarchical support vector random fields: Joint training to combine local and global features. In *European Conference on Computer Vision*, 2008.

P. Schnitzspan, M. Fritz, S. Roth, and B. Schiele. Discriminative structure learning of hierarchical representations for object detection. In *Computer Vision and Pattern Recognition*, 2009.

P. Schnitzspan, S. Roth, and B. Schiele. Automatic discovery of meaningful object parts with latent CRFs. In *Computer Vision and Pattern Recognition*, 2010.

Q. Shi, M. D. Reid, and T. S. Caetano. Conditional random fields and support vector machines: A hybrid approach. *CoRR*, abs/1009.3346, 2010.

J. Shotton, J. M. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81:2–23, 2009.

C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional random fields for contextual human motion recognition. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, volume 2, pages 1808–1815. IEEE Computer Society, 2005.

D. Sontag, O. Meshi, T. Jaakkola, and A. Globerson. More data means less inference: A pseudo-max approach to structured learning. In *Advances in Neural Information Processing Systems*. MIT Press, 2010.

E. B. Sudderth, M. J. Wainwright, and A. S. Willsky. Loop series and Bethe variational bounds in attractive graphical models. In *Advances in Neural Information Processing Systems*, 2007.

C. Sutton and A. Mccallum. Piecewise training of undirected models. In *Uncertainty in Artificial Intelligence*, 2005.

C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. In *Introduction to Statistical Relational Learning*. MIT Press, 2007a.

C. Sutton and A. McCallum. Piecewise pseudolikelihood for efficient training of conditional random fields. In Zoubin Ghahramani, editor, *International Conference on Machine learning (ICML)*, volume 227, pages 863–870, 2007b.

C. Sutton and A. McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, To appear., 2011.

R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *Pattern Analysis and Machine Intelligence*, 30(6):1068–1080, 2008.

M. Szummer, P. Kohli, and D. Hoiem. Learning CRFs using graph cuts. In *European Conference on Computer Vision*, 2008.

B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Advances in neural information processing systems*, 2004.

K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 2003.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning*, 2004.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

S.V. N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In William W. Cohen and Andrew Moore, editors, *Proc. of the 24th International Conf. on Machine Learning*, volume 148, pages 969–976. ACM Press, 2006.

M. J. Wainwright. Estimating the "wrong" graphical model: Benefits in the computation-limited setting. *Journal of Machine Learning Research*, 7: 1829–1859, 2006.

M. J. Wainwright and M. I. Jordan. *Graphical models, exponential families, and variational inference.* Foundations and Trends in Machine Learning, 2008.

M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Map estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005a.

M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *Information Theory*, 51:2313–2335, 2005b.

L. Wang and D. Suter. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *Computer Vision and Pattern Recognition*, 2007.

S. B. Wang, A. Quattoni, L.-P. Morency, and D. Demirdjian. Hidden conditional random fields for gesture recognition. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

M. Wertheimer. Untersuchungen zur lehre von der gestalt. II. *Psychologische Forshung*, 4:301–350, 1923.

G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction.* Springer, 2nd edition, 2006.

C. Wojek and B. Schiele. A dynamic conditional random field model for joint labeling of object and scene classes. In *European Conference on Computer Vision*, 2008.

M. Y. Yang, W. Förstner, and M. Drauschke. Hierarchical conditional random field for multi-class image classification. In *International Conference on Computer Vision Theory and Applications*, 2010.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. Technical Report TR-2001-22, Mitsubishi Electric Research Laboratories, 2002.

S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, 2:259–362, 2006.

List of Figures

List of Tables

Index