Die approbierte Originalversion dieser Dissertation ist an der Hauptbibliothek der Technischen Universität Wien aufgestellt (http://www.ub.tuwien.ac.at).

The approved original version of this thesis is available at the main library of the Vienna University of Technology (http://www.ub.tuwien.ac.at/englweb/).

Jeremy Jancsary

Approximate Discriminative Training of Graphical Models

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of:

Doktor der technischen Wissenschaften (Dr.techn.)

to the:

Faculty of Electrical Engineering and Information Technology, Vienna University of Technology

advised by:

ao.Univ.-Prof. Dr.techn. Gerald Matz, Institute of Telecommunications

reviewed by:

ao.Univ.-Prof. Dr.techn. Gerald Matz *ao.Univ.-Prof. Dr.techn. Harald Trost*

authored by:

Dipl.-Ing. Jeremy Jancsary (reg. no. 0126401)
Zur Spinnerin 1/3/313 · A-1100 Vienna · Austria

Vienna, August 2012

Copyright © 2012 Jeremy Jancsary

SELF-PUBLISHED BY THE AUTHOR

Licensed under the Creative Commons Attribution license, version 3.0
http://creativecommons.org/licenses/by/3.0/legalcode

First printing, Vienna, August 2012

Contents

List of Figures

List of Tables

Dedicated to the people who helped me bring this thesis to life—through teaching, discussion, and collaboration:

- *Manuel Alcantara-Plà*
- *Mat Cook*
- *Andrew Fitzgibbon*
- *Franz Hlawatsch*
- *Pushmeet Kohli*
- *Alexandra Klein*
- *Brigitte Krenn*
- *Christoph Lampert*
- *Katalin Lejtovicz*
- *Johannes Matiasek*
- *Gerald Matz*
- *Friedrich Neubarth*
- *Sebastian Nowozin*
- *Johann Petrak*
- *Hannes Pirker*
- *Carsten Rother*
- *Toby Sharp*
- *Jamie Shotton*
- *Gregor Sieber*
- *Marcin Skowron*
- *Stephanie Schreitter*
- *Martin Szummer*
- *Harald Trost*
- *Gerhard Widmer*

I would also like to acknowledge 25 years of essentially free public education, provided by the Republic of Austria.

Finally, special thanks go to my dear ones—you know who you are.

Abstract

Over the past decade, graphical models have emerged as a workhorse for statistical processing of data in disciplines as diverse as computer vision, natural language processing, digital communications and computational biology. Problems from all of these disciplines have in common that the objects of interest possess rich internal structure. Graphical models help us make this structure explicit and exploit it during statistical inference.

The proliferation of freely available data has lead to reinforced interest in approaches that *learn* from existing examples how to infer the properties of previously unseen instances. Graphical models provide a sound formal framework towards this end—the *discriminative* learning approach seeks to estimate the parameters of a graphical model such that its predictions are consistent with the observed data. While conceptually simple, the prevalent approaches suffer from computational intractability if the underlying graphical model contains cycles. Unfortunately, this is the case in many applications of practical interest. During recent years, the understanding of approximate inference in graphical models has improved dramatically. Yet, in a learning scenario, intractability is still often dealt with in an ad-hoc or heuristic manner. This thesis aims to aid the goal of bridging this gap.

The first approach we present draws heavily on tools from convex optimization. Based on a variational characterization of the inference problems in graphical models, we present a whole catalogue of equivalent formulations of discriminative learning, each exposing different merits. The idea underlying this approach is to *relax* the inference subproblems, that is, to optimize over a simpler constraint set. We introduce new algorithms that can be used to solve these relaxed problems more efficiently than previously possible. Alternatively, we demonstrate how the variational viewpoint allows to formulate discriminative learning as a single unconstrained convex optimization problem that can be solved using off-the-shelf solvers.

Our second approach is based on a Gaussian model and allows for treatment of both discrete and continuous learning tasks. Discrete variables can be handled either via a high-dimensional encoding, or by optimizing a specific loss function. While the use of a Gaussian predictive density may seem overly restrictive at first, we demonstrate how the expressiveness of the model can be increased significantly via non-parametric conditioning.

We present applications from computer vision and natural language processing that demonstrate the wide applicability of our algorithms and the importance of employing principled approximations. A highlight among our results is that we obtain the best published numbers for natural image denoising and related image restoration problems.

Kurzfassung

Im vergangenen Jahrzehnt haben sich grafische Modelle als wichtiges Werkzeug zur statistischen Analyse von Daten aus unterschiedlichsten wissenschaftlichen Disziplinen—wie etwa maschinellem Sehen, Sprachverarbeitung, Nachrichtentechnik und Biologie—herauskristallisiert. Diesen Disziplinen ist gemein, dass die Aufgabenstellungen typischerweise Objekte mit komplexer interner Struktur behandeln. Grafische Modelle sind dabei behilflich, diese Struktur explizit zu machen, und sie zum Zwecke der statistischen Inferenz auszunützen.

Neuerdings hat die erhöhte Verfügbarkeit von Daten aller Art zu verstärktem Interesse an Ansätzen geführt, die aus vorhandenen Beispielen *lernen*, Eigenschaften bisher ungesehener Objekte vorherzusagen. Grafische Modelle bieten zu diesem Zweck ein wohldefiniertes theoretisches Rüstzeug. Der *diskriminative* Lernansatz zielt darauf ab, die Parameter eines grafischen Modells so zu schätzen, dass die Vorhersagen des Modells mit den beobachteten Daten konsistent sind. Die vorherrschenden Ansätze sind elegant und konzeptuell einfach, leiden jedoch unter dem Problem, dass sie—sofern das grafische Modell Zyklen aufweist—aus rechnerischer Sicht für praktische Zwecke zu langsam sind. In den letzten Jahren hat sich das Verständnis von approximativer Inferenz in grafischen Modellen dramatisch verbessert. Dennoch wird dem ausufernden rechnerischen Aufwand in Lernanwendungen häufig mit "ad-hoc"-Ansätzen oder Heuristiken begegnet. Ziel dieser Dissertation ist es, Teile der neuen theoretischen Erkenntnisse auf die Anwendung in Lernverfahren zu übertragen.

Der erste in dieser Dissertation präsentierte Ansatz basiert primär auf Werkzeugen aus der konvexen Optimierung. Basierend auf einer variationalen Charakterisierung von Inferenz in grafischen Modellen werden mehrere äquivalente Formulierungen des diskriminativen Lernproblems vorgestellt, von denen jede ihre eigenen Stärken aufweist. Die Idee, die diesem Ansatz zugrunde liegt, ist die Inferenz—als Unterproblem des Lernens—zu *relaxieren*, worunter eine Aufweichung der Beschränkungen des ursprünglichen Optimierungsproblems verstanden wird. Es werden neue Inferenzalgorithmen vorgestellt, die solche relaxierten Probleme effizienter lösen können, als dies bisher möglich war. Alternativ dazu wird demonstriert, wie das Problem des diskriminativen Lernens als ein einziges unbeschränktes konvexes Programm dargestellt, und somit handelsübliche Optimierungssoftware zum Einsatz gelangen kann.

Der zweite verfolgte Ansatz basiert auf einem Gauss'schen Modell und erlaubt es, sowohl kontinuierliche als auch diskrete Lernanwendungen zu behandeln. Die Festlegung auf eine Gauss'sche Dichtefunktion mag auf den

ersten Blick als starke Einschränkung empfunden werden; die Mächtigkeit des Modells kann jedoch durch nicht-parametrische Konditionierung auf die beobachteten Daten massiv erhöht werden.

Die in der Dissertation betrachteten Anwendungen aus den Bereichen des maschinellen Sehens und der Sprachverarbeitung untermauern die vielseitige Einsetzbarkeit der vorgestellten Algorithmen, sowie die Notwendigkeit prinzipienbasierter Approximationen. Als besonders erwähnenswert soll unter den in der Dissertation präsentierten Ergebnissen hervorgehoben werden, dass durch die vorgestellten Verfahren die besten bisher publizierten Ergebnisse im Entrauschen natürlicher Bilder und in damit verwandten Bildrestaurierungsproblemen gewonnen werden konnten.

Part I

Introduction and Foundations

Preface

In recent years, the amount of data that is readily available on the Internet has steadily increased. On the one hand, content is actively being created and published by individual users and made available for others to use on-line; on the other hand, an ever-growing amount of data is collected by governments and non-governmental organizations and made accessible in structured or semi-structured formats.

While there is justified concern about the use of sensitive data leading to violation of every human's right to privacy, it is also undoubtedly the case that somewhere within these enormous piles of data, the answers to many scientific questions are buried. In some cases, being able to answer such questions could lead to enormous social and economic benefits. In any case, a reversal of the trend is unlikely, so if we have to live with the negative consequences of an increasingly transparent society, we should at least strive to make the most out of its potential benefits.

Analysis of data has traditionally been the playing field of *statistics*. Originally, the data under consideration was characterized by small sample sizes, as well as low-dimensional explanatory and response variables. In more recent years, a young scientific discipline by the name of *machine learning* has heralded a new era in data analysis, where the number of explanatory variables (typically called *features* in machine learning), as well as the number of data points, have become increasingly large. Inevitably, the trend of increasing dimensionality has eventually swept back to statistics, with scientists frequently working on data sets where the dimensionality of the explanatory variables exceeds the number of data points. Likewise, machine learning has benefitted tremendously from the amount of mathematical rigor that has been developed in statistics over the course of centuries. Differences between the two fields are nowadays perhaps more often rooted in philosophy than in methodology.

Probably the most recent revolution has been brought about by data sets that feature *response variables* of high dimensionality. In this regime, the usual approaches towards classification and regression fall short. Such data sets occur frequently in areas like computer vision, computational biology and natural language processing, where the considered objects have internal structure. *Graphical models* provide a sound formal framework that helps us make this structure explicit and exploit it during statistical inference. However, even the most fundamental queries, such as computation of marginal probabilities and modes, are intractable in discrete graphical models of high tree width. As a consequence, inference and parameter estimation in such graphical models have proved to be a significant challenge.

Contributions of the Thesis

This thesis considers the problems of inference and parameter estimation in graphical models of high tree width. In particular, *discriminative* approaches to parameter estimation or *training* are explored. Besides a unifying review of the relevant literature, the thesis presents the following contributions:

1. Convergent inference algorithms

- Convergent algorithms for approximate inference in discrete graphical models of high tree width; both for computation of marginal probabilities and modes.

2. Training of discrete models

- An in-depth treatment of the topic of convex duality in discrete graphical models, convex relaxations, and their use in discriminative parameter estimation; providing a whole catalogue of tractable convex formulations, some of which have not been considered in the literature before.

3. Training of Gaussian models

- Introduction of a particular *Gaussian* conditional random field model with more expressive parameterization than has been considered before; discriminative training algorithms for discrete and continuous response variables; and finally, a non-parametric extension of the basic model.

The utility of all algorithms is evaluated in a substantial number of experiments, and the wide applicability of the two different approaches to discriminative training is demonstrated by means of several practical applications. In particular, we present the most effective approach towards *natural image denoising* published so far.

Papers that directly contribute to this thesis

J. Jancsary, S. Nowozin, and C. Rother. Loss-Specific Training of Non-Parametric Image Restoration Models: A New State of the Art. In *12th European Conference on Computer Vision (ECCV)*, Florence, Italy, 2012.

J. Jancsary, S. Nowozin, T. Sharp, and C. Rother. Regression Tree Fields – An Efficient, Non-Parametric Approach to Image Labeling Problems. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA.

J. Jancsary and G. Matz. Convergent Decomposition Solvers for Tree-reweighted Free Energies. In *14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Ft. Lauderdale, FL, USA, 2011.

J. Jancsary, G. Matz, and H. Trost. An Incremental Subgradient Algorithm for Approximate MAP Estimation in Graphical Models. In *NIPS 2010 Workshop on Optimization for Machine Learning*, Whistler, BC, Canada, 2010.

J. Jancsary, J. Matiasek, and H. Trost. Revealing the Structure of Medical Dictations with Conditional Random Fields. In *2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Honolulu, HI, USA, 2008.

Further papers that are not directly contained

J. Jancsary, F. Neubarth, S. Schreitter, and H. Trost. Multi-Faceted Analysis of News Articles for Intelligent User- and Context-Sensitive Presentation. *Journal article under preparation*, 2012.

S. Petrik, C. Drexel, **J. Jancsary**, A. Klein, G. Kubin, J. Matiasek, F. Pernkopf, and H. Trost. Semantic and Phonetic Automatic Reconstruction of Medical Dictations. *Computer Speech & Language*, 25(2):363–385, 2011.

J. Jancsary, F. Neubarth, S. Schreitter, and H. Trost. Towards a Context-Sensitive Online Newspaper. In *IUI 2011 Workshop on Context-awareness in Retrieval and Recommendation*, Palo Alto, CA, USA, 2011.

J. Jancsary, F. Neubarth, and H. Trost. Towards Context-Aware Personalization and a Broad Perspective on the Semantics of News Articles. In 4^{th} *ACM Conference on Recommender Systems (RECSYS)*, Barcelona, Spain, 2010.

J. Matiasek, **J. Jancsary**, A. Klein, and H. Trost. Identifying Segment Topics in Medical Dictations. In *EACL 2009 Workshop on Semantic Representation of Spoken Language*, Athens, Greece, 2009.

J. Jancsary, A. Klein, J. Matiasek, and H. Trost. Semantics-Based Automatic Literal Reconstruction of Dictations. In *CAEPIA 2007 Workshop on Semantic Representation of Spoken Language*, Salamanca, Spain, 2007.

M. Huber, **J. Jancsary**, A. Klein, and H. Trost. Mismatch interpretation by semantics-driven alignment. In 8^{th} *Conference on Natural Language Processing (KONVENS)*, Konstanz, Germany, 2006.

Organization

First Part

This thesis is organized in three parts. In the remainder of the first part, we set forth our notion of *structured prediction* and *discriminative training*, giving examples and definitions by other authors, and discussing the main challenges arising in this setting. Moreover, we give an introduction to graphical models—the framework on which we base our two approaches to structured prediction—and point out their relation to exponential families and statistical mechanics. Finally, we consider in some detail the prevailing approaches to discriminative parameter estimation, such as the *maximum conditional likelihood* and *large margin* principles.

Second Part

In the second part of the thesis, we consider inference and discriminative training in discrete graphical models by means of convex relaxations. We start by demonstrating how exact inference in discrete graphical models can be understood as an optimization problem, moving on to discuss the specific relaxation taken by the *local polytope*, and introducing new algorithms for computation of approximate marginal probabilities and modes over this relaxed polytope. Finally, we use our insights to derive several classes of equivalent convex optimization formulations of discriminative parameter estimation, and provide comparisons in terms of practical applications.

Third Part

In the third and final part, we first take an in-depth look at Gaussian graphical models, again from a convex optimization perspective. We move on to describe several efficient and exact inference algorithms, and use our insights to devise two approaches to parameter estimation within our Gaussian model that are applicable both to discrete and continuous response or output variables: the first approach is based on maximizing the condi-

tional likelihood of an encoding of the response, while the second approach draws on differentiable loss functions. We discuss how the expressiveness of Gaussian conditional random fields can be increased by means of non-parametric conditioning on the input. In doing so, we alleviate the restriction to modelling of uni-modal data. Finally, we conclude the thesis by describing several applications, both discrete and continuous in nature, and comparing the effectiveness of our Gaussian model to previous state-of-the-art approaches.

Structured Prediction

In the overview of the thesis, we already briefly mentioned that the methods presented herein are designed to predict structured objects. Indeed, this task has become important enough for there to be a whole discipline dedicated to the topic, which goes by the name of *structured prediction*.

General Definition

Ironically enough, although the problem has received significant attention in recent years, a generally accepted definition of *structured prediction* is conspicuously hard to find. Very broadly, the problem consists of finding a map from an input space $\mathcal{X}$ to a *structured* output space $\mathcal{Y}$. Beyond this generally accepted definition, structured prediction is mostly defined in terms of examples. For instance, Smith[1] defines the problem as follows:

> *The word structure evokes ideas about complexity and interconnectedness; in machine learning the term structure prediction (or structured prediction) is used to refer to prediction of a set of interrelated variables. Such problems arise in areas like computer vision (e.g., interpreting parts of a visual scene), computational biology (e.g., modeling how protein molecules fold), and, of course, speech and text processing.*

A similar definition is set forth by Daumé,[2] who emphasizes the differences from typical classification and regression tasks:

> *Structured prediction is a generalized task that encompasses many problems in natural language processing, as well as many problems from computational biology, computational vision and other areas. The key issue in structured prediction that differentiates it from more canonical machine learning tasks (such as classification or regression) is that the objects being predicted have internal structure.*

Again, this definition does not add much in terms of a more formal characterization of the problem.

Since structured prediction mostly seems to be defined in terms of examples, then, it is only natural to give examples of the kind of problems that will be considered in this thesis. In Figure 1, we show four exemplary tasks that are meant to illustrate the discipline. In *semantic segmentation*, the goal is to assign class labels to the pixels of an image that encode which parts of the image belong to what kind of object (a *bird*, in this example). Problems of this kind have been considered by Shotton et al.[3], among many others. As another example, one may be interested in *parsing* transcripts of medical dictations[4] to make explicit the structure (sections, headings, etc.) that is underlying the report. To give one more example from natural language and speech processing, consider the problem of predicting the *pronunciation*

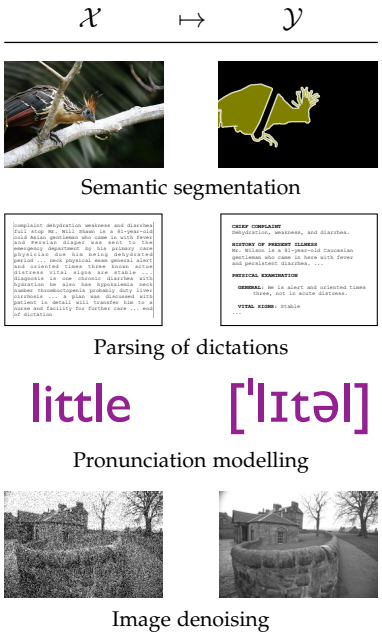


Figure 1: **Examples**: Structured prediction seeks to find a map from an input space $\mathcal{X}$ to a structured output space $\mathcal{Y}$.

[1] Noah A. Smith. *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, 2011

[2] Hal Daumé III. *Practical Structured Learning Techniques for Natural Language Processing*. PhD thesis, University of Southern California, 2006

[3] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation. In *European Conference on Computer Vision (ECCV)*, 2006

[4] Jeremy Jancsary, Johannes Matiasek, and Harald Trost. Revealing the Structure of Medical Dictations with Conditional Random Fields. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008

of a word given its written representation. This problem has been treated in-depth by Bisani and Ney,[5] and clearly it is the case that the output space consists of several interconnected units (*phonemes*, in this case). Finally, we consider an example we will treat in great detail in this thesis, namely *natural image denoising*. Here, the goal is to recover an image that has been corrupted by additive white Gaussian noise. The output space is defined by the set of all uncorrupted natural images, and again, the pixel values are clearly interrelated. The current state of the art for this widely considered problem is held by Zoran and Weiss,[6] but will be shown to be surpassed using the methods developed in this thesis.

While a universally accepted formal definition of *structured prediction* does not seem to exist, we can still formalize the setting we will be concerned with in this thesis.

[5] Maximilian Bisani and Hermann Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, 2008

[6] Daniel Zoran and Yair Weiss. From Learning Models of Natural Image Patches to Whole Image Restoration. In *International Conference on Computer Vision (ICCV)*, 2011

Our Notion of Structured Prediction

The kind of structured prediction problems that will be considered in this thesis can be defined in formal terms as follows. We assume the standard setup for statistical learning, i.e. an unknown distribution $p(\mathbf{x}, \mathbf{y})$ over the set of labelled examples, $\mathcal{X} \times \mathcal{Y}$, where $\mathbf{x} \in \mathcal{X}$ denotes the observed input of an example and $\mathbf{y} \in \mathcal{Y}$ denotes the correct output, typically referred to as the corresponding *ground truth* or the *labels* of the example.

Our goal in this setting is to *learn* a map

$$\hat{\mathbf{y}} \colon \mathcal{X} \mapsto \mathcal{Y} \tag{1}$$

that achieves low expected error

$$R_\ell[\hat{\mathbf{y}}(\cdot)] = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}[\ell(\hat{\mathbf{y}}(\mathbf{x}), \mathbf{y})] \tag{2}$$

with respect to a loss function $\ell \colon \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}^+$. The loss functions we will consider satisfy the intuitive constraint that $\ell(\mathbf{y}, \mathbf{y}) = 0$.

The Role of Structure

The structured nature of the problems under consideration enters our definition in terms of the output space $\mathcal{Y}$. In particular, we restrict our attention to problems in which $\mathcal{Y} \subseteq \mathbb{R}^{v(\mathbf{x})}$, i.e. where the output space is a random vector of a fixed size $v(\mathbf{x})$ that can be determined readily from the observed input $\mathbf{x} \in \mathcal{X}$. Notably, this definition excludes problems in which the dimensionality of the structured output is a priori unknown and needs to be inferred as part of the prediction process. While this may seem restrictive at first, in many cases of interest it is in fact possible to encode variable-length structures in terms of a fixed number of random variables.

We make the additional assumption that the components of our output space $\mathcal{Y}$ are statistically dependent, such that it is not a sound approach to predict each random variable in the vector independently. Nonetheless, some of the variables may be statistically independent given the outcome of others. Such structural *sparsity* can be made precise using probabilistic graphical models,[7] which we will consider in some detail in the following chapter. To make a long story short, we will consider the problem of predicting the outcome of a random vector.

[7] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008

Three Main Ingredients

Given our characterization of structured prediction as loss-conscious prediction of the outcome of a random vector, several questions arise naturally.

Modelling

Perhaps the most immediate question is how the distribution of this random vector depends on the observed input $\mathbf{x}$. Our approach in this thesis will be *discriminative*, that is, we will directly model the posterior density $p(\mathbf{y} \mid \mathbf{x})$ to obtain a map from input to ground truth. In contrast, a *generative* approach models the joint probability $p(\mathbf{x}, \mathbf{y})$ and makes predictions by using Bayes' rule to compute $p(\mathbf{y} \mid \mathbf{x})$. However, as Vapnik notes,[8]

> *... one should solve the problem directly and never solve a more general problem as an intermediate step.*

In other words, if our ultimate goal is to infer $\mathbf{y}$, we should directly focus on this aspect of the problem. In passing, we want to note that the generative approach does have its own share of advantages, such as being able to draw samples from the joint distribution. In any case, treatment of these topics exceeds the scope of this thesis, as our primary goal is to develop new techniques for the *discriminative* approach.

The second important aspect in modelling the distribution of the output vector $\mathbf{y}$ is to choose from a family of distributions and an according parameterization, which we denote by $\mathbf{w}$. For instance, we may wish to restrict $p(\mathbf{y} \mid \mathbf{x}; \mathbf{w})$ to being Gaussian. In that case, the parameterization in terms of $\mathbf{w}$ determines the mean and the covariance of the density. More generally, we will restrict our attention to *exponential families*. We will make this point more precise in the chapters to follow.

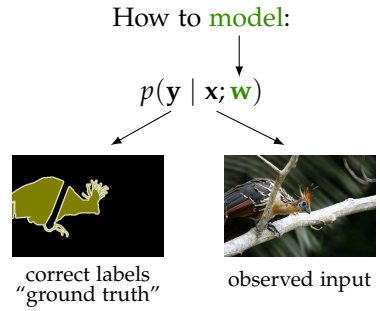


Figure 2: **Modelling**: Define a predictive density capable of representing the statistics of the task at hand faithfully.

[8] Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998

Prediction

Given the predictive density $p(\mathbf{y} \mid \mathbf{x}; \mathbf{w})$, the question arises how the map $\hat{\mathbf{y}} \colon \mathcal{X} \mapsto \mathcal{Y}$ should be defined in terms of this object. We make the assumption that $p(\mathbf{y} \mid \mathbf{x}; \mathbf{w})$ equals the *true* posterior distribution $p(\mathbf{y} \mid \mathbf{x})$.

Naïvely, one might simply choose the most likely labeling,

$$\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}) \stackrel{\text{def}}{=} \arg\max_{\mathbf{y} \in \mathcal{Y}} p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}). \tag{3}$$

However, this choice is optimal only if our goal is to optimize a 0-1 loss function that assigns equal cost of 1 to any incorrect prediction.

More generally, remember that our goal is to minimize the expected loss with respect to a loss function $\ell \colon \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}^+$. In this case, from basic decision theory,[9] the optimal map should be chosen as

$$\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}) \stackrel{\text{def}}{=} \arg\min_{\hat{\mathbf{y}} \in \mathcal{Y}} \int_{\mathcal{Y}} p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) \ell(\hat{\mathbf{y}}, \mathbf{y}) \, \mathrm{d}\mathbf{y}. \tag{4}$$

In both cases, a central question is whether the operation can be carried out efficiently. The answer to this question depends both on the nature of the density, as well as on the loss function. Hence, already in the modelling

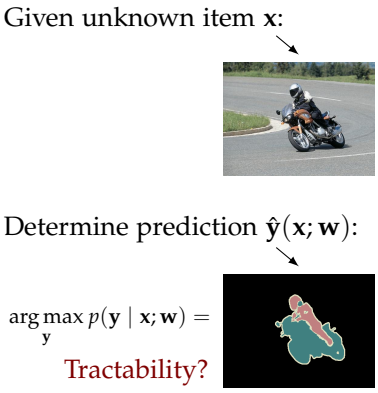


Figure 3: **Prediction**: Given the observed input of an unknown item, the goal is to infer the correct labels, i.e. the "prediction" under our model.

[9] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. Prentice Hall, 1998

Given pairs $(\mathbf{x}, \mathbf{y})$:

Estimate parameters as:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \underbrace{\prod_{(\mathbf{x},\mathbf{y})} p(\mathbf{y} \mid \mathbf{x}; \mathbf{w})}_{\text{likelihood}} \quad \text{(ML)}$$

Tractability?

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \underbrace{\sum_{(\mathbf{x},\mathbf{y})} \overbrace{\ell(\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}), \mathbf{y})}^{\text{loss of prediction}}}_{\text{empirical risk}} \quad \text{(ER)}$$

Figure 4: **Parameter estimation**: Given pairs of labelled items, estimate the parameters of our model such that the data is faithfully represented.

step, one should ensure tractability through the choice of an appropriate family of density functions. An alternative, of course, is to carry out the optimization *approximately*, in which case, however, it is often difficult to make meaningful statements about the quality of the approximate solution.

Parameter Estimation

The final ingredient in our structured prediction framework is to estimate the parameters $\mathbf{w}$ that determine the predictive density. Towards this end, we are given a number of independent and identically distributed (i.i.d.) labelled examples $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\}$. This step is often referred to as the *learning* or *training* phase in the machine learning literature. We will restrict our discussion to classical point estimates—a Bayesian treatment, as in Qi et al.[10] is possible, but often prohibitively expensive for structured prediction tasks. Tractability is again a major concern, even more so than in prediction.

Remember that we have chosen a discriminative approach, i.e. to model $p(\mathbf{y} \mid \mathbf{x}; \mathbf{w})$ directly. A principal method from estimation theory[11] is to choose the parameters so as to optimize the (conditional) *likelihood* of the data. In our setting, this translates to

$$\hat{\mathbf{w}}_{\text{ML}} = \arg\max_{\mathbf{w}} \prod_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}). \tag{5}$$

A prime motivation for following this route is that the maximum likelihood estimate thus obtained is asymptotically *consistent*,[12] i.e. it converges to the true parameters in the limit of infinite data.

A complication arises if our model is *misspecified*, for instance because the chosen family of the predictive density is too restrictive. In this case, one cannot expect a consistent maximum likelihood estimate, since the data we consider was not generated from the family of distributions our model belongs to in the first place. This fact is of high practical relevance. It is perhaps not an overstatement to claim that the majority of machine learning models are misspecified. The reason is simply that practitioners tend to choose *tractable* probability densities to work with, and often these are not quite sufficiently expressive to model the data at hand.

An alternative is then to directly choose the parameters $\mathbf{w}$ following the *empirical risk minimization* principle,

$$\hat{\mathbf{w}}_{\text{ER}} = \arg\min_{\mathbf{w}} \frac{1}{n} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} \ell(\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}), \mathbf{y}) \tag{6}$$

$$\approx \arg\min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim p(\mathbf{x},\mathbf{y})} [\ell(\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}), \mathbf{y})] \tag{7}$$

whereby the parameters are chosen to minimize an empirical estimate of the loss incurred by our map $\hat{\mathbf{y}}(\cdot; \mathbf{w})$. This has the effect of directly selecting $\mathbf{w}$—within the possibly restrictive family of our predictive density—such that the predictions under the model are of high quality in the sense of ℓ. Moreover, this approach gives us flexibility in choosing the map $\hat{\mathbf{y}}(\cdot, \mathbf{w})$. Commonly, it is chosen as in (3), that is, to return a mode of the density $p(\mathbf{y} \mid \mathbf{x}; \mathbf{w})$. Compared to the minimum expected loss problem (4), this simplifies the prediction task for previously unseen items, because the loss function was already accounted for during parameter estimation.

[10] Yuan Qi, Martin Szummer, and Thomas P. Minka. Bayesian Conditional Random Fields. In *Artificial Intelligence and Statistics (AISTATS)*, 2005

[11] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory*. Prentice Hall, 1999

[12] Robert V. Hogg, Allen Craig, and Joseph W. McKean. *Introduction to Mathematical Statistics*. Pearson Education, 2005

Graphical Models and Discriminative Training

In the previous chapter, we took an abstract viewpoint and characterized structured prediction as the problem of predicting the outcome of a random vector. From this bird's-eye view, our task may appear to have been solved (as you may have guessed, this is not the case). The main complication arises from the sheer size of the random vectors we will be considering. For instance, in image processing, it is not uncommon to deal with images of size $1,000 \times 1,000$ or even larger, resulting in over a million random variables if one chooses to model at the pixel level.

As a consequence, it becomes completely intractable to specify the predictive density $p(\mathbf{y} \mid \mathbf{x})$ without further provisions. We will need to exploit the *structure* of the problem. Probabilistic graphical models[13] allow us to specify independence statements regarding the individual variables of the joint density. This in turn enables more efficient computations and reduced memory requirements as opposed to working with the full, unstructured density over all variables.

In this chapter, we will introduce graphical models formally and specialize the key ingredients we previously introduced (modelling, predicting, and parameter estimation) to this new setting. In particular, we will introduce two of the most commonly used approaches towards discriminative training of graphical models, *conditional random fields*[14] and *max-margin Markov networks*.[15] As we shall see, tractability of these approaches depends crucially on the graph structure, motivating the need for approximations.

In general, graphical models are an immensely deep and well-studied topic, so we will only be able to scratch the surface and introduce the tools that will be required by our approach in the sequence.

[13] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009

[14] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *International Conference on Machine Learning (ICML)*, 2001

[15] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-Margin Markov Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2003

Notation for Random Vectors

We will consider random vectors $\mathbf{Y}$ of multiple random variables Y_s taking on values $y_s \in \mathcal{Y}_s$. Of particular interest will be the case where each Y_s is discrete, taking on one of m_s values such that $\mathcal{Y}_s = \{0, 1, \ldots, m_s - 1\}$, as well as the case where Y_s is continuous and $\mathcal{Y}_s = \mathbb{R}$. A joint realization of the random vector will be denoted by $\mathbf{y} \in \mathcal{Y}$.

In this chapter, to abstract over the precise nature of the random vector, we will work with probability distributions represented as densities p that are absolutely continuous with respect to a measure ν. For discrete random variables, we will choose the counting measure on their state space to obtain a probability mass function, whereas for continuous random variables, ν will denote the ordinary Lebesgue measure on $\mathbb{R}$.

Undirected Graphical Models

Undirected graphical models are defined by a graph $G = (V, E)$ consisting of vertices $s \in V$ and undirected edges $(s, t) \in E$, where $E \subseteq V \times V$ denotes the edge set of G. Each vertex $s \in V$ corresponds to a variable Y_s of random vector $\mathbf{Y}$. We can then specify the joint probability density in terms of fully connected[16] subsets $F \subseteq V$ of vertices of G, via

$$p(\mathbf{y}) = \frac{1}{Z} \prod_{F \in \mathcal{F}} \psi_F(\mathbf{y}_F) \; . \tag{8}$$

We refer to each ψ_F (or F for short) as a *factor*, and to $\mathcal{F}$ as the set of factors, containing each maximal clique[17] of G at least once. In order for $p(\mathbf{y})$ to be a valid probability density, we require each ψ_F to be positive and finite. We use $\mathbf{y}_F$ to denote a realization of $\mathbf{Y}_F$, the random vector consisting of the subset of variables specified by $F \subseteq V$. Finally, Z is a normalization constant and usually referred to as the *partition function*.

[16] Full connectivity within F prescribes that $(s, t) \in E$ for all $s, t \in F$.

[17] A maximal clique is a fully connected set of vertices that cannot be extended by including one more adjacent vertex.

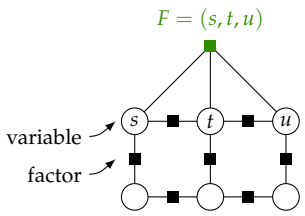


Figure 5: A factor graph involving multiple factors defined over pairs of variables, and a single factor defined over a triplet (s, t, u).

Visualization

If each factor is uniquely defined over subsets of at most two variables, the graph describing the factorization of $p(\mathbf{y})$ can easily be visualized by means of vertices and edges.

On the other hand, if the factors involve larger subsets of V, or multiple factors are defined over the same subset, we shall find it convenient to use a *bipartite* graph involving variable nodes and factor nodes. This representation is called a *factor graph*.[18] We show an exemplary factor graph involving factors of cardinality two and three in Figure 5. Variables are generally represented as circles, whereas factors are depicted as (solid) boxes.

[18] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001

Inference in Graphical Models

Now that we know how to specify a graphical model, it will be interesting to see what kind of queries can be posed to a graphical model.

Marginalization. Perhaps the most important operation within a graphical model is to compute the marginal distribution of a single variable or a subset of variables,

$$p_s(y_s) = \int_{\mathcal{Y}_{V \setminus s}} p(\mathbf{y}) \, \nu(\mathrm{d}\mathbf{y}_{V \setminus s}) \tag{9}$$

and

$$p_F(\mathbf{y}_F) = \int_{\mathcal{Y}_{V \setminus F}} p(\mathbf{y}) \, \nu(\mathrm{d}\mathbf{y}_{V \setminus F}). \tag{10}$$

As we will point out later in this chapter, this problem is intimately related to computation of the partition function Z, defined as

$$Z = \int_{\mathcal{Y}} \prod_{F \in \mathcal{F}} \psi_F(\mathbf{y}_F) \, \nu(\mathrm{d}\mathbf{y}). \tag{11}$$

Our particular interest in marginalization arises from the fact that it has important applications in discriminative training, to be discussed presently.

MAP estimation. Another fundamental operation in a graphical model is to compute a mode of the distribution,

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}). \tag{12}$$

This task is also referred to as *maximum a-posteriori* (MAP) estimation. In general, since we do not assume a uni-modal distribution, the solution to this optimization problem need not be unique.

Tractability. Whether or not the above operations can be carried out efficiently (i.e. in polynomial time) depends both on the family of the probability density $p(\mathbf{y})$ and the structure of the graph G specifying our graphical model. In any case, an important selling point of graphical models is that the factorization of the density can be exploited gainfully. This part is important, since if the structure of the graph didn't gain us anything, there wouldn't be a point in using a graphical model after all.

We will discuss tractability of two particular types of graphical models—*discrete* and *Gaussian* Markov random fields—in detail later in this thesis.

Graphical Models as Exponential Families

Any undirected graphical model with strictly positive density p can be expressed as a member of an exponential family.[19] By virtue of positivity of the factors ψ_F, we can thus equivalently express the joint density as

$$p(\mathbf{y};\theta) = \frac{1}{Z(\theta)} \prod_{F\in\mathcal{F}} \exp\langle \theta_F, \boldsymbol{\phi}_F(\mathbf{y}_F)\rangle \tag{13}$$

where $\boldsymbol{\phi}_F(\mathbf{y}_F)$ denotes the vector of sufficient statistics of factor F, and θ_F are the corresponding exponential parameters. Clearly, this follows again the factorization over subsets of vertices of G we previously assumed.

By collecting all sufficient statistics $\boldsymbol{\phi}_F(\mathbf{y}_F)$ into a single vector $\boldsymbol{\phi}(\mathbf{y})$, the density can be written more compactly as

$$p(\mathbf{y};\theta) = \exp\left(\langle \theta, \boldsymbol{\phi}(\mathbf{y})\rangle - A(\theta)\right), \tag{14}$$

with

$$A(\theta) = \log Z(\theta) = \int_{\mathcal{Y}} \exp\langle \theta, \boldsymbol{\phi}(\mathbf{y})\rangle \, \nu(\mathrm{d}\mathbf{y}). \tag{15}$$

This form follows the *canonical* representation of an exponential family, in which $A(\theta)$ plays the role of the log-partition function.

A lot hinges on the question of how the vector of sufficient statistics $\boldsymbol{\phi}(\mathbf{y})$ is defined. In particular, these statistics determine the exponential family of our undirected graphical model G. In general, if $\boldsymbol{\phi}(\mathbf{y})$ is a d-dimensional vector, then $\theta \subseteq \mathbb{R}^d$, as it may need to satisfy various constraints.

As illustrated in Figure 6, within a fixed exponential family, each factor is fully specified in terms of its exponential parameters θ_F. We will use this fact later on when we introduce *discriminative* graphical models, in which the θ_F are allowed to depend on the observed input.

We will now have a first look at two particular exponential families on which we are going to build our structured prediction framework in the sequence, and make the choice of sufficient statistics concrete. For details on further exponential families, we refer to Wainwright and Jordan.[20]

[19] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009

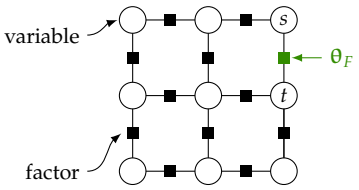


Figure 6: Within a given exponential family, each factor is fully specified by its exponential parameters $\boldsymbol{\theta}_F$.

[20] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008

Y_s	Y_t	$\theta_F(\cdot,\cdot)$
0	0	-0.3
0	1	1.4
⋮	⋮	⋮
y_s	y_t	$\theta_F(y_s, y_t)$
⋮	⋮	⋮

Figure 7: In a discrete MRF, each factor can be completely specified by means of a table that assigns a potential to each joint state of the variables of the factor.

[21] A random vector $\mathbf{Y}$ is Markov with respect to graph G if any two non-adjacent variables are conditionally independent given all other variables, i.e.

$$Y_s \perp\!\!\!\perp Y_t \mid Y_{V\setminus\{s,t\}} \quad \text{for all } s,t \in V.$$

[22] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009

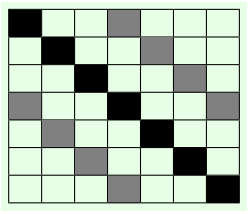


Figure 8: In a Gaussian MRF, the precision matrix $\mathbf{J}$ is *sparse*, and if the pairwise factors are instantiated in a repetitive manner, it typically exposes a sparsity pattern defined by diagonal bands. The main diagonal stores the variances.

Example 1 (Discrete MRF) In a discrete Markov random field (MRF), the sufficient statistics of a factor are chosen as a vector of binary indicators. Formally, the joint state space is given by $\times_{s\in F} \mathcal{Y}_s$. Let m_F be the cardinality of this state space. In this case, the sufficient statistics of factor F are a map $\boldsymbol{\phi}_F \colon \mathcal{Y}_F \mapsto \{0,1\}^{m_F}$ to an m_F-dimensional indicator vector $\mathbb{I}$, each component α of which represents a particular joint state $\mathbf{y}_{F;\alpha}$ via

$$\mathbb{I}_\alpha(\mathbf{y}_F) = \begin{cases} 1 & \text{if } \mathbf{y}_F = \mathbf{y}_{F;\alpha} \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

The vector of exponential parameters $\boldsymbol{\theta}_F$ is hence also of size m_F, and each of its components corresponds to a particular joint state. This is illustrated for a pairwise factor in Figure 7. The vector consisting of all sufficient statistics is *unconstrained*, that is, $\boldsymbol{\theta} \in \mathbb{R}^d$.

Note that the distribution of any discrete random vector $\mathbf{Y}$ that is Markov with respect to G can be described in this manner.[21] However, in general, the representation in terms of indicator vectors as defined above is *overcomplete*, that is, there can be multiple exponential parameter vectors $\boldsymbol{\theta}$ resulting in the same distribution. □

Example 2 (Gaussian MRF) The density of the v-dimensional normal distribution $\mathcal{N}(\mathbf{u}, \mathbf{C})$ with mean $\mathbf{u}$ and covariance $\mathbf{C}$ in standard form is

$$p(\mathbf{y}; \mathbf{u}, \mathbf{C}) = (2\pi)^{-\frac{v}{2}} \det(\mathbf{C})^{-\frac{1}{2}} \exp(-\tfrac{1}{2}(\mathbf{y}-\mathbf{u})^\mathsf{T}\mathbf{C}^{-1}(\mathbf{y}-\mathbf{u})), \quad \mathbf{C} \succ 0. \tag{17}$$

An alternative parameterization of the same Gaussian density, referred to as the *canonical* or *information* form $\mathcal{C}(\mathbf{h}, \mathbf{J})$, is defined as[22]

$$p(\mathbf{y}; \mathbf{h}, \mathbf{J}) = \exp(-\tfrac{1}{2}\mathbf{y}^\mathsf{T}\mathbf{J}\mathbf{y} + \mathbf{y}^\mathsf{T}\mathbf{h} - A(\mathbf{h}, \mathbf{J})), \quad \mathbf{J} \succ 0. \tag{18}$$

This canonical form maps to standard form using the equalities

$$\mathbf{J} = \mathbf{C}^{-1}, \qquad \mathbf{C} = \mathbf{J}^{-1}, \tag{19}$$

$$\mathbf{h} = \mathbf{C}^{-1}\mathbf{u}, \qquad \mathbf{u} = \mathbf{J}^{-1}\mathbf{h}, \tag{20}$$

and the normalization constant A can be computed as

$$A(\mathbf{h}, \mathbf{J}) = \tfrac{1}{2}\mathbf{h}^\mathsf{T}\mathbf{J}^{-1}\mathbf{h} + \tfrac{v}{2}\log(2\pi) + \tfrac{1}{2}\log\det(\mathbf{J}^{-1}). \tag{21}$$

By defining $\boldsymbol{\phi}(\mathbf{y}) = \begin{pmatrix} \mathbf{y} \\ \operatorname{vec}(\mathbf{y}\mathbf{y}^\mathsf{T}) \end{pmatrix}$ and $\boldsymbol{\theta} = \begin{pmatrix} \mathbf{h} \\ -\frac{1}{2}\operatorname{vec}(\mathbf{J}) \end{pmatrix}$, one can easily verify that Gaussians form an exponential family. Observe that since $\mathbf{J} \succ 0$, the exponential parameters are constrained such that $\boldsymbol{\theta} \subset \mathbb{R}^d$.

The positive-definite $v \times v$ matrix $\mathbf{J} \succ 0$, defined as the inverse of the covariance matrix, goes by the names *precision*, *information*, or *concentration* matrix. The Markov property of a Gaussian MRF can be defined succinctly in terms of $\mathbf{J}$: if $(s,t) \notin E$, then $J_{st} = 0$. This is illustrated in Figure 8.

By the Hammersley-Clifford theorem, we can factor this distribution as

$$p(\mathbf{y}) \propto \prod_{F\in\mathcal{F}} \psi_F(\mathbf{y}_F) \tag{22}$$

with $\psi_F \sim \mathcal{C}(\mathbf{h}_F, \mathbf{J}_F)$, where the $\mathbf{h}_F$ and $\mathbf{J}_F$ parameters of the individual factors must be chosen to add up to $\mathbf{h}$ and $\mathbf{J}$, respectively. This choice is not in general unique; moreover, the parameterization in terms of $\mathbf{h}$ and $\mathbf{J}$ is *overcomplete* itself, since $\mathbf{J}$ is symmetric and thus redundant. □

Maximum Entropy Justification

The representation as an exponential family can be motivated as follows. Assume we are given n i.i.d. realizations $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(n)}$ of our random vector and form the empirical estimate of the sufficient statistics, $\hat{\mu} = \frac{1}{n}\sum_i \boldsymbol{\phi}(\mathbf{y}^{(i)})$. We now want to pick our distribution p such that it is *consistent* with the data, i.e. ,

$$\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta)}[\boldsymbol{\phi}(\mathbf{y})] \stackrel{!}{=} \hat{\mu} \tag{23}$$

holds. An important observation is that in general, there are many distributions p that meet this constraint. Therefore, we need a principle for choosing among these distributions. For lack of further information, it is reasonable to choose a *maximally vague* distribution, as characterized by possessing the greatest entropy among all distributions satisfying the consistency constraint. Under weak technical conditions, the density can be shown to follow the form $\exp(\langle\theta, \boldsymbol{\phi}(\mathbf{y})\rangle - A(\theta))$ of an exponential family.

Importance of the Mean Parameters

In the previous section, we saw that the *expected* sufficient statistics play an important role in exponential families. We will refer to

$$\boldsymbol{\mu}(\theta) = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta)}[\boldsymbol{\phi}(\mathbf{y})] \tag{24}$$

as the *mean parameters* resulting from the exponential density over graph G, parameterized by θ. The set of all realizable mean parameters,

$$\mathcal{M}(G) = \{\boldsymbol{\mu} \mid \exists p \text{ s.t. } \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}[\boldsymbol{\phi}(\mathbf{y})] = \boldsymbol{\mu}\}, \tag{25}$$

will turn out to be of particular importance in the sequence. In the case of a discrete MRF, where the vector of sufficient statistics is defined in terms of indicator functions, the mean parameters are given by the marginal probabilities of the factors $F \in \mathcal{F}$. In contrast, in a Gaussian MRF, the mean parameters comprise the mean vector $\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta)}[\mathbf{y}]$ as well as the second-order moment matrix $\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta)}[\mathbf{y}\mathbf{y}^\mathsf{T}]$ and satisfy $\mathbb{E}[\mathbf{y}\mathbf{y}^\mathsf{T}] - \mathbb{E}[\mathbf{y}]\mathbb{E}[\mathbf{y}^\mathsf{T}] \succ 0$.

Variational characterization of the log-partition function Via the mean parameters, $A(\theta)$ can be described in terms of an optimization problem

$$A(\theta) = \max_{\boldsymbol{\mu}\in\mathcal{M}^\circ} \left\{\langle\theta, \boldsymbol{\mu}\rangle + H(p_{\theta(\boldsymbol{\mu})})\right\}, \tag{26}$$

where

$$H(p_{\theta(\boldsymbol{\mu})}) = -\int_{\mathcal{Y}} p(\mathbf{y};\theta(\boldsymbol{\mu}))\log p(\mathbf{y};\theta(\boldsymbol{\mu}))\,\nu(\mathrm{d}\mathbf{y}) \tag{27}$$

denotes the entropy of an exponential density parameterized by a vector $\theta(\boldsymbol{\mu})$ for which the condition

$$\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta(\boldsymbol{\mu}))}[\boldsymbol{\phi}(\mathbf{y})] \stackrel{!}{=} \boldsymbol{\mu} \tag{28}$$

holds.[23] At least one such $\theta(\boldsymbol{\mu})$ exists for each point in $\mathcal{M}^\circ$, the interior of $\mathcal{M}$. In turn, the maximum in (26) is attained uniquely at the interior point

$$\boldsymbol{\mu}(\theta) = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta)}[\boldsymbol{\phi}(\mathbf{y})], \tag{29}$$

i.e. the mean parameters obtained for θ. These conditions define a dual coupling between exponential parameters θ and mean parameters $\boldsymbol{\mu}$, which is however not one-to-one for an overcomplete vector of sufficient statistics.

[23] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008

Properties of the log-partition function. In terms of the variational representation, it is easy to see the following properties of $A(\theta)$:

a) Since we determine the maximum over a family of functions that are trivially convex in θ (linear), it follows that $A(\theta)$ is convex, too.[24]

b) Since the maximum in (26) is uniquely attained and the objective is continuous in μ and θ, as well as convex in θ, under mild technical conditions on the set of mean parameters over which we optimize, Danskin's theorem[25] states that $A(\theta)$ is differentiable with respect to θ, and the gradient is given by

$$\nabla_\theta A(\theta) = \mu(\theta), \tag{30}$$

the unique maximizing vector of mean parameters.

[24] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004

[25] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999

These properties will prove useful in our subsequent attempts at solving optimization problems involving the log-partition function.

Tractability. The preceding discussion allows us to draw general conclusions about the difficulty of the inference problem. First, the set of mean parameters itself poses a challenge, as it can be hard to specify all possible mean parameters explicitly. In particular, the size of the set will turn out to be problematic. Second, then entropy, which is only defined via the dual coupling, cannot in general be characterized explicitly, so it is unclear how to compute this function (as we will see, important exceptions exist).

Maximum A-Posteriori Estimation in Exponential Families

So far, we have discussed the log-partition function $A(\theta)$ in some detail. As we have seen, this function is closely related to the mean parameters, which in discrete graphical models correspond to marginal probabilities.

We shall also find it useful to introduce a similar function corresponding to the *maximization* problem, rather than marginalization. Consider

$$\mathring{A}(\theta) = \max_{\mathbf{y}\in\mathcal{Y}} \langle \theta, \phi(\mathbf{y}) \rangle. \tag{31}$$

The function so defined is clearly convex in θ, since it maximizes over a family of functions that are convex in θ.[26]

[26] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004

Unlike the log-partition function, we cannot in general assume that the maximum is attained uniquely (depending on the set of mean parameters), so $\mathring{A}(\theta)$ need not be differentiable. However, by standard results in convex optimization, a *subgradient* with respect to θ is given by

$$\mathbf{g}(\theta) = \phi(\hat{\mathbf{y}}), \quad \hat{\mathbf{y}} \in \arg\max_{\mathbf{y}} \langle \theta, \phi(\mathbf{y}) \rangle. \tag{32}$$

Again, this fact will be useful in optimization algorithms, both for inference in graphical models and for discriminative training. The observant reader may have noticed that $\mathring{A}$ is named suggestively close to the log-partition function A. Indeed, there exists a fundamental relationship between these two functions, as will be pointed out in Part II of this thesis.

Relation to Statistical Mechanics

Graphical models can also be interpreted from a statistical mechanics viewpoint, as has been pointed out by Yedidia et al.[27] We will now introduce a few of these concepts that are important to our discussion.

[27] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms. *IEEE Transactions on Information Theory*, 51:2282–2312, 2004

Statistical mechanics tells us that in thermal equilibrium, the probability of a joint state $\mathbf{y} = \{y_1, y_2, \ldots, y_v\}$ of a system of v particles, each of which can be in one of a discrete number of states, follows *Boltzmann's law*, viz.:

$$p(\mathbf{y}) = \frac{1}{Z_{\mathfrak{T}}} e^{-E(\mathbf{y})/\mathfrak{T}}, \tag{33}$$

where $\mathfrak{T}$ denotes the temperature and $Z_{\mathfrak{T}}$ is a partition function that normalizes the probability mass. We refer to the rich literature on statistical mechanics for further discussion on Boltzmann's law—of greater interest to us is the connection to exponential families. Under this viewpoint, Boltzmann's law defines the probability density of an exponential model in terms of an *energy* function $E(\mathbf{y})$. The temperature $\mathfrak{T}$ is inconsequential to our discussion, as it merely sets a scale for the unit of the energy, and will simply be assumed to be 1.

In the notation for exponential models we previously set forth, the energy of a state $\mathbf{y}$ is hence given by

$$E(\mathbf{y}; \boldsymbol{\theta}) = -\boldsymbol{\phi}(\mathbf{y})^\mathsf{T} \boldsymbol{\theta}. \tag{34}$$

Recall that $\boldsymbol{\phi}(\mathbf{y})$ denotes the sufficient statistics of our exponential family.

Likewise, the *Helmholtz free energy* of a system, $F_{\text{Helmholtz}}$, plays an important role in statistical mechanics. It is defined as

$$F_{\text{Helmholtz}} = -\log Z, \tag{35}$$

and can thus be seen to correspond the the negative log-partition function, that is $-A(\boldsymbol{\theta})$, in the exponential family framework.

Knowledge of these concepts helps understand the terminology used in different communities. For instance, computing a mode of a density,

$$\hat{\mathbf{y}}(\boldsymbol{\theta}) = \arg\max_{\mathbf{y}} p(\mathbf{y}; \boldsymbol{\theta}) = \arg\max_{\mathbf{y}} \log p(\mathbf{y}; \boldsymbol{\theta}) \tag{36}$$

$$= \arg\min_{\mathbf{y}} E(\mathbf{y}; \boldsymbol{\theta}), \tag{37}$$

is often referred to as *energy minimization*. Similarly, in computation of the log partition function, one often encounters references to the *free energy*.

Discriminative Models

We have seen that within a given exponential family, the distribution of a random vector can be specified completely in terms of the vector of exponential parameters associated with the sufficient statistics. We have also seen examples of the sufficient statistics of two types of graphical models, discrete MRFs and Gaussian MRFs.

In both cases, the distribution can be defined in terms of *factors* and the associated exponential parameters. In *discriminative* graphical models, the exponential parameters of the factors can depend on the observed input $\mathbf{x}$ (remember our abstract discussion of the structured prediction problem at the beginning of this chapter, cf. Figure 2).

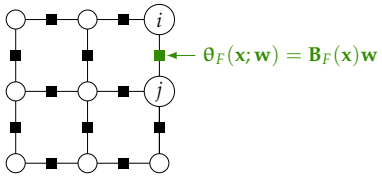


Figure 9: In a discriminative model, the exponential parameters of factor F depend on a subset of rows of feature matrix $\mathbf{B}$, as well as on model weights $\mathbf{w}$.

Y_s	Y_t	$\theta_F(\cdot,\cdot)$
0	0	0.3
0	1	1.4
⋮	⋮	⋮
y_s	y_t	$\theta_F(y_s, y_t) \leftarrow \mathbf{b}_{F;st}(\mathbf{x})^\mathsf{T}\mathbf{w}$
⋮	⋮	⋮

Figure 10: For discrete factors, the entries of the factor table arise as the inner product of a row of the feature matrix and the weights vector.

Linear Parameterization

We will consider linear models—parameterized by a vector $\mathbf{w} \in \Omega \subseteq \mathbb{R}^p$ of model parameters—that initialize the exponential parameters $\theta \subseteq \mathbb{R}^d$ from weighted sums of derived features or basis functions of input $\mathbf{x}$, via

$$\theta(\mathbf{x}; \mathbf{w}) = \mathbf{B}(\mathbf{x})\mathbf{w}, \tag{38}$$

or equivalently, making the factor structure explicit,

$$\theta_F(\mathbf{x}; \mathbf{w}) = \mathbf{B}_F(\mathbf{x})\mathbf{w}, \qquad F \in \mathcal{F}. \tag{39}$$

[28] Actually, the dimensionality of the output space $\mathcal{Y}$ can vary depending on input $\mathbf{x}$, such that $\theta \in \mathbb{R}^{d(\mathbf{x})}$, and the signature of $\mathbf{B}$ must be adjusted accordingly.

Here, $\mathbf{B}\colon \mathcal{X} \mapsto \mathbb{R}^{d \times p}$ is a matrix-valued function[28] returning the features derived from input $\mathbf{x}$, and we use $\mathbf{B}_F(\mathbf{x})$ to denote the subset of rows that apply to factor F. This is illustrated in Figures 9–10.

Typically, depending on the nature of the structured prediction task, factors are instantiated in a repetitive manner and $\mathbf{B}(\mathbf{x})$ is defined such that model parameters $\mathbf{w}$ are tied among factors of a common type.

Note that since the exponential parameters θ are in general restricted to a subset of $\mathbb{R}^d$, the model parameters $\mathbf{w}$, and possibly the features $\mathbf{B}(\mathbf{x})$, cannot be chosen completely freely. We denote this by requiring $\mathbf{w} \in \Omega$.

The above linear parameterization may seem restrictive at first; however, we want to emphasize that while the model is linear in $\mathbf{w}$, it can depend on the observed input $\mathbf{x}$ in an almost arbitrary manner—including highly non-linear functions. Indeed, in Part III of the thesis, we will use this fact to devise a model with non-parametric dependence on $\mathbf{x}$.

Conditional Random Fields

Recall that the exponential parameters $\theta(\mathbf{x}; \mathbf{w})$, now depending on input $\mathbf{x}$, define a probability distribution. This distribution is *conditional* on $\mathbf{x}$, and parameterized by the model parameters $\mathbf{w} \in \Omega$, as follows:

$$p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) = \exp\{\langle \theta(\mathbf{x}; \mathbf{w}), \phi(\mathbf{y}) \rangle + A(\theta(\mathbf{x}; \mathbf{w}))\} \tag{40}$$

$$= \exp\{-E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) - A(\theta(\mathbf{x}; \mathbf{w}))\}. \tag{41}$$

By defining the *conditional* energy

$$E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) = -\langle \theta(\mathbf{x}; \mathbf{w}), \phi(\mathbf{y}) \rangle.$$

We call such a distribution over a random vector a *conditional random field*,[29] or CRF for short. The principal advantage of CRFs is that they let us handle structured prediction tasks within a probabilistic framework. We will elaborate on this point by discussing how to obtain predictions from a CRF, and how to estimate its parameters.

[29] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *International Conference on Machine Learning (ICML)*, 2001

Predicting

We already alluded to the fact that given the *true* posterior density $p(\mathbf{y} \mid \mathbf{x})$, the optimal decision-theoretic prediction with respect to a loss ℓ is given by

$$\hat{\mathbf{y}}_\ell(\mathbf{x}) = \operatorname*{arg\,min}_{\hat{\mathbf{y}} \in \mathcal{Y}} R_\ell(\hat{\mathbf{y}} \mid \mathbf{x}), \quad R_\ell(\hat{\mathbf{y}} \mid \mathbf{x}) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y} \mid \mathbf{x})}[\ell(\hat{\mathbf{y}}, \mathbf{y})]. \tag{42}$$

Conditional random fields provide us with a posterior density $p(\mathbf{y} \mid \mathbf{x}; \mathbf{w})$—estimated from training data in a manner that will be made precise shortly—and consequently, this approach is applicable in principle.

Compared to a general multi-variate distribution, remember that we are now operating on a graphical model. Consequently, a solution to (42) can often be found at reasonable computational expense, at least approximately.

As an example, assume we are working in a discrete model and want to find the prediction that minimizes 0-1 loss, defined as

$$\ell_{0\text{-}1}(\hat{\mathbf{y}}, \mathbf{y}) = \mathbb{I}_{\hat{\mathbf{y}} \neq \mathbf{y}}. \tag{43}$$

It can easily be verified that any mode of our posterior density constitutes an optimal prediction under the above optimality criterion. This makes sense intuitively—if we want to minimize the risk of predicting an incorrect joint labeling, we should pick the most likely state. Computing maximum a-posteriori states in discrete graphical models is NP-hard in general[30], however, it is a well-studied problem, and a multitude of algorithms exploiting the structure of the graph exist, both for those cases where exact inference is feasible, as well as for the remaining cases that need to be handled approximately.[31]

[30] Venkat Chandrasekaran, Nathan Srebro, and Prahladh Harsha. Complexity of Inference in Graphical Models. Technical report, 2010

[31] Indeed, we will present our own algorithm towards this end in Part II of the thesis.

Decomposition of loss. More generally, the problem can often be solved efficiently if the loss function decomposes over the variables. In that case, the expected loss of a prediction $\hat{\mathbf{y}}$ factors in terms of marginal distributions,

$$\begin{aligned} R_\ell(\hat{\mathbf{y}} \mid \mathbf{x}) &= \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y}|\mathbf{x})}[\ell(\hat{\mathbf{y}}, \mathbf{y})] && (44) \\ &= \sum_{s \in V} \int_{\mathcal{Y}_s} p_s(y_s \mid \mathbf{x}) \ell_s(\hat{y}_s, y_s) \, \nu(\mathrm{d}y_s) && (45) \\ &= \sum_{s \in V} R_{\ell;s}(y_s \mid \mathbf{x}), && (46) \end{aligned}$$

and the optimal prediction can be found individually for each variable,

$$\hat{y}_{\ell;s}(\mathbf{x}) = \operatorname*{arg\,min}_{y_s \in \mathcal{Y}_s} R_{\ell;s}(\hat{y}_s \mid \mathbf{x}), \qquad s \in V. \tag{47}$$

This decomposition replaces a single complex optimization problem by a large number of primitive optimization problems.

For instance, in a discrete MRF, we only need to minimize over the finite (and typically small) label space of individual variables. A typical example is the so-called *Hamming* loss, defined as

$$\ell_{\text{Hamming}}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{s \in V} \mathbb{I}_{\hat{y}_s \neq y_s}. \tag{48}$$

This loss can be minimized by picking for each variable individually the state that maximizes the posterior marginals.

A similar example for the case of continuous random variables is the *mean squared error* (or MSE), measured via

$$\ell_{\text{MSE}}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{s \in V} (\hat{y}_s - y_s)^2. \tag{49}$$

Since the loss decomposes over the variables, the optimum can again be obtained in terms of the posterior marginal distributions by computing for each variable its conditional expectation, i.e. $\hat{y}_{\ell;s}(\mathbf{x}) = \mathbb{E}_{y_s \sim p(y_s|\mathbf{x})}[y_s]$.[32]

[32] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007

Evidently, the above approach depends crucially on our ability to compute marginal distributions. The situation is broadly the same as for maximum a-posteriori inference: for discrete models, marginalization is NP-hard in general, but good approximate algorithms exist.

Maximum Likelihood Estimation

Consider now the problem of estimating the model parameters $\mathbf{w}$ given n labelled i.i.d. training examples $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\}$. The prevailing approach in CRF training is to maximize the *conditional* likelihood of the data:

$$\hat{\mathbf{w}}_{\text{ML}} = \arg\max_{\mathbf{w}\in\Omega} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}). \tag{50}$$

Commonly, a zero-mean, spherical Gaussian prior $p(\mathbf{w})$ on the model parameters is assumed, such that we actually seek the maximum a-posteriori estimate of $\mathbf{w}$:

$$\hat{\mathbf{w}}_{\text{MAP}} = \arg\max_{\mathbf{w}\in\Omega} \left\{ p(\mathbf{w}) \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) \right\}. \tag{51}$$

Computationally, it is more convenient to minimize the negative log-likelihood of the data. Observe that

$$\begin{aligned} -\log p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) &= -\langle \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}), \boldsymbol{\phi}(\mathbf{y}) \rangle + A(\boldsymbol{\theta}(\mathbf{x}; \mathbf{w})) & (52) \\ &= E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) + A(\boldsymbol{\theta}(\mathbf{x}; \mathbf{w})) & (53) \end{aligned}$$

and

$$-\log p(\mathbf{w}) = -\frac{C}{2} \|\mathbf{w}\|_2^2 \; \cancel{+ \mathit{const}}, \tag{54}$$

where C is inversely proportional to the variance of the spherical Gaussian prior over the model parameters. The objective hence turns into

$$\mathcal{O}_{\text{CRF}}(\mathbf{w}) = \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \big[E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) + A(\boldsymbol{\theta}(\mathbf{x}; \mathbf{w})) \big], \tag{55}$$

and the corresponding optimization problem is convex, subject to $\mathbf{w} \in \Omega$ (for discrete models, $\Omega = \mathbb{R}^d$, so the problem is unconstrained). To see convexity, first note that any norm is a convex function.[33] Moreover, the conditional energy is linear in $\mathbf{w}$, and hence convex. Finally, the convexity of $A(\boldsymbol{\theta})$ in $\boldsymbol{\theta}$ was established when we discussed its properties, and $\boldsymbol{\theta}(\mathbf{x}; \mathbf{w})$ is linear in $\mathbf{w}$, so the composition is again convex in $\mathbf{w}$.

[33] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004

Gradient. To actually minimize the objective, we will find it useful to derive the gradient with respect to the model parameters, given by

$$\nabla \mathcal{O}_{\text{CRF}}(\mathbf{w}) = C\mathbf{w} + \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} [\mathbf{B}(\mathbf{x})]^{\mathsf{T}} [\boldsymbol{\mu}(\mathbf{x}; \mathbf{w}) - \boldsymbol{\phi}(\mathbf{y})], \tag{56}$$

where $\boldsymbol{\mu}(\mathbf{x}; \mathbf{w}) \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x};\mathbf{w})}[\boldsymbol{\phi}(\mathbf{y})]$ denotes the mean parameters coupled to the exponential parameters $\boldsymbol{\theta}(\mathbf{x}; \mathbf{w})$. Recall that the mean parameters are generated by differentiating $A(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. In contrast, the sufficient statistics of the observed labeling $\mathbf{y}$ arise from differentiating the energy with respect to $\boldsymbol{\theta}$. Finally, note that $\boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) = \mathbf{B}(\mathbf{x})\mathbf{w}$ is itself a function of $\mathbf{w}$—the final gradient follows from the chain rule.

Intuitively, the gradient measures the impact of each feature on the mismatch between the expected sufficient statistics under our model, and the sufficient statistics actually observed on training data $\mathcal{D}$. In optimizing $\mathbf{w} \in \Omega$, the weights on the features are chosen to minimize this mismatch.

Tractability. Being able to compute this gradient depends crucially on our ability to compute the mean parameters of each example. Moreover, even just evaluating the objective function requires that we compute the log-partition function, the difficulty of which—as we saw—depends crucially on the set of mean parameters. Unlike for the purpose of prediction, it is not acceptable to simply compute the mean parameters or the log-partition function using *any* approximate inference algorithm, since we wish to maintain convexity of the objective.

In Part II of this thesis, we will show how this goal can be achieved for discrete models, whereas Part III will treat the same problem for Gaussian models. In fact, this is one of the main challenges considered in this thesis.

Misspecification

The decision-theoretic approach we outlined depends crucially on our ability to estimate the *true* posterior distribution $p(\mathbf{y} \mid \mathbf{x})$. We already mentioned that maximum likelihood estimation is in principle asymptotically *consistent*. The underlying assumption is that $\mathbf{y} \mid \mathbf{x} \sim p(\mathbf{y} \mid \mathbf{x}; \mathbf{w})$, i.e. follows a parametric family that is known up to its parameters θ. However, as we intimated, this assumption rarely holds up in machine learning practice, since the distribution of the data, much less the *conditional* distribution, is rarely known in advance and often does not follow the parametric family that was chosen. This situation is referred to as *misspecification*.[34] To make things worse, the notion of *consistency* itself is rather intricate for conditional random fields.[35]

[34] Halbert White. Maximum Likelihood Estimation of Misspecified Models. *Econometrica*, 50(1):1–25, 1982

[35] Mathieu Sinn and Pascal Poupart. Asymptotic Theory for Linear-Chain Conditional Random Fields. In *Artificial Intelligence and Statistics (AISTATS)*, pages 679–687, 2011

Empirical Risk of a Model

Assuming *misspecification*, decision-theory is no longer applicable, because we cannot represent the true posterior density. It is time to step back and consider our true goal again, which is to learn a map $\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w})$ that exposes low expected loss under the true distribution $p(\mathbf{x}, \mathbf{y})$,

$$R_\ell[\hat{\mathbf{y}}(\cdot)] = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}[\ell(\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}), \mathbf{y})]. \tag{57}$$

This expectation can be approximated empirically from the n i.i.d. training examples $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\}$ at our disposal, via

$$\tilde{R}_\ell[\hat{\mathbf{y}}(\cdot)] = \frac{1}{n} \sum_{(\mathbf{x},\mathbf{y})} \ell(\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}), \mathbf{y}). \tag{58}$$

The definition of the map $\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w})$ can in principle be constructed from the misspecified model posterior density $p(\mathbf{y} \mid \mathbf{x}; \mathbf{w})$ in an arbitrary manner. Most commonly, it is defined to return a mode of the density, or equivalently, a minimum energy realization:

$$\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}) = \arg\max_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) \tag{59}$$

$$= \arg\min_{\mathbf{y}} E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}). \tag{60}$$

Choosing $\mathbf{w} \in \Omega$ to minimize the empirical risk then draws the mode of the misspecified model posterior density towards the observed ground truth (for each training example), in the sense of loss function ℓ.

[36] Thomas P. Minka. Empirical Risk Minimization is an incomplete inductive principle. Technical report, MIT Media Lab, 2000

Criticism. Empirical risk minimization has been described as an "incomplete inductive principle" by Minka,[36] and in particular been criticized for being agnostic about sampling distributions. However, in our setting, the sampling distribution either does not follow a known parametric family, or it is intractable. In this regime, Minka's characterization of empirical risk minimization as a "maximally vague model, which assumes nothing beyond the training data," sounds appealing rather than problematic.

Generalized notion of losses. The notion of a loss function $\ell\colon \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ that assigns a loss to a *prediction*, i.e. a minimum energy realization under our model according to the previous development, can be extended such that the loss function is defined in terms of the energies the model assigns to *any* realization. The signature of such loss functions is then $\ell\colon \mathcal{Y} \times \mathcal{X} \times \Omega \mapsto R$.

Note that this is a strict generalization of the previous definition, since the prediction can always be obtained as a function of the energies the model assigns to the possible realizations of the random vector. We will use $\ell(\mathbf{y} \mid \mathbf{x}; \mathbf{w})$ to denote the loss a model parameterized by $\mathbf{w}$ incurs on ground truth $\mathbf{y}$ given the observed input $\mathbf{x}$.

Loss of model $\mathbf{w}$ for ground truth y

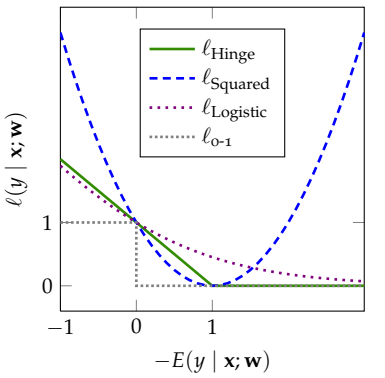


Figure 11: **Binary classification:** We plot the loss incurred by a model on an example as a function of the energy the model assigns to its ground truth y. The losses considered are defined as

$$\ell_{\text{Hinge}}(y|\mathbf{x};\mathbf{w}) = \max(0, 1 + E(y|\mathbf{x};\mathbf{w}))$$
$$\ell_{\text{Squared}}(y|\mathbf{x};\mathbf{w}) = (1 + E(y|\mathbf{x};\mathbf{w}))^2$$
$$\ell_{\text{Logistic}}(y|\mathbf{x};\mathbf{w}) = \log_2(1 + e^{E(y|\mathbf{x};\mathbf{w})})$$
$$\ell_{\text{0-1}}(y|\mathbf{x};\mathbf{w}) = [\![-E(y|\mathbf{x};\mathbf{w}) < 0]\!],$$

and the energy is given by

$$E(y|\mathbf{x};\mathbf{w}) = -y\langle \mathbf{b}(\mathbf{x}), \mathbf{w}\rangle.$$

Relation to Log-Likelihood. The above extended notion of loss functions opens up the possibility of analyzing various parameter estimation approaches in the empirical risk minimization framework. For instance, the negative log-likelihood of the training data, optimized during maximum likelihood estimation, can be understood as a convex *surrogate* for the empirical risk under 0-1 loss. This is depicted for the special case of binary classification in Figure 11. The logistic loss, which specializes the negative log-likelihood to binary classification, forms an upper bound on the 0-1 loss applied to the prediction of the model.

An important consequence of the above insight is that even in the regime of *misspecification*, maximum likelihood estimation is in fact a sound approach as long as we wish to minimize 0-1 loss and define the map $\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w})$ so as to return a mode of our misspecified posterior density.

Other convex surrogates. More generally, if we want to minimize the risk with respect to an arbitrary loss function ℓ, maximum likelihood estimation is no longer an appropriate approach. However, a different approach, *Max-Margin Markov Networks*,[37] or M3Ns for short, can be used to construct a convex surrogate for *any* loss function. A similar approach called *Softmax-Margin CRFs*[38] has recently been introduced that "injects" a loss in maximum likelihood estimation; however, it has not yet gained a momentum comparable to M3Ns, so we will focus on the latter in the following. Convex surrogate losses have been studied in great detail by Bartlett et al.[39]

[37] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-Margin Markov Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2003

[38] Kevin Gimpel and Noah A. Smith. Softmax-Margin CRFs: Training Log-Linear Models with Cost Functions. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010

[39] Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, Classification, and Risk Bounds. *Journal of the American Statistical Association*, 101(473):138–156, March 2006

Max-Margin Markov Networks

Max-margin Markov networks (M3Ns) are generally considered a non-probabilistic model. However, as we will point out later in this thesis, they are in fact intimately related to conditional random fields through choice of a specific loss function.

We will motivate M3Ns in terms of the *energy* the model assigns to realizations $\mathbf{y}$. Remember that in our discriminative setting, the energy depends on the observed input $\mathbf{x}$, as follows:

$$E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) = -\langle \theta(\mathbf{x}; \mathbf{w}), \phi(\mathbf{y}) \rangle. \tag{61}$$

The lower the energy, the higher the likelihood of a realization $\mathbf{y}$ of our random vector $\mathbf{Y}$ in an exponential model.

Constraint formulation. They key idea is now to choose the model parameters $\mathbf{w} \in \Omega$ such that given i.i.d. training data $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\}$, our model assigns to each observed realization $\mathbf{y}$ an energy that is lower than that of any other realization $\hat{\mathbf{y}}$, by a margin that corresponds to the loss $\ell(\hat{\mathbf{y}}, \mathbf{y})$ that would be incurred by predicting $\hat{\mathbf{y}}$. Formally, the constraints are

$$E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) \leq E(\hat{\mathbf{y}} \mid \mathbf{x}; \mathbf{w}) - \ell(\hat{\mathbf{y}}, \mathbf{y}), \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{D}, \hat{\mathbf{y}} \in \mathcal{Y}. \tag{62}$$

Note that this is in fact equivalent to demanding that

$$E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) \leq \min_{\hat{\mathbf{y}}} \{E(\hat{\mathbf{y}} \mid \mathbf{x}; \mathbf{w}) - \ell(\hat{\mathbf{y}}, \mathbf{y})\}, \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathcal{D}, \tag{63}$$

i.e. , that the energy of each observed realization be lower than the *lowest* energy of any other realization.

Regularized risk function. In general, it is impossible to meet all constraints. Therefore, we introduce a slack variable $\zeta_{(\mathbf{x},\mathbf{y})}$ for each example by which we penalize constraint violation, and form the optimization problem

$$\begin{aligned} \underset{\mathbf{w} \in \Omega, \zeta}{\text{minimize}} \quad & \frac{C}{2} \|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}} \zeta_{(\mathbf{x},\mathbf{y})} \\ \text{s.t.} \quad & E(\mathbf{y}|\mathbf{x}; \mathbf{w}) - \zeta_{(\mathbf{x},\mathbf{y})} \leq \min_{\hat{\mathbf{y}}} \{E(\hat{\mathbf{y}}|\mathbf{x}; \mathbf{w}) - \ell(\hat{\mathbf{y}}, \mathbf{y})\}, \ \forall (\mathbf{x}, \mathbf{y}). \end{aligned} \tag{64}$$

The squared norm on the model parameters are a regularization term that can be tuned using the hyper parameter C so as to trade off model complexity against constraint violation.[40]

A key observation here is that, by virtue of regularization, the inequalities are always tight at the optimum. We can thus eliminate the slack variables, turning our problem into minimization of a regularized risk function:

$$\underset{\mathbf{w} \in \Omega}{\text{minimize}} \quad \frac{C}{2} \|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \left(E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) + \max_{\hat{\mathbf{y}}} \{\ell(\hat{\mathbf{y}}, \mathbf{y}) - E(\hat{\mathbf{y}} \mid \mathbf{x}; \mathbf{w})\} \right). \tag{65}$$

Note that we replaced the inner minimization by an equivalent maximization problem, which we refer to as the *loss-augmented inference* problem.

Loss-augmented inference. Observe that modulo the loss term, the inner problem amounts to finding a maximum a-posteriori realization, or equivalently, a minimum energy realization. If the loss function decomposes over the sufficient statistics in terms of a map $\mathbf{e} \colon \mathcal{Y} \mapsto \mathbb{R}^d$ via

$$\ell(\hat{\mathbf{y}}, \mathbf{y}) = \langle \mathbf{e}(\mathbf{y}), \phi(\hat{\mathbf{y}}) \rangle, \tag{66}$$

then the problem is exactly as hard as MAP inference in our model.

[40] In the case of binary classification and linearly separable data, this choice of regularizer has the additional interpretation of choosing the hyperplane that maximizes the margin (among all hyperplanes that separate the data).

Interestingly, the objective of problem (65) can then be written in terms of the "maximum a-posteriori" function $\mathring{A}(\theta)$, defined in (31):

$$\mathcal{O}_{\text{M3N}}(\mathbf{w}) = \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \left[E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) + \mathring{A}(\theta(\mathbf{x}; \mathbf{w}) + \mathbf{e}(\mathbf{y}))\right]. \tag{67}$$

We can hence use the properties of $\mathring{A}(\theta)$ that are already known to us. In particular, it is straightforward to see that just like the negative log-likelihood, the objective function is convex in $\mathbf{w}$. However, the differentiability of $\mathcal{O}_{\text{M3N}}(\mathbf{w})$ depends on the exponential family. In particular, for a discrete MRF, the objective function is non-smooth.

Subgradient. Even if the objective function is non-differentiable, we can still find a subgradient with respect to $\mathbf{w}$,

$$\mathbf{g}_{\text{M3N}}(\mathbf{w}) = C\mathbf{w} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} [\mathbf{B}(\mathbf{x})]^{\mathsf{T}} [\boldsymbol{\phi}(\hat{\mathbf{y}}_{\mathbf{e}}(\mathbf{x}; \mathbf{w})) - \boldsymbol{\phi}(\mathbf{y})], \tag{68}$$

where $\hat{\mathbf{y}}_{\mathbf{e}}(\mathbf{x}; \mathbf{w})$ denotes any maximizer of the loss-augmented inference problem. One can then employ a variety of non-smooth convex optimization algorithms that make use of subgradients, e.g. bundle methods.[41]

[41] Choon Hui Teo, S. V. N. Vishwanathan, Alex Smola, and Quoc V. Le. Bundle Methods for Regularized Risk Minimization. *Journal of Machine Learning Research*, 11:311–365, 2010

Tractability. Evidently, the tractability of M3Ns depends crucially on our ability to solve the loss-augmented inference problem. This, in turn, is typically only feasible if the loss function decomposes according to (66). But even then, computation of a subgradient requires that we be able to solve the maximum a-posteriori problem *exactly*. This is not in general possible in discrete MRFs. A possible remedy is to relax the loss-augmented inference problem, i.e. to solve a related but simpler problem, which we *can* solve exactly so as to maintain convexity. We will discuss this approach in great detail in Part II of the thesis.

Direct Risk Minimization

The observant reader may wonder why we go to great lengths to optimize convex surrogate losses, rather than directly optimize the empirical risk (58) of the model. There are several reasons for this:

- First, the loss we are interested in is often neither smooth nor convex, as in the case of $\ell_{\text{0-1}}$. This makes optimization infeasible.
- Second, even if the loss function is smooth and convex, the overall empirical risk is typically not convex, since it involves the prediction under the model as a function of the model parameters.
- Finally, it is often difficult to differentiate the prediction under the model with respect to the model parameters.

Nonetheless, direct risk minimization has repeatedly been considered in the literature.

For discrete MRFs, the situation is particularly difficult, since the natural loss functions, such as $\ell_{\text{0-1}}$ and ℓ_{Hamming}, are typically non-convex and non-differentiable. In addition, MAP predictions under the model are a

non-smooth function of the model parameters, which makes optimization a challenging task. These two problems were addressed independently by Domke[42] and Stoyanov et al.[43] by a) smoothing the loss function, and b) through reverse-mode differentiation of a finite number of steps of an iterative, approximate *marginalization* algorithm. Hence, the prediction under the model is obtained in terms of marginal probabilities, rather than the MAP state. Of course, step a) implies that the optimized function is no longer the *true* empirical risk. Moreover, the approach is not convex in the model parameters, making optimization difficult.

In contrast, Gaussian MRFs are somewhat more amenable to direct risk minimization. As has been shown by Tappen,[44] the prediction under a Gaussian model can be differentiated with respect to the model parameters. Moreover, loss functions for continuous predictions are typically smooth and differentiable. By using a logistic loss function, it is also possible to train Gaussian MRFs such as to make them suitable for discrete binary prediction problems.[45] The issue of non-convexity remains, however.

[42] Justin Domke. Parameter learning with truncated message-passing. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2937–2943, 2011

[43] Veselin Stoyanov, Alexander Ropson, and Jason Eisner. Empirical Risk Minimization of Graphical Model Parameters Given Approximate Inference, Decoding, and Model Structure. In *Artificial Intelligence and Statistics (AISTATS)*, pages 725–733, 2011

[44] Marshall Tappen, Ce Liu, Edward Adelson, and William Freeman. Learning Gaussian Conditional Random Fields for Low-Level Vision. In *Computer Vision and Pattern Recognition (CVPR)*, 2007

[45] Marshall Tappen, Kegan Samuel, Craig Dean, and David Lyle. The Logistic Random Field—A convenient graphical model for learning parameters for MRF-based labeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2008

Discussion and Outlook

We have seen that in working with discriminative graphical models, there are two key problems we need to be able to solve efficiently:

- *Prediction:* Making optimal predictions amounts to being able to solve two basic inference tasks in our graphical model: maximum a-posteriori estimation (i.e. , computation of a mode of the posterior density), as well as marginalization (i.e. , computation of marginal distributions).
- *Training:* For maximum likelihood estimation, we need to be able to compute the mean parameters for each training example to obtain a gradient. M3Ns require that we be able to solve—for each training example—a loss-augmented inference problem, or equivalently, to compute a mode of the posterior density arising from augmented exponential parameters, if the loss function decomposes. The difficulty of direct risk minimization depends on how we obtain the prediction from the model.

Whether and how the required operations can be carried out efficiently depends on the exponential family. In part II of this thesis, we will consider discrete MRFs, whereas part III will deal with Gaussian MRFs.

Part II - Discrete Discriminative MRFs

Obtaining predictions from a discrete MRF is a difficult problem, unless the graph or the exponential parameters follow a particular structure. For the general case, both for marginalization and MAP estimation, we will introduce novel convergent message passing algorithms solving a convex objective function that can be understood to be a relaxation of the original problem. Both algorithms effectively perform a *reparameterization* of the original exponential parameters.

As far as training is concerned, we will focus on CRFs (i.e. maximum likelihood estimation) and M3Ns in Part II, since direct risk minimization in discrete models is associated with several problems, as discussed in the

previous section. For CRFs and M3Ns, we build on our notion of *convex relaxations* developed while introducing the new inference algorithms. We first discuss how the respective estimation problem can still be approached as described in this chapter, except that a convex relaxation of the original mean parameter computation task, or the loss-augmented inference task, is solved. Subsequently, building on the *reparametrization* perspective, we introduce several convex re-formulations of the relaxed estimation problems that do not require to repeatedly solve an inner optimization problem, but rather move this optimization task into the overall objective.

Part III - Gaussian Discriminative MRFs

In Gaussian MRFs, computation of marginal probabilities and modes is equivalent, as the unique mode is given precisely by the mean of the Gaussian bell curve. While a trivial analytic solution—in terms of a matrix inversion—exists, it is typically not feasible to actually invert that matrix, since its size can be on the order of $1,000,000 \times 1,000,000$. In this regime, iterative algorithms are by far preferable, and we introduce several methods towards that end.

In order to train discriminative Gaussian models, we will consider two approaches in detail. First, we will be concerned with Gaussian CRFs, that is, we will attempt to maximize the (conditional) likelihood of the data. This approach requires the mean parameters of each training example given the current model parameters. A major problem in this context is that the second-order moment matrix $\mathbb{E}_{\mathbf{y} \sim p(\mathbf{y}|\mathbf{x};\mathbf{w})}[\mathbf{y}\mathbf{y}^\mathsf{T}]$, of the size mentioned above, is typically not sparse. For this reason, exact maximum likelihood estimation is infeasible. Instead, we demonstrate how the conditional *pseudo-likelihood*[46] of the data can be optimized. This approach is asymptotically consistent, and the corresponding objective function is convex.

The second approach we follow is direct risk minimization. As opposed to Tappen et al.[47], our parameterization is more powerful, necessitating positive-definite constraints on the model parameters. We show how these constraints can be handled efficiently. Moreover, we generalize the approach of Tappen et al.[48] to discrete multi-label problems, using a multinomial logistic loss. This allows to handle discrete multi-label problems within a Gaussian framework, enabling extremely efficient predictions.

Discriminative Gaussian random fields form an elegant framework, but without fully exploiting the *conditional* dependency on the observed input, the limited expressiveness of the posterior density (which is uni-modal and symmetric) can become too restrictive. We will demonstrate how a Gaussian model can be made to depend on the observed input in a non-parametric manner, further adding to the expressiveness of the approach.

[46] Julian Besag. Efficiency of pseudo-likelihood estimation for simple Gaussian fields. *Biometrica*, 64(3):616–618, 1977

[47] Marshall Tappen, Ce Liu, Edward Adelson, and William Freeman. Learning Gaussian Conditional Random Fields for Low-Level Vision. In *Computer Vision and Pattern Recognition (CVPR)*, 2007

[48] Marshall Tappen, Kegan Samuel, Craig Dean, and David Lyle. The Logistic Random Field—A convenient graphical model for learning parameters for MRF-based labeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2008

Part II

Tractability through Convex Relaxations

Relaxed Computation of Marginals and Modes

In this part of the thesis, we will discuss ways of dealing with intractability in discrete graphical models. We will start with an overview of inference problems, discussing tractable special cases and characterizing the hardness of inference in a given graphical model. By understanding inference as an optimization problem, tractable *relaxations* arise naturally.

In the subsequent chapters of this part of the thesis, we will first introduce two novel inference algorithms that solve such relaxations of exact inference problems. Based on the insights gained en route, we will then proceed to discriminative training. Previously, we have seen that the popular approaches towards discriminative training are closely related to inference, in the sense that inference problems need to be solved repeatedly as part of parameter optimization. From various perspectives, we will address the question of what happens if these inference problems are replaced by their respective relaxation. Finally, we will consider several practical structured prediction tasks and compare different ways of handling intractability during training empirically.

Exact Inference in Discrete Graphical Models

In general, both computation of marginal probabilities, as well as computation of the most likely state of a discrete graphical model, are NP-hard. An in-depth analysis of the theoretical complexity of inference was conducted by Chandrasekaran et al.[49]

Rather than elaborate on this point, we will first give well-known examples of *specific* discrete graphical models that allow for efficient solution of marginalization and the MAP problem; if the graphical model of the reader falls into any of the below categories, this thesis is only of marginal[50] interest, as it discusses precisely the remaining cases:

- Perhaps most famously, inference in *tree-structured* graphs can be implemented efficiently using belief propagation.[51] Indeed, the difficulty of inference in a general graph depends crucially on how "tree-like" it is.
- Models with binary variables and *sub-modular* energies admit to find MAP states efficiently using graph cuts.[52]
- Finally, binary *planar* graphs allow for efficient MAP estimation using graph cuts, and marginalization using the Kasteleyn matrix.[53]

The identification of tractable subclasses of graphical models is currently an active research area. The methods we present in this part of the thesis are meant to be used for graphical models without structural constraints.

[49] Venkat Chandrasekaran, Nathan Srebro, and Prahladh Harsha. Complexity of Inference in Graphical Models. Technical report, 2010

[50] No pun intended.

[51] Judea Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *National Conference on Artificial Intelligence (AAAI)*, pages 133–136, 1982

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

[53] Nicol N. Schraudolph and Dmitry Kamenetsky. Efficient Exact Inference in Planar Ising Models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1417–1424, 2009

Tree-Structured Distributions

Before we move on to intractable graphical models, we will discuss the special case of tree-structured distributions for pedagogical reasons. The importance lies both in their use as a basic building block for the algorithms we shall introduce subsequently, as well as in the deep connection to the difficulty of an inference problem.

Notation. We will specialize our notation when working with trees. A direct consequence of the tree property is that the cliques of the graph comprise at most two vertices. Hence, the distribution is fully described in terms of factors over nodes and edges. Without loss of generality, we will assume there is precisely one factor per node, and one factor per edge. Each factor of a node $s \in V$ is associated with exponential parameters θ_s, while each factor of an edge $(s,t) \in E$ is parameterized by a vector θ_{st}.

Inference. Let us first consider the problem of computing the marginal probabilities $\mu_s = [p_s(y_s)]_{y_s \in \mathcal{Y}_s}$ of a single variable Y_s, and the marginals $\mu_{st} = [p_{st}(y_s, y_t)]_{(y_s,y_t) \in \mathcal{Y}_s \times \mathcal{Y}_t}$ of a pair of variables $Y_{st} = (Y_s, Y_t)$. With some abuse of notation, we will use the functional notation $\mu_s(y_s)$ and $\mu_{st}(y_s, y_t)$ to refer to the component of a vector of marginals that corresponds to a specific outcome.

A classical algorithm towards this end is *belief propagation*,[54] illustrated in Figure 12. Each message $\mathbf{m}_{s\to t}$ from node s to node t is defined recursively in terms of the messages $\mathbf{m}_{u\to s}$ by neighbors u of node s:

$$m_{s\to t}(y_t) = \frac{1}{Z} \sum_{y_s \in \mathcal{Y}_s} \exp[\theta_s(y_s) + \theta_{st}(y_s, y_t)] \prod_{(u \neq t,s)} m_{u\to s}(y_s). \tag{69}$$

These messages live in the same space as the marginal probabilities, and indeed, as we will see, they are intimately related. The messages must be passed in a specific order, first running from the leaves of the tree up to an arbitrarily chosen root, and then running back down to the leaves.

After all messages have been sent, the marginals (or the *belief*) at a node or an edge can be retrieved from the relations

$$\mu_s(y_s) = \frac{1}{Z} \exp \theta_s(y_s) \prod_{(t,s) \in E} m_{t\to s}(y_s) \tag{70}$$

and

$$\mu_{st}(y_s, y_t) = \frac{1}{Z} \exp \theta_{st}(y_s, y_t) \frac{\mu_s(y_s)}{m_{t\to s}(y_s)} \frac{\mu_t(y_t)}{m_{s\to t}(y_t)}. \tag{71}$$

These formulas can be understood as collecting the evidence of the sub-trees around a node or edge, as illustrated in Figure 13. This viewpoint makes clear how belief propagation works: By computing the messages in a specific order, it enables us to obtain the marginal probabilities in terms of these pre-computed messages, without recursively descending into the definition of the messages for each marginal vector we want to obtain.

Belief propagation is simply a specific instance of *dynamic programming*. Notably, if we only want to obtain a single marginal vector, it is no more efficient than recursively expanding the definitions (70)–(71). Moreover, it

[54] Judea Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *National Conference on Artificial Intelligence (AAAI)*, pages 133–136, 1982

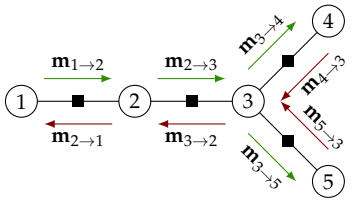


Figure 12: **Belief propagation:** For tree-structured distributions, marginal probabilities can be found using a sweep of messages passed from the leaves to an (arbitrary) root, and a subsequent sweep back down.

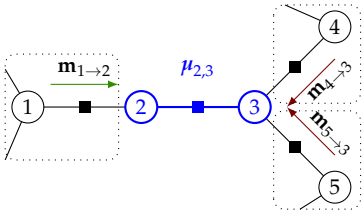


Figure 13: **Belief of a node or an edge:** Due to the conditional independencies in the tree, each belief depends on adjacent sub-trees solely in terms of the messages by immediate neighbors.

can be understood as a generalization of the well-known *forward-backward* algorithm in hidden Markov models (HMMs).[55] This insight suggests that an algorithm analogous to the Viterbi algorithm for HMMs, used to find the MAP state, may exist for trees. Indeed, this is the case.

By replacing the summation over $y_s \in \mathcal{Y}_s$ in (69) by a maximization over all $y_s \in \mathcal{Y}$, one obtains precisely an algorithm towards this end. If the maximum is non-unique in any of the message updates, one needs to collect back-pointers during the pass from the leaves the root, and follow these pointers back down to the leaves. Otherwise, the jointly optimal state can be found by picking, for each node, the state that maximizes the so-called *max-marginals* resulting from the altered update rule. Again, this is exactly analogous to the Viterbi algorithm for HMMs.

More generally, in both cases, the *distributive law* can be seen to be the common underlying principle.[56]

[55] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989

[56] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001

Factorization. A convenient property of tree-structured distributions is that the likelihood of a joint state $\mathbf{y}$ factors in terms of the marginals,

$$p(\mathbf{y}) = \prod_{s \in V} \mu_s(y_s) \prod_{(s,t) \in E} \frac{\mu_{st}(y_s, y_t)}{\mu_s(y_s)\mu_t(y_t)} = \frac{\prod_{(s,t)} \mu_{st}(y_s, y_t)}{\prod_s \mu_s(y_s)^{d_s - 1}}, \tag{72}$$

where d_s denotes the number of neighbors of node s. This is illustrated in Figure 14. As a consequence, the entropy of a distribution over a tree is a function of the marginal probabilities,

$$H(\boldsymbol{\mu}) = \sum_s H(\boldsymbol{\mu}_s) - \sum_{(s,t) \in E} I_{st}(\boldsymbol{\mu}_{st}) = \sum_{(s,t) \in E} H(\boldsymbol{\mu}_{st}) - \sum_s (d_s - 1) H(\boldsymbol{\mu}_s), \tag{73}$$

where $I(\boldsymbol{\mu}_{st}) = H(\boldsymbol{\mu}_s) + H(\boldsymbol{\mu}_t) - H(\boldsymbol{\mu}_{st})$ denotes the *mutual information* of variables Y_s and Y_t.

This is a generalization of the well-known result for Markov chains and will turn out to be extremely useful in the sequence.

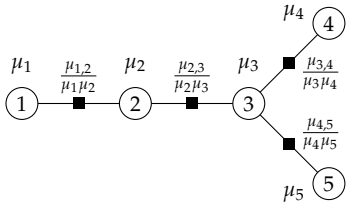


Figure 14: **Factorization of the likelihood:** Tree-structured distributions allow to factor the likelihood of a joint state in terms of the marginal probabilities of nodes and edges.

Junction Trees and Tree Width

More generally, given an arbitrary discrete random vector $\mathbf{Y}$, the distribution over which factors according to a cyclic graph structure, one may wonder whether the results we developed for the special case of trees are still applicable.

An important result in this context is that any structured probability distribution can be brought into the form of a tree, using the *junction tree* algorithm.[57] The algorithm works by clustering the vertices of the cyclic graph such that the *running intersection* property is fulfilled: Specifically, if clusters C_i and C_j both contain a vertex $s \in V$, then all clusters C_k of the junction tree in the unique path between C_i and C_j must contain s as well, as illustrated in Figure 15. This condition guarantees that marginal probabilities, as well as MAP states, can again be found using dynamic programming, via an algorithm that is very similar to belief propagation on regular trees.

There are two items of bad news associated with junction trees. First of all, the complexity of operations on the junction tree is exponential in

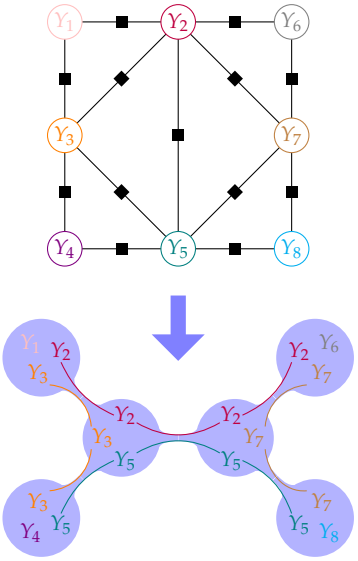


Figure 15: Cyclic graphs can be turned into junction trees by clustering the vertices such that the clusters satisfy the running intersection property (example adapted from `http://en.wikipedia.org/wiki/Tree_decomposition`).

[57] Steffen. L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50(2):157–224, 1988

the number of variables of the largest cluster, due to the combinatorial explosion of their joint state space. Second, the problem of finding an *optimal* junction tree, i.e. one that achieves the lowest possible cardinality of of the largest cluster, is NP-hard.[58] Besides, in the worst case, the largest cluster contains *every* variable of the random vector, in which case we have gained nothing.

[58] Finn V. Jensen and Frank Jensen. Optimal Junction Trees. In *Uncertainty in Artificial Intelligence (UAI)*, pages 360–366, 1994

The junction tree algorithm is hence only of practical importance if it is possible to greedily establish a junction tree with small clusters. In many applications for instance from computer vision, this is not the case—the clusters simply become too large to be practical. Nonetheless, junction trees provide us with important theoretical insights.

Tree width. Assume it is possible to find an optimal junction tree. We refer to the cardinality (the number of variables) of the largest cluster, minus one, as the *tree width* of the graph. The importance of the tree width lies in the fact that it provides a measure of the complexity of inference in a graph-structured distribution: In order to propagate belief through the junction tree, we need a number of operations that is *linear* in the number of clusters of the junction tree, but the complexity of these operations is *exponential* in the cardinality of the clusters.

Inference as Optimization

Since the junction tree algorithm is typically not a practical choice, we need different methods of dealing with discrete graphical models of high tree width. An important class of approaches, and indeed the dominant one in this thesis, considers inference as an optimization problem.

This viewpoint is perhaps most obvious for the problem of finding a maximum a-posteriori state. Clearly, by definition, this is a combinatorial optimization problem. However, as we have already seen, the log-partition function of any exponential family can also be defined in a *variational* manner, i.e. in terms of optimization over the set of mean parameters—cf. (26). We will now make this process concrete for the important special case of discrete MRFs. We will also provide an additional viewpoint in terms of minimization of the Kullback-Leibler (KL) divergence of a trial distribution.

The Marginal Polytope

Let us first consider the set of all realizable mean parameters, which is the space over which variational approaches to inference optimize. In discrete graphical models, this set exposes a particular structure, and is referred to as the *marginal polytope*.[59]

[59] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008

Remember that the mean parameters in a model with exponential parameters θ are defined as the expectation of the sufficient statistics, that is,

$$\boldsymbol{\mu}(\theta) = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta)}[\boldsymbol{\phi}(\mathbf{y})] = \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y};\theta)\boldsymbol{\phi}(\mathbf{y}). \tag{74}$$

Our use of the symbol $\boldsymbol{\mu}$, which we also used for marginal probabilities in the previous section, is not coincidental. As we intimated, in a discrete

model, the sufficient statistics of a joint state $\mathbf{y}$ expose the form of an indicator vector, the components of which are one if and only if the corresponding state of a factor is consistent with $\mathbf{y}$. From the above definition, we conclude that each component of $\boldsymbol{\mu}(\boldsymbol{\theta})$ is precisely a marginal probability.

Recall now the definition of the set of all realizable mean parameters of a graph G, repeated here for convenience:

$$\mathcal{M}(G) = \{\boldsymbol{\mu} \mid \exists p \text{ s.t. } \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}[\boldsymbol{\phi}(\mathbf{y})] = \boldsymbol{\mu}\}. \tag{75}$$

Let us characterize this set for the special case of discrete models. Following Wainwright and Jordan,[60] we will refer to this set as the *marginal polytope*, denoted by $\mathbb{M}(G)$, rather than $\mathcal{M}(G)$, to highlight the special case.

Fom our knowledge about the sufficient statistics in discrete MRFs, we can immediately draw the following conclusions:

a) The marginal polytope is a *convex* set, since by definition, it is the set of all convex combinations of a finite number of vectors, or equivalently, their convex hull $\text{conv}\{\boldsymbol{\phi}(\mathbf{y})\}_{\mathbf{y}\in\mathcal{Y}}$.

b) By the theorem of Minkowski-Weyl,[61] since the marginal polytope is finitely generated, it can equivalently be described as the intersection of a finite number of half-spaces.

The depiction of the marginal polytope in Figure 16, again due to Wainwright and Jordan, is illustrative, if idealized. In order to appreciate the complexity of this polyhedron, it is insightful to reason about the number of its facets. Koller and Friedman[62] state the following result:

c) The marginal polytope has an exponential number of facets in general.

Finally, Wainwright and Jordan[63] give two concrete examples related to the number of facets of the marginal polytope.

d) The marginal polytope of an Ising model on a complete graph with 7 nodes is known to have more than 2×10^8 facets.[64]

e) For trees, the junction tree theorem[65] guarantees that the number of facets grows only linearly in the number of nodes.

Clearly, these results indicate that in general, it will be difficult to work with the marginal polytope. Nonetheless, we will find it useful to consider optimization problems over the marginal polytope, since these provide us with a straightforward way of obtaining sound approximations.

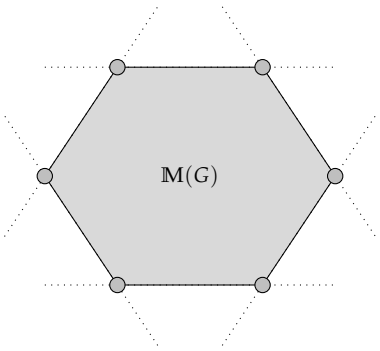


Figure 16: The marginal polytope is fully described by the intersection of a finite number of half spaces.

[60] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008

[61] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1998

[62] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009

[63] Martin J. Wainwright and Michael I. Jordan. Variational inference in graphical models: The view from the marginal polytope. In *Allerton Conference on Communication, Control, and Computing*, 2003

[64] Michel M. Deza and Monique Laurent. *Geometry of cuts and metric embeddings*. Springer Verlag, 1997

[65] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic networks and expert systems*. Statistics for Engineering and Information Science. Springer Verlag, 1999

The Log-Partition Function as the Conjugate Dual of the Negative Entropy

In the first part of the thesis, for exponential families in general, we already pointed out the close connection between the mean parameters $\boldsymbol{\mu}(\boldsymbol{\theta})$ and the log-partition function $A(\boldsymbol{\theta})$. We will now make this connection precise for the special case of discrete MRFs.

As our starting point, consider the *convex conjugate*[66] of the function $A(\boldsymbol{\theta})$, defined as

$$A^*(\boldsymbol{\theta}^*) = \sup_{\boldsymbol{\theta}} \big\{ \langle \boldsymbol{\theta}, \boldsymbol{\theta}^* \rangle - A(\boldsymbol{\theta}) \big\}, \quad \boldsymbol{\theta}, \boldsymbol{\theta}^* \in \mathbb{R}^d. \tag{76}$$

[66] Dimitri P. Bertsekas, Angelia Nedic, and Asuman E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003

Let us now work out the solution to this optimization problem, so as to obtain an analytic expression for the convex conjugate.
By differentiating the objective function in (76) with respect to θ, we obtain the stationary condition

$$\theta^* \stackrel{!}{=} \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta)}[\phi(\mathbf{y})], \tag{77}$$

that is, θ^* must be equal to the marginal probabilities arising from the vector of exponential parameters θ.[67]

[67] Recall that differentiating $A(\theta)$ yields the mean parameters, which correspond to marginal probabilities in a discrete MRF.

Which choice of θ satisfies this constraint? To answer this question, as shown by Wainwright and Jordan,[68] we need to distinguish three cases.

[68] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008

a) If θ^* is an interior point of the marginal polytope, that is, $\theta^* = \mu \in \mathbb{M}^\circ$, from our general discussion of mean parameters, we already know that there exists at least one $\theta(\mu)$ such that $\nabla A(\theta(\mu)) = \mu$ and the above constraint is satisfied. Consequently, we have

$$\begin{aligned} A^*(\mu) &= \langle \theta(\mu), \mu \rangle - A(\theta(\mu)) && (78)\\ &= \langle \theta(\mu), \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta(\mu))}[\phi(\mathbf{y})] \rangle - A(\theta(\mu)) && (79)\\ &= \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta(\mu))}[\langle \theta(\mu), \phi(\mathbf{y}) \rangle - A(\theta(\mu))] && (80)\\ &= \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta(\mu))}[\log p(\mathbf{y};\theta(\mu))] && (81)\\ &= -H(p_{\theta(\mu)}). && (82) \end{aligned}$$

   The convex conjugate is hence precisely the negative entropy of the distribution with marginal probabilities $\mu = \theta^* \in \mathbb{M}^\circ$.

b) Let $\overline{\mathbb{M}}$ the denote the *closure* of $\mathbb{M}$. If the dual point lies at the boundary of $\mathbb{M}$, that is, $\theta^* = \mu \in \overline{\mathbb{M}} \setminus \mathbb{M}^\circ$, then

$$A^*(\mu) = \lim_{n\to\infty} A^*(\mu^{(n)}), \tag{83}$$

   where $\{\mu^{(n)}\}$ is a sequence of interior points converging to μ.

c) Finally, if θ^* lies outside of $\mathbb{M}$, to be precise, $\theta^* \notin \overline{\mathbb{M}}$, then

$$A(\theta^*) = +\infty, \tag{84}$$

   which completes the three possible cases.

Conjugate of the conjugate. The above development fully characterizes the convex conjugate A^*. Importantly, since $A(\theta)$ is lower semi-continuous, $A = A^{**}$, that is, the log-partition function can in turn be described as the convex conjugate of its convex conjugate:

$$A(\theta) = \sup_{\theta^*} \{ \langle \theta^*, \theta \rangle - A^*(\theta^*) \}. \tag{85}$$

By the definition of A^*, for any $\theta^* \notin \overline{\mathbb{M}}$, the objective value is $-\infty$, and certainly not optimal. On the other hand, if $\theta^* = \mu \in \overline{\mathbb{M}}$, we know from our discussion of the map from exponential to mean parameters that the optimum is uniquely attained at an interior point $\mu = \mathbb{E}_{p(\mathbf{y};\theta)}[\phi(\mathbf{y})] \in \mathbb{M}^\circ$.

An equivalent, but more explicit characterization of the log-partition function is therefore

$$A(\theta) = \max_{\mu\in\mathbb{M}^\circ} \{ \langle \theta, \mu \rangle + H(p_{\theta(\mu)}) \}, \tag{86}$$

which is in line with the result we previously mentioned without proof.

Alternative Interpretation in Terms of the Kullback-Leibler Divergence

Above, we have demonstrated how the relationship between the log-partition function and the entropy of a distribution can be understood in terms of convex conjugacy. Here, we provide an additional characterization that does not require knowledge of convex analysis, due to Yedidia and Weiss.[69]

[69] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms. *IEEE Transactions on Information Theory*, 51:2282–2312, 2004

Again, through our restriction to exponential families, our distribution follows Boltzmann's law, viz.:

$$p(\mathbf{y};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} e^{-E(\mathbf{y};\boldsymbol{\theta})}. \tag{87}$$

Recall that the energy of a state is given by $E(\mathbf{y};\boldsymbol{\theta}) = -\langle \boldsymbol{\theta}, \boldsymbol{\phi}(\mathbf{y}) \rangle$. We wish to compute $A(\boldsymbol{\theta}) = -\log Z(\boldsymbol{\theta}) = F_{\text{Helmholtz}}$, which is deemed to be intractable.

We will choose to approximate p by means of a trial distribution b. Towards this end, consider the *Gibbs free energy*,

$$F_{\text{Gibbs}}(b) = F_{\text{Helmholtz}} + D(b\|p), \tag{88}$$

where

$$D(b\|p) \stackrel{\text{def}}{=} \sum_{\mathbf{y}\in\mathcal{Y}} b(\mathbf{y}) \log \frac{b(\mathbf{y})}{p(\mathbf{y})} \tag{89}$$

denotes the Kullback-Leibler (KL) divergence between b and p. From basic information-theoretic results, we know that $D(b\|q)$ is non-negative and zero if and only if $b = p$ almost everywhere. It follows that the Gibbs free energy is equal to the Helmholtz free energy if $b = p$ holds.

Consequently, computation of $F_{\text{Helmholtz}}$ can equivalently be achieved by solving the following optimization problem:

$$F_{\text{Helmholtz}} = \min_{b} F_{\text{Gibbs}}(b) \tag{90}$$

over all valid trial distributions b. Of course, without further restrictions on b, this is intractable as well. In any case, it is insightful to further consider this problem in order to see its relation to our previous development.

By expanding the definition of the Gibbs free energy, we obtain

$$\begin{aligned} F_{\text{Gibbs}}(b) &= F_{\text{Helmholtz}} + \sum_{\mathbf{y}} b(\mathbf{y}) \log b(\mathbf{y}) - \sum_{\mathbf{y}} b(\mathbf{y}) \log p(\mathbf{y}) && (91)\\ &= F_{\text{Helmholtz}} - H(b) - \sum_{\mathbf{y}} b(\mathbf{y})(F_{\text{Helmholtz}} - E(\mathbf{y})) && (92)\\ &= \cancel{F_{\text{Helmholtz}}} - H(b) - \cancel{F_{\text{Helmholtz}}} + \sum_{\mathbf{y}} b(\mathbf{y}) E(\mathbf{y}) && (93)\\ &= U(b) - H(b). && (94) \end{aligned}$$

defining $H(b) = -\sum_{\mathbf{y}} b(\mathbf{y}) \log b(\mathbf{y})$ and expanding $p(\mathbf{y}) = \exp(F_{\text{Helmholtz}} - E(\mathbf{y}))$

defining $U(b) = -\sum_{\mathbf{y}} b(\mathbf{y}) E(\mathbf{y})$

We refer to $U(b)$ as the *variational average energy* and to $H(b)$ as the *variational entropy*.

From our definition of the energy E, we obtain

$$F_{\text{Gibbs}} = -\sum_{\mathbf{y}} b(\mathbf{y}) \langle \boldsymbol{\theta}, \boldsymbol{\phi}(\mathbf{y}) \rangle - H(b). \tag{95}$$

Assume now that the trial distribution belongs to an exponential family and gives rise to marginal probabilities $\boldsymbol{\mu} \in \mathbb{M}^\circ$. We will refer to the distribution

as $b_{\boldsymbol{\theta}(\boldsymbol{\mu})}$ to make this connection explicit. The average energy can then equivalently be expressed in terms of these marginal probabilities, viz.:

$$U(b_{\boldsymbol{\theta}(\boldsymbol{\mu})}) = -\mathbb{E}_{\mathbf{y}\sim b_{\boldsymbol{\theta}(\boldsymbol{\mu})}}[\sum_F \langle \boldsymbol{\theta}_F, \boldsymbol{\phi}_F(\mathbf{y}_F)\rangle] = -\sum_F \langle \boldsymbol{\theta}_F, \boldsymbol{\mu}_F\rangle \tag{96}$$

$$= -\langle \boldsymbol{\theta}, \boldsymbol{\mu}\rangle. \tag{97}$$

Note that it is not in general possible to express the variational entropy in terms of the marginals, hence we use $H(b_{\boldsymbol{\theta}(\boldsymbol{\mu})})$ to denote the entropy corresponding to a given set of marginals.

We are now ready to re-state the result we previously obtained using tools from convex analysis in our KL-divergence framework:

$$A(\boldsymbol{\theta}) = -F_{\text{Helmholtz}} = -\min_b F_{\text{Gibbs}}(b) = \max_b -F_{\text{Gibbs}}(b) \tag{98}$$

$$= \max_{\boldsymbol{\mu}\in\mathbb{M}^\circ} \{\langle \boldsymbol{\theta}, \boldsymbol{\mu}\rangle + H(b_{\boldsymbol{\theta}(\boldsymbol{\mu})})\}. \tag{99}$$

We have thus shown how the variational inference problem can be derived from a different viewpoint. Evidently, however, this does not improve tractability of the problem: still, the constraint set $\mathbb{M}$ comprises an exponential number of half spaces, and there is not an explicit formula for computation of the entropy $H(b_{\boldsymbol{\theta}(\boldsymbol{\mu})})$.

Computation of a Mode in the Variational Framework

Our previous discussion focused on the log-partition function and its relation to the marginal probabilities. The second fundamental inference problem we consider in this thesis is *maximum a-posteriori* estimation, i.e. computation of a mode of the posterior density, or equivalently, one of the most likely joint states $\mathbf{y}$.

Remember that we defined this problem as

$$\mathring{A}(\boldsymbol{\theta}) = \max_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y};\boldsymbol{\theta}) = \max_{\mathbf{y}\in\mathcal{Y}} \log p(\mathbf{y};\boldsymbol{\theta}) \tag{100}$$

$$= \max_{\mathbf{y}\in\mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\phi}(\mathbf{y})\rangle. \tag{101}$$

An important insight is that since we are working in a discrete MRF, the state space of $\mathbf{y}$ is actually *finite*. Hence, one can equivalently think of the problem in terms of optimizing over a finite number of vectors of sufficient statistics $\{\boldsymbol{\phi}(\mathbf{y})\}_{\mathbf{y}\in\mathcal{Y}}$, leading us to

$$\mathring{A}(\boldsymbol{\theta}) = \max_{\{\boldsymbol{\phi}(\mathbf{y})\}_{\mathbf{y}\in\mathcal{Y}}} \langle \boldsymbol{\theta}, \boldsymbol{\phi}(\mathbf{y})\rangle. \tag{102}$$

The sufficient statistics consist of binary indicators, so this is an *integer linear program*.[70] Of course, we are optimizing over a number of vectors that is exponential in the number of variables.

[70] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997

Formulation as a linear program. We can use a standard result from linear programming to turn the above *combinatorial* optimization problem into a standard *convex* optimization problem. This has little practical value, since the asymptotic complexity will remain the same; however, it will provide us with a characterization of the problem that is easier to reason about.

Previously, we already noted that by the Minkowski-Weyl theorem, the convex hull of finitely many vectors is a bounded polyhedron. Consider the problem of optimizing our objective function over this polyhedron,

$$\max_{\text{conv}\{\boldsymbol{\phi}(\mathbf{y})\}_{\mathbf{y}\in\mathcal{Y}}} \langle \boldsymbol{\theta}, \boldsymbol{\phi}(\mathbf{y}) \rangle. \tag{103}$$

Any linear program over a polyhedron will attain an optimal solution at an *extreme point* of the polyhedron, assuming that the polyhedron has at least one extreme point and an optimal solution exists.[71] In our case, by definition, the polyhedron is precisely the marginal polytope $\mathbb{M}(G)$, the extreme points or vertices of which are the vectors of sufficient statistics. As a consequence, we obtain the following linear programming formulation of the maximum-a-posteriori problem,

$$\mathring{A}(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu}\in\mathbb{M}} \langle \boldsymbol{\theta}, \boldsymbol{\phi}(\mathbf{y}) \rangle, \tag{104}$$

and any extreme point solution to the above problem is precisely a maximum a-posteriori state. Again, it is important to stress the fact that the marginal polytope is defined by an exponential number of half-spaces, so solving this linear program is *not* tractable.

[71] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997

Connection to the log-partition function. The previous result allows us to see the connection between $\mathring{A}(\boldsymbol{\theta})$ and the log-partition function $A(\boldsymbol{\theta})$, from a statistical mechanics perspective. Towards this end, we will consider the log-partition function in the so-called "zero-temperature limit".

Recall that according to Boltzmann's law, at temperature $\mathfrak{T}$, the probability of a joint state $\mathbf{y}$ of a system of v particles, each of which can be in one of a discrete number of states, is given by

$$p(\mathbf{y}; \boldsymbol{\theta}) = \exp(-E(\mathbf{y}; \boldsymbol{\theta})/\mathfrak{T} - A_{\mathfrak{T}}(\boldsymbol{\theta})). \tag{105}$$

By expanding the variational representation of the log-partition function at this temperature, we obtain

$$\begin{aligned} A_{\mathfrak{T}}(\boldsymbol{\theta}) &= \sup_{\boldsymbol{\mu}\in\mathbb{M}} \left\{ \langle \boldsymbol{\theta}, \boldsymbol{\phi}(\mathbf{y}) \rangle / \mathfrak{T} - A^*(\boldsymbol{\mu}) \right\}. && (106) \\ &= A(\boldsymbol{\theta}/\mathfrak{T}). && (107) \end{aligned}$$

Finally, by letting the temperature go to zero, and re-scaling accordingly, we arrive at

$$\begin{aligned} \lim_{\mathfrak{T}\to 0} \mathfrak{T} A(\boldsymbol{\theta}/\mathfrak{T}) &= \lim_{\mathfrak{T}\to 0} \sup_{\boldsymbol{\mu}\in\mathbb{M}} \left\{ \langle \boldsymbol{\theta}, \boldsymbol{\phi}(\mathbf{y}) \rangle - \mathfrak{T} A^*(\boldsymbol{\mu}) \right\} && (108) \\ &= \max_{\boldsymbol{\mu}\in\mathbb{M}} \left\{ \langle \boldsymbol{\theta}, \boldsymbol{\phi}(\mathbf{y}) \rangle \right\} && (109) \\ &= \mathring{A}(\boldsymbol{\theta}). && (110) \end{aligned}$$

Intuitively, as the temperature decreases, the influence of the variational entropy $-A^*$ diminishes, and the solutions are gradually drawn towards an extreme point of the marginal polytope, until the problem eventually turns into maximum a-posteriori estimation—as depicted in Figure 17.

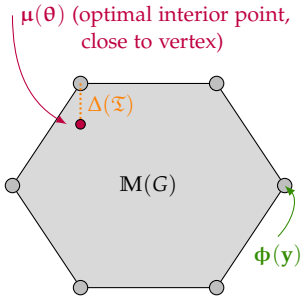


Figure 17: As $\mathfrak{T}$ decreases, the optimal mean parameters are gradually drawn towards a vertex, which corresponds to a vector of sufficient statistics.

Relaxing the Variational Problems

We have seen how the problem of exact inference (both marginalization and maximum a-posteriori estimation) in a graphical model can be understood as a convex optimization problem. Seemingly, then, we are mostly done, since general convex optimization problems can be solved efficiently using interior point methods.[72]

[72] Arkadi S. Nemirovski and Michael J. Todd. Interior-point methods for optimization. *Acta Numerica*, 17:191–234, 2008

Of course, this is not the case. As we pointed out, the problem lies in the fact that the marginal polytope, the constraint set over which we need to optimize, is described by an exponential number of facets in general. Moreover, for computation of the log-partition function and marginal probabilities, an additional complication stems from the lack of an explicit formula for the entropy corresponding to a given set of marginals.

Relaxations. The good news is that dealing with intractable constraint sets and objective functions has a long history in operations research and mathematical programming. In the sequence, we will focus on a principled approach based on *relaxations*[73] in particular.

[73] Arthur M. Geoffrion. Duality in Nonlinear Programming: A Simplified Applications-Oriented Development. *SIAM Review*, 13(1):1–37, 1971

Consider an optimization problem of the kind

$$\underset{\mathbf{x}\in\mathbb{X}}{\text{maximize}} \quad f(\mathbf{x}), \tag{111}$$

where both the constraint set $\mathbb{X}$ and the objective function f are intractable. A relaxation of the above problem is a similar optimization problem

$$\underset{\mathbf{x}\in\tilde{\mathbb{X}}}{\text{maximize}} \quad \tilde{f}(\mathbf{x}) \tag{112}$$

satisfying the properties

$$\mathbb{X} \subseteq \tilde{\mathbb{X}} \tag{113}$$

and

$$f(\mathbf{x}) \leq \tilde{f}(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{X}. \tag{114}$$

The first condition states that the feasible set of the original problem is a subset of the feasible set of the relaxed problem, while the second condition demands that for any feasible point of the original problem, the function value of the relaxed objective function must be greater than that of the original function.

Properties of relaxations. The above definition of relaxations is particularly useful because two important properties follow immediately:

a) The optimum of the relaxed problem is an upper bound on the optimum of the original problem,

$$\max_{\mathbf{x}\in\mathbb{X}} f(\mathbf{x}) \leq \max_{\mathbf{x}\in\tilde{\mathbb{X}}} \tilde{f}(\mathbf{x}). \tag{115}$$

To see this, note that for any optimum that is feasible according to the original constraint set, this is true by definition; moreover, we are now optimizing over a larger set of points, so the optimum must be at least as large as if we were optimizing over the original constraint set.

b) If, in addition to the above assumptions, $f(\mathbf{x}) = \tilde{f}(\mathbf{x})\ \forall \mathbf{x} \in \mathbb{X}$, then any optimal point of the relaxed problem is also a maximizer of the original problem if it lies within the original constraint set,

$$\bar{\mathbf{x}} \in \arg\max_{\mathbf{x} \in \tilde{\mathbb{X}}} \tilde{f}(\mathbf{x}) \text{ and } \bar{\mathbf{x}} \in \mathbb{X} \quad \Rightarrow \quad \bar{\mathbf{x}} \in \arg\max_{\mathbf{x} \in \mathbb{X}} f(\mathbf{x}). \tag{116}$$

Again, this follows directly—no solution to the original problem can achieve a better optimum, since it optimizes over a smaller space of possible solutions. Hence, in some cases, it is possible to obtain a certificate of optimality for the solution of the relaxed problem.

We will now specialize this framework to inference in discrete graphical models—it is applicable both to marginalization and MAP estimation.

Giving up Global Consistency

Let us first consider the marginal polytope and find a relaxed constraint set in the above sense. Towards this end, it is helpful to try and characterize the marginal polytope explicitly for a *tree-structured* graph T.

Assume a hypothetical vector $\boldsymbol{\tau}$—what are the requirements for this vector to consist of realizable marginal probabilities? First, marginal probabilities must be non-negative, so we have $\boldsymbol{\tau} \geq \mathbf{0}$. Second, by definition, marginal probabilities must always sum to one. Finally, as their name indicates, the marginal probabilities of a tree need to satisfy various marginalization constraints between nodes and edges. Putting these *local* constraints together, we obtain[74]

$$\mathbb{L}(T) = \left\{ \boldsymbol{\tau} \geq \mathbf{0} \;\middle|\; \begin{aligned} \sum_{y_s} \tau_s(y_s) &= 1, && \forall s \in V \\ \sum_{y_t} \tau_{st}(y_s, y_t) &= \tau_s(y_s), && \forall y_s \in \mathcal{Y}_s, (s,t) \in E \\ \sum_{y_s} \tau_{st}(y_s, y_t) &= \tau_t(y_t), && \forall y_t \in \mathcal{Y}_t, (s,t) \in E \end{aligned} \right\}. \tag{117}$$

[74] One might explicitly require that the edge marginals sum to one, this is however redundant due to the marginalization constraints.

Are these constraints sufficient to ensure $\boldsymbol{\tau}$ is a *realizable* vector of marginal probabilities? As Wainwright and Jordan[75] show, for a tree, $\mathbb{L}(T)$ is indeed equivalent to our previous characterization of the marginal polytope as the convex hull of the vectors of sufficient statistics, $\mathbb{M}(T) = \operatorname{conv}\{\boldsymbol{\phi}(\mathbf{y})\}_{\mathbf{y} \in \mathcal{Y}}$.

[75] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008

Clearly, any valid set of marginal probabilities must satisfy at least these *local* consistency constraints, so $\mathbb{M}(T) \subseteq \mathbb{L}(T)$. To establish equivalence, one also needs to show the reverse inclusion $\mathbb{L}(T) \subseteq \mathbb{M}(T)$. For a tree, this follows from the particular factorization of the distribution, cf. (72).

Cyclic graphs. For a cyclic graph G, the reverse inclusion does not necessarily hold, so $\mathbb{M}(G) \subset \mathbb{L}(G)$. A vector $\boldsymbol{\tau} \in \mathbb{L}(G)$ is then possibly only a *pseudo-marginal*, since there is not a distribution p that realizes $\boldsymbol{\tau}$. However, $\mathbb{L}(G)$, called the *local polytope* by Wainwright and Jordan, provides us with a useful outer approximation as required by relaxations.

The local polytope is illustrated in Figure 18. It is important to point out that—as Wainwright and Jordan note—the illustration is highly idealized. In particular, the number of facets of the local polytope is in fact smaller than that of the marginal polytope (after all, this is the primary motivation

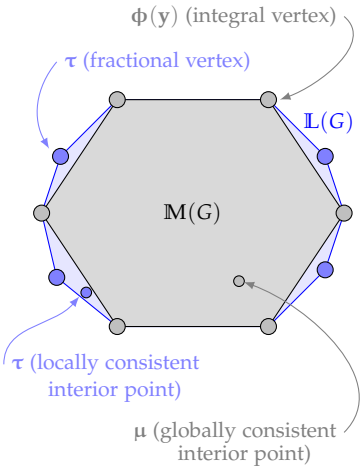


Figure 18: The local polytope $\mathbb{L}(G)$ is an outer approximation of the marginal polytope $\mathbb{M}(G)$.

behind the approximation), but this cannot be displayed in 2D. Compared to the marginal polytope, the local polytope contains additional points that adhere to local marginalization constraints, but cannot arise from a valid probability distribution. In particular, it contains *fractional vertices* that do not correspond to a sufficient statistics vector of any joint state $\mathbf{y}$.

Extension to larger factors. The local polytope can be readily extended to distributions involving factors $F \in \mathcal{F}$ over more than two variables. In this case, the marginalization constraints simply need to hold for all variables of a factor, as follows:

$$\mathbb{L}(G) = \left\{ \boldsymbol{\tau} \geq \mathbf{0} \;\middle|\; \begin{array}{ll} \sum_{y_s} \tau_s(y_s) = 1, & \forall s \in V \\ \sum_{\mathbf{y}_F \sim y_s} \tau_F(\mathbf{y}_F) = \tau_s(y_s), & \forall y_s \in \mathcal{Y}_s, s \in F, F \in \mathcal{F} \end{array} \right\}. \quad (118)$$

We use the notation $\mathbf{y}_F \sim y_s$ to denote all states $\mathbf{y}_F \in \mathcal{Y}_F$ of a factor F that are consistent with $Y_s = y_s$.

Application to MAP estimation. The local polytope enables us to define a relaxation of the exact MAP problem,

$$\mathring{A}_{\mathrm{LP}}(\boldsymbol{\theta}) = \max_{\boldsymbol{\tau} \in \mathbb{L}(G)} \langle \boldsymbol{\theta}, \boldsymbol{\tau} \rangle. \quad (119)$$

Compared to the original problem (104), we now optimize over a simpler constraint set involving only $O(|V| + |F|)$ constraints. Note that the objective function remains unchanged. Hence, from what we learned about relaxations, if the optimal $\boldsymbol{\tau}(\boldsymbol{\theta}) \in \mathbb{M}(G)$, we found an optimal solution to the exact MAP problem. In general, however, we must expect the solution to be a fractional vertex that must be rounded back to a valid state.

Wainwright and Jordan[76] refer to the above problem as the *first-order LP relaxation*, since it enforces marginalization constraints only between factors and single variables (rather than, say, between larger clusters). It is practical to solve this problem even for graphs of substantial size. While one might simply use an off-the-shelf linear programming solver, it is desirable to design algorithms exploiting the specific structure of the problem. A popular class of algorithms towards this end is max-product message passing. As shown by Yanover et al.,[77] such algorithms are by far more efficient for this problem than industrial-strength linear programming solvers.

However, many message passing algorithms do not actually solve the problem, but can get stuck at sub-optimal solutions. As we will see, this makes them rather unsuitable for use in discriminative training. Consequently, we will introduce our own message passing algorithm towards this end in the chapter to follow.

[76] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008

[77] Chen Yanover, Talya Meltzer, and Yair Weiss. Linear Programming Relaxations and Belief Propagation - An Empirical Study. *Journal of Machine Learning Research*, 7:1887–1907, 2006

Region-based Entropy Approximations

So far, we have obtained a tractable relaxation of the maximum a-posteriori problem. Towards this end, we introduced a tractable outer approximation of the marginal polytope, the so-called local polytope. The local polytope

will also prove to be useful in obtaining relaxations of the marginalization problem; however, for this purpose, we need one additional ingredient since the objective function itself is intractable.

Remember that besides the marginal polytope, the entropy term in

$$A(\theta) = \max_{\mu \in \mathbb{M}^\circ} \left\{ \langle \theta, \mu \rangle + H(p_{\theta(\mu)}) \right\} \tag{120}$$

is an additional source of intractability. The problem is that the entropy is only defined *implicitly*, as the entropy of the distribution that realizes the marginals μ.

Bethe approximation. One may be tempted to again consider the special case of trees first, and try to generalize it to a valid relaxation for cyclic graphs. Indeed, as we saw at the beginning of the chapter, tree-structured distributions admit a particular factorization of the entropy,

$$H(p_{\theta(\mu)}) = H(\mu) = \sum_{(s,t) \in E} H(\mu_{st}) - \sum_{s \in V} (d_s - 1) H(\mu_s). \tag{121}$$

By assuming that this decomposition holds for *any* graph, and relaxing the marginal polytope as previously, so that only local consistency is enforced, one obtains the *Bethe*[78] approximation of the log-partition function:[79]

$$A_{\text{Bethe}}(\theta) = \max_{\tau \in \mathbb{L}(G)} \left\{ \langle \theta, \tau \rangle + \sum_{(s,t) \in E} H(\tau_{st}) - \sum_{s \in V} (d_s - 1) H(\tau_s) \right\}. \tag{122}$$

Indeed, as Yedidia et al. show, this is precisely the objective function *loopy belief propagation*[80] seeks to optimize. Can we use the Bethe approximation as a valid relaxation of the marginalization problem, in the sense we previously outlined? Unfortunately, the answer to that question is negative. First of all, unlike the true entropy, the Bethe approximation to the entropy is not concave. In practice, this means that the above problem cannot be solved exactly. Second, the Bethe approximation does not form an upper bound on the true entropy.

Concave entropy approximations. There has been considerable interest in obtaining concave entropy approximations. One obvious rationale is that the true entropy is concave itself. For a class of approximations that decomposes the true entropy over variables and factors, via

$$\tilde{H}(\tau) = \sum_{s \in V} c_s H(\tau_s) + \sum_{F \in \mathcal{F}} c_F H(\tau_F), \tag{123}$$

Heskes[81] derives the following sufficient conditions for concavity on the *counting numbers* c_s and c_F:

$$\exists c_{FF}, c_{ss}, c_{sF} \geq 0 \;\middle|\; \begin{array}{ll} c_F = c_{FF} + \sum_{s \in F} c_{sF}, & \forall F \in \mathcal{F} \\ c_s = c_{ss} - \sum_{F : s \in F} c_{sF}, & \forall s \in V. \end{array} \tag{124}$$

Empirically, it has been found that such concave counting numbers often yield approximations that are inferior to the Bethe approximation in terms of the error of the resulting pseudo-marginals and the normalization constant. Hence, Meshi et al.[82] explore counting numbers that are as close as possible to the Bethe approximation, while still satisfying the above sufficient conditions for concavity.

[78] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*, chapter 8, pages 239–270. Morgan Kaufmann, 2002

[79] For graphs exceeding pairwise connectivity, the definition can easily be extended by summing over factor entropies, rather than edge entropies.

[80] Yair Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000

[81] Tom Heskes. Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies. *Journal of Artificial Intelligence Research*, 26:153–190, 2006

[82] Ofer Meshi, Ariel Jaimovich, Amir Globerson, and Nir Friedman. Convexifying the Bethe Free Energy. In *Uncertainty in Artificial Intelligence (UAI)*, 2009

Concave upper bounds. Concavity solves part of the problem of the Bethe approximation, but still, a concave approximation does not necessarily provide us with a relaxation. The second ingredient we require is for the approximation to form an upper bound on the true log-partition function. This property does not necessarily follow from concavity. Indeed, Weiss et al.[83] argue that the family of region-based approximations providing a bound on $A(\theta)$ is a strict subset of the family of concave approximations.

[83] Yair Weiss, Chen Yanover, and Talya Meltzer. MAP Estimation, Linear Programming and Belief Propagation with Convex Free Energies. In *Uncertainty in Artificial Intelligence (UAI)*, 2007

An important member of the former is the family of *tree-reweighted* approximations.[84] Such approximations are of the form

$$H_{\text{TRW}}(\boldsymbol{\tau}) = \sum_{(s,t)\in E} \nu_{st} H(\boldsymbol{\tau}_{st}) - \sum_{s\in V} \big[\textstyle\sum_{(s,t)} \nu_{st} - 1\big] H(\boldsymbol{\tau}_s), \tag{125}$$

[84] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

where the *edge occurrence* probabilities $\boldsymbol{\nu}_{st}$ denote the probability of an edge (s,t) belonging to a spanning tree of the graph, according to a distribution over all spanning trees. Since tree-reweighted approximations provide a bound on the true entropy, one obtains a valid *relaxation*

$$A_{\text{TRW}}(\theta) = \max_{\boldsymbol{\tau}\in\mathbb{L}(G)} \big\{\langle \theta, \boldsymbol{\tau}\rangle + H_{\text{TRW}}(\boldsymbol{\tau})\big\}. \tag{126}$$

The second relaxation we are going to consider subsequently,

$$A_{\text{TRIV}}(\theta) = \max_{\boldsymbol{\tau}\in\mathbb{L}(G)} \big\{\langle \theta, \boldsymbol{\tau}\rangle + \sum_{F\in\mathcal{F}} H(\boldsymbol{\tau}_F)\big\}, \tag{127}$$

is based purely on factor entropies. As we are going to point out in the next chapter, as long as each variable is covered by a factor, this is again an upper bound on the true log-partition function.

Application to marginalization. Either of the two relaxations we described above is suitable for our purposes. By virtue of convexity, it is in principle possible to obtain the exact solution. Moreover, by virtue of being relaxations, we obtain an upper bound on the log-partition function.

If applicable, the tree-reweighted approximation must be expected to yield more accurate pseudo-marginals, since it is *variable valid*, that is, for a distribution that factors completely over the variables, the estimated entropy is going to be correct. Variable validity has been identified by Meshi et al.[85] as a crucial factor as far as approximation accuracy is concerned. Nonetheless, for discriminative training, the factor-based relaxation is still useful. In this context, accuracy of the pseudo-marginals is not of surmount importance—from an empirical risk minimization perspective, the factor-based approximation will simply induce a slightly different logistic loss function that still seeks to align the mode of the conditional posterior distribution with the data.

[85] Ofer Meshi, Ariel Jaimovich, Amir Globerson, and Nir Friedman. Convexifying the Bethe Free Energy. In *Uncertainty in Artificial Intelligence (UAI)*, 2009

An important question is how the relaxed optimization problems are solved in practice. Given their close similarity to the Bethe approximation, which, as we intimated, is sought to be optimized by loopy belief propagation, it seems likely that similar message passing algorithms should exist. Indeed, Wainwright et al.[86] introduced precisely such a message passing algorithm along with their approximation. However, despite convexity, this algorithm is not guaranteed to converge. In the next chapter, we will hence introduce a novel algorithm that is guaranteed to find the exact optimum.

[86] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

Related Relaxations

MAP estimation. Relaxations for MAP estimation are a well-studied field. Besides the linear programming relaxation we described in this chapter, quadratic programming relaxations[87] and second-order cone programming relaxations[88], as well as various variants thereof, have been considered.

However, as shown by Kumar et al.[89], the first-order linear programming relaxation *dominates* both the quadratic programming and the second-order cone programming relaxation mentioned above, i.e. , for any vector of exponential parameters θ, it establishes a tighter bound on $\mathring{A}(\theta)$.

This is a strong argument in support of linear programming relaxations over the local polytope. Moreover, as Sontag et al.[90] show, the local polytope can even be tightened by adding further marginalization constraints between larger clusters of variables. Such constraints can be added iteratively. To find out exactly which cluster results in the largest gain is intractable, Sontag et al. use a particular heuristic that results in guaranteed improvement in the convex dual of the objective; a different heuristic based on local duality gaps has been suggested by Batra et al.[91]

A problem of these iterative tightening approaches is that the cost of passing messages grows exponentially in the size of the clusters. This has been addressed by Sontag et al.[92] in follow-up work by partitioning the state space of a cluster and enforcing consistency only across partitions.

It is not clear how to make use of iterative tightening approaches in discriminative training, because it possibly results in a different relaxation at each evaluation of the learning objective, effectively changing the objective function we wish to optimize. One would need to fix the additional clusters in advance, which is either too expensive (if a large number of clusters are added), or possibly does not result in any improvement (because initially, it is unclear, which clusters to add). Hence, we will only consider the first-order LP relaxation in the sequence, which is already a veritable computational challenge in the context of discriminative training.

Marginalization. As far as the log-partition function is concerned, the literature is somewhat sparser. Wainwright and Jordan[93] propose a relaxation that combines semidefinite constraints and the constraints of the local polytope to form an outer approximation of the marginal polytope, and draws on a log-determinant bound on the true entropy. However, computationally, this relaxation is somewhat less convenient than the ones we introduced in this chapter, and it received less attention. Nonetheless, it would be interesting to explore its utility for discriminative training.

[87] Pradeep Ravikumar and John Lafferty. Quadratic programming relaxations for metric labelling and Markov random field MAP estimation. In *International Conference on Machine Learning (ICML)*, 2006

[88] Pawan M. Kumar, Philip H. S. Torr, and Andrew Zisserman. Solving Markov Random Fields using Second Order Cone Programming Relaxations. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1045–1052, 2006

[89] Pawan M. Kumar, Vladmir Kolmogorov, and Philip H. S. Torr. An Analysis of Convex Relaxations for MAP Estimation. In *Advances in Neural Information Processing Systems (NIPS)*, 2007

[90] David Sontag, Talya Meltzer, Amir Globerson, Yair Weiss, and Tommi Jakkola. Tightening LP Relaxations for MAP using Message Passing. In *Uncertainty in Artificial Intelligence (UAI)*, 2008

[91] Dhruv Batra, Sebastian Nowozin, and Pushmeet Kohli. A Local Primal-Dual Gap based Separation Algorithm. In *Artificial Intelligence and Statistics (AISTATS)*, 2011

[92] David Sontag, Amir Globerson, and Tommi Jakkola. Clusters and Coarse Partitions in LP Relaxations. In *Neural Information Processing Systems (NIPS)*, 2008

[93] Martin J. Wainwright and Michael I. Jordan. Log-Determinant Relaxation for Approximate Inference in Discrete Graphical Models. *IEEE Transactions on Signal Processing*, 54(6):2099–2109, 2006

Novel Convergent Inference Algorithms

Overview

We have seen that for discriminative training, it is of particular importance to be able to compute mean parameters (corresponding to marginal probabilities in discrete models), as well as modes (corresponding to maximum a-posteriori states), efficiently. We start by introducing two *inference* algorithms towards this end. Both algorithms optimize a convex objective function, thereby yielding pseudo-marginals and pseudo-states that can be used in a principled manner within a convex learning objective.

In this chapter, we will already see examples of the richness of equivalent convex formulations that characterize optimization over the local polytope. In the chapter thereafter, we will exploit these duality relations to derive various equivalent objective functions for *learning*, each characterized by different strengths and weaknesses in terms of memory requirements and computational demands.

Convergent Solvers for the Tree-Reweighted Relaxation

In this section, we investigate minimization of the tree-reweighted relaxation we briefly introduced in the previous chapter, for the purpose of obtaining approximate marginal probabilities and upper bounds on the partition function of cyclic graphical models. The solvers we present for this problem work by directly tightening tree-reweighted upper bounds. As a result, they are particularly efficient for tree-reweighted relaxations arising from a small number of spanning trees. While this assumption may seem restrictive at first, we show how small sets of trees can be constructed in a principled manner. An appealing property of our algorithms, which results from the problem decomposition, is that they are embarrassingly parallel. In contrast to the original message passing algorithm introduced for this problem, we obtain global convergence guarantees.

Motivation

As we have seen, exact computation of marginal probabilities and the partition function in general graphical models is an NP-hard problem that scales exponentially in the tree width of the graph.[94] Much effort has been put into construction of approximate inference algorithms that remain tractable even for graphs of large tree width, such as those involving many cycles. Good results were initially obtained using loopy belief propagation, which

[94] Venkat Chandrasekaran, Nathan Srebro, and Prahladh Harsha. Complexity of Inference in Graphical Models. Technical report, 2010

ignores the cycles and performs message updates as if the graph were a tree.[95] Theoretical justification was later given to the method by showing that it can be understood to minimize the so-called Bethe free energy.[96] However, the Bethe free energy is convex only for tree-structured graphs and other special cases, such as graphs involving a single loop. Hence, loopy belief propagation cannot in general be expected to establish the global minimum, nor is it guaranteed to converge.

In the seminal work of Wainwright et al.,[97] a tree-reweighted (TRW) relaxation of the log-partition function was introduced to rectify this problem. This relaxation arises from a convex combination of log-partition functions of spanning trees that forms a natural upper bound on the exact log-partition function. The tree-reweighted free energy itself then only depends on edge occurrence probabilities resulting from the choice of trees. An adapted message passing algorithm was derived that is reminiscent of loopy belief propagation, but minimizes a tree-reweighted free energy instead of the Bethe free energy. However, despite convexity of its objective function, the algorithm is not guaranteed to converge. Indeed, we will demonstrate this failure of convergence.

Consequently, recent work has focused on establishing convergent variants of the original algorithm. Previous attempts have aimed at optimization of the tree-reweighted free energy itself, rather than direct minimization of the convex upper bound. In part, this is due to the original presentation by Wainwright et al., who form the convex combination over *all* spanning trees of the cyclic graph. Naturally, direct minimization of this bound is infeasible. However, if the upper bound is restricted to a small number of spanning trees, this optimization problem has favorable properties. Moreover, approximate marginal probabilities result naturally as a byproduct.

In fact, for the related maximum-a-posteriori (MAP) problem, Komodakis et al.[98] have shown that a similar convex upper bound, formed over a small number of trees, can be minimized efficiently using the projected subgradient algorithm. Optimization of the upper bound on the log-partition function differs in two key ways. First, the problem is smooth, which suggests improved asymptotic properties. Second, the choice of spanning trees can have significant influence on the tightness of the optimum.

Contributions. In this section, we make the following contributions:

a) We investigate direct minimization of tree-reweighted upper bounds on the log-partition function using the spectral projected gradient algorithm[99] and the projected quasi-Newton algorithm[100]. The core of the resulting algorithms is embarrassingly parallel and we demonstrate that it scales accordingly in the number of processors.

b) We present strategies for choosing small sets of spanning trees and study their effect on the error of marginal probabilities, tightness of the upper bound and computational cost. These results are of general interest as the choice of trees (or edge probabilities) is mandated by any tree-reweighted algorithm.

[95] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In *Uncertainty in Artificial Intelligence (UAI)*, 1999

[96] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*, chapter 8, pages 239–270. Morgan Kaufmann, 2002

[97] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

[98] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF Optimization via Dual Decomposition: Message-Passing Revisited. In *International Conference on Computer Vision (ICCV)*, 2007

[99] Ernesto G. Birgin, José M. Martínez, and Marcos Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000

[100] Mark Schmidt, Ewout Van den Berg, Michael P. Friedlander, and Kevin Murphy. Optimizing Costly Functions with Simple Constraints: A Limited-Memory Projected Quasi-Newton Algorithm. In *Artificial Intelligence and Statistics (AISTATS)*, 2009

To our knowledge, our approach was the first to consider direct minimization of the upper bounds on the log partition function, as opposed to optimization of the corresponding tree-reweighted free energy. Since these problems are connected through strong duality, either formulation can be used to obtain the tightest bound and the corresponding pseudomarginals.

Recently, it came to our notice that a similar approach was followed independently by Domke[101] and published shortly after ours.

[101] Justin Domke. Dual Decomposition for Marginal Inference. In *Conference on Artificial Intelligence (AAAI)*, 2011

Background

Let us now make clear our notation and recapitulate a few concepts that will be useful in understanding our approach.

Model and notation. We consider undirected graphical models defined over discrete random variables with at most pairwise interactions. Remember that the probability of a joint variable state $\mathbf{y} \in \mathcal{Y}$ thus factors as

$$p(\mathbf{y};\theta) = \exp\Big(\sum_{s\in V} \theta_s(y_s) + \sum_{(s,t)\in E} \theta_{st}(y_s,y_t) - A(\theta)\Big), \tag{128}$$

or equivalently, expressed using the sufficient statistics corresponding to the vector of exponential parameters $\theta \in \mathbb{R}^d$,

$$p(\mathbf{y};\theta) = \exp(\langle \theta, \boldsymbol{\phi}(\mathbf{y})\rangle - A(\theta)). \tag{129}$$

Subsequently, we will be concerned with computation of approximations to $A(\theta)$ and the marginal probabilities

$$\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta)}[\phi_\alpha(\mathbf{y})] = \sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y};\theta)\phi_\alpha(\mathbf{y}), \quad \alpha \in \mathcal{I}, \tag{130}$$

where we use $\alpha \in \mathcal{I} = \{1, 2, \dots d\}$ to refer to a single index corresponding to a particular state of a vertex s or an edge (s,t). The first and second derivatives of $A(\theta)$ then generate the cumulants

$$\frac{\partial A(\boldsymbol{\theta})}{\partial \theta_\alpha} = \mathbb{E}[\phi_\alpha(\mathbf{y})] \quad \text{and} \quad \frac{\partial^2 A(\boldsymbol{\theta})}{\partial\theta_\alpha \partial\theta_\beta} = \operatorname{cov}[\phi_\alpha(\mathbf{y}), \phi_\beta(\mathbf{y})], \quad \alpha,\beta \in \mathcal{I}.$$

Hence, the marginal probabilities are given precisely by the gradient of the log-partition function. Moreover, the covariance matrix, which is by definition positive semi-definite, forms the Hessian. Convexity of the log-partition function follows from this property; this provides a different perspective on convexity, compared to the one based on the variational representation of $A(\theta)$ and Danskin's theorem we have already seen.

Tree-reweighted upper bounds. Consider now the set $\mathcal{T} = \{T\}$ of all spanning trees of a cyclic graph G. We use $\mathcal{I}(T) = \{\alpha\}$ to denote the set of indices corresponding to states y_s of vertices and (y_s, y_t) of edges that belong to a particular tree T.[102]

Each of the spanning trees is associated with a vector of exponential parameters $\theta(T) \in \mathbb{R}^d$ that is tractable by the structural assumption. Wainwright et al.[103] observe that a convex combination $\sum_T \rho(T) A(\theta(T))$ over

[102] Since the trees are spanning, they cover all vertices $s \in V$. Hence, all $y_s \in \mathcal{Y}_s$ are contained by definition.

[103] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

trees yields an upper bound on $A(\theta)$ if the full set of tractable parameters $\boldsymbol{\Theta} = \{\theta(T)\}_{T\in\mathcal{T}}$ lies in the convex set

$$\mathcal{C}(\theta) = \left\{ \boldsymbol{\Theta} \;\middle|\; \begin{array}{ll} \theta_\alpha(T) = 0, & \forall T, \alpha \notin \mathcal{I}(T) \\ \sum_T \rho(T)\theta(T) = \theta & \end{array} \right\}, \tag{131}$$

and $\boldsymbol{\rho} = \{\rho(T)\}$ is constrained to belong to the simplex of distributions over $\mathcal{T}$,

$$\Delta = \left\{ \boldsymbol{\rho} \;\middle|\; \textstyle\sum_T \rho(T) = 1, \quad \rho(T) \geq 0, \forall T \right\}. \tag{132}$$

Observe that $\boldsymbol{\rho}$ must also be valid in the sense that each edge is covered with non-zero probability, otherwise $\mathcal{C}(\theta)$ is empty. The upper bound property now follows directly from Jensen's inequality:

$$A(\theta) = A\big(\textstyle\sum_T \rho(T)\theta(T)\big) \leq \sum_T \rho(T) A(\boldsymbol{\theta}(T)). \tag{133}$$

The structural constraints $\theta_\alpha(T) = 0$ in $\mathcal{C}(\theta)$ are not required for the upper bound to hold, but we include them in our presentation to make explicit the fact that the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}(T)\}$ are tractable.

A natural question is then how to obtain the tightest upper bound possible within this framework. For a given distribution $\boldsymbol{\rho}$ over spanning trees, and target parameters θ, we can simply optimize over the set of tractable parameterizations $\boldsymbol{\Theta}$ to obtain

$$\min_{\boldsymbol{\Theta}\in\mathcal{C}(\theta)} \sum_{T\in\mathcal{T}} \rho(T) A(\theta(T)). \tag{134}$$

Since the upper bound is a convex combination of convex functions, and the constraint set is convex, this is a convex optimization problem.

Tree-reweighted free energies. By forming the Lagrangian of (134) and exploiting the conjugate duality relation between the log-partition function and the negative entropy of a distribution, one can obtain an equivalent dual problem:[104]

$$\max_{\boldsymbol{\tau}\in\mathbb{L}(G)} \left\{ \langle \boldsymbol{\tau}, \theta \rangle + \textstyle\sum_s H(\boldsymbol{\tau}_s) - \sum_{(s,t)} \nu_{st} I(\boldsymbol{\tau}_{st}) \right\}, \tag{135}$$

where $\boldsymbol{\tau}_s$ and $\boldsymbol{\tau}_{st}$ have interpretations as node and edge pseudo-marginals, $H(\cdot)$ and $I(\cdot)$ denote the Shannon entropy and the mutual information, respectively, and the constraint set

$$\mathbb{L}(G) = \left\{ \boldsymbol{\tau} \geq \mathbf{0} \;\middle|\; \begin{array}{ll} \sum_{y_s} \tau_s(y_s) = 1, & \forall s \in V \\ \sum_{y_t} \tau_{st}(y_s, y_t) = \tau_s(y_s), & \forall y_s \in \mathcal{Y}_s, (s,t) \in E \\ \sum_{y_s} \tau_{st}(y_s, y_t) = \tau_t(y_t), & \forall y_t \in \mathcal{Y}_t, (s,t) \in E \end{array} \right\}. \tag{136}$$

ensures proper local normalization and marginalization consistency. The edge probabilities $\boldsymbol{\nu} = \{\nu_{st}\}$ are strictly positive and arise from the valid distribution $\boldsymbol{\rho} \in \Delta$ over spanning trees.

Constraint set (136) is the *local polytope* we discussed in the previous chapter, and the objective function in (135) is the negative *tree-reweighted free energy*. As the dual of a convex function, it is concave in $\boldsymbol{\tau}$, and strong duality holds. The primary advantage of problem (135) over (134) is its reduced dimensionality: it is independent of the number of spanning trees involved. However, constraint set $\mathbb{L}(G)$ is more complicated than $\mathcal{C}(\theta)$.

[104] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

Approach

Message passing algorithms commonly seek to optimize the tree-reweighted free energy given in (135), and thus have to handle constraint set $\mathbb{L}(G)$. The original message passing algorithm by Wainwright et al.[105] can be understood to perform block coordinate updates in the Lagrangian of (135). However, without further precautions, the scheme is not guaranteed to converge. In practice, "damping" strategies are often applied to improve the convergence characteristics. In contrast, we investigate efficient methods for direct minimization of (134). The coupling constraints in $\mathcal{C}(\boldsymbol{\theta})$ are easier to handle than $\mathbb{L}(G)$, and convergent minimization schemes thus arise naturally. We next discuss several key aspects of our approach.

[105] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

Obtaining marginals. An approximation to the log-partition function is naturally given by the optimum of problem (134). In contrast, it is not so obvious how to obtain approximate marginals from the solution. The key observation here arises en route of deriving (135) from (134): By forming the Lagrangian of (134), and taking derivatives with respect to θ_α, one obtains the stationary conditions

$$\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\hat{\boldsymbol{\theta}}(T))}[\phi_\alpha(\mathbf{y})] \stackrel{!}{=} \tau_\alpha, \qquad \forall T \in \mathcal{T}, \alpha \in \mathcal{I}(T). \tag{137}$$

Consequently, at an optimal solution $\hat{\boldsymbol{\Theta}}$, all trees share a single set of marginals. To construct a full set of pseudo-marginals $\boldsymbol{\tau}$, for each index α, we can thus use the marginal probability of any tree T for which $\alpha \in \mathcal{I}(T)$ once (134) has been solved to optimality. Notably, as we pointed out in the previous chapter, the marginals of any tree can be obtained efficiently.

Computing the gradient. As we pointed out previously, the derivative of the log-partition function $A(\cdot)$ with respect to θ_α is given by the corresponding marginal probability, $\mathbb{E}[\phi_\alpha(\mathbf{x})]$. Given that (134) is a weighted sum of such partition functions, it is easy to see that the full gradient with respect to all tractable parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}(T)\}_{T\in\mathcal{T}}$ is thus given by the gradients of the weighted individual terms,

$$\boldsymbol{\nabla}_{\boldsymbol{\Theta}} = \left\{\rho(T)\mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\boldsymbol{\theta}(T))}[\boldsymbol{\phi}(\mathbf{y})]\right\}_{T\in\mathcal{T}}.$$

In principle, this gradient can be computed very efficiently; the only concern is the number of spanning trees involved. We will discuss this issue in great detail in the sequence.

Handling the constraints. We now turn to discussion of the constraint set $\mathcal{C}(\boldsymbol{\theta})$, defined in (143). Both the coupling constraints $\sum_T \rho(T)\boldsymbol{\theta}(T) = \boldsymbol{\theta}$ and the structural constraints $\theta_\alpha(T) = 0$ are linear, so $\mathcal{C}(\boldsymbol{\theta})$ is convex.

As we shall point out, projection onto this set can be realized very efficiently. Formally, we search the solution to the following optimization problem:

$$\mathcal{P}_{\boldsymbol{\theta}}(\boldsymbol{\Theta}') = \operatorname*{arg\,min}_{\boldsymbol{\Theta}\in\mathcal{C}(\boldsymbol{\theta})} \left\|\boldsymbol{\Theta} - \boldsymbol{\Theta}'\right\|_2^2. \tag{138}$$

For all T, if $\alpha \notin \mathcal{I}(T)$, the structural constraints prescribe $\theta_\alpha(T) = 0$. These components are hence fully specified. Otherwise, the coupling constraints

Figure 19: TIGHTENBOUND algorithm. Shown here is the variant drawing on the spectral projected gradient (SPG) method. A key feature is that through the decomposition into independent problems, the log-partition function can be evaluated for each tree in parallel.

input : set of trees $\mathcal{T}$ and valid distribution ρ, target parameters θ,
arbitrary initial $\Theta^{(1)}$, step size interval $[a_{\min}, a_{\max}]$, history length h
output: pseudo-marginals τ, upper bound $A_{\text{TRW}} \geq A(\theta)$

$\Theta^{(1)} \leftarrow \mathcal{P}_\theta(\Theta^{(1)})$
$A^{(1)}_{\text{TRW}} \leftarrow$ **parallelized** $\sum_T \rho(T) A(\theta^{(1)}(T))$
$a^{(1)} \leftarrow 1/\|\mathcal{P}_\theta(\Theta^{(1)} - \nabla^{(1)}_\Theta) - \Theta^{(1)}\|$
$k \leftarrow 1$
while $\|\mathcal{P}_\theta(\Theta^{(k)} - \nabla^{(k)}_\Theta) - \Theta^{(k)}\| < \varepsilon$ **do**
  $\mathbf{d}^{(k)} \leftarrow \mathcal{P}_\theta(\Theta^{(k)} - a^{(k)}\nabla^{(k)}_\Theta) - \Theta^{(k)}$
  repeat
    choose $\lambda \in (0,1)$ *; e.g. via interpolation*
    $\Theta^{(k+1)} \leftarrow \Theta^{(k)} + \lambda \mathbf{d}^{(k)}$
    $A^{(k+1)}_{\text{TRW}} \leftarrow$ **parallelized** $\sum_T \rho(T) A(\theta^{(k+1)}(T))$
  until $A^{(k+1)}_{\text{TRW}} < \max\{A^{(k)}_{\text{TRW}}, \ldots, A^{(k-h)}_{\text{TRW}}\} + \epsilon\lambda \nabla^{(k)}_\Theta \cdot \mathbf{d}^{(k)}$
  $\mathbf{s}^{(k)} \leftarrow \Theta^{(k+1)} - \Theta^{(k)}$
  $\mathbf{y}^{(k)} \leftarrow \nabla^{(k+1)}_\Theta - \nabla^{(k)}_\Theta$
  $a^{(k+1)} \leftarrow \min\{a_{\max}, \max\{a_{\min}, (\mathbf{s}^{(k)} \cdot \mathbf{s}^{(k)})/(\mathbf{s}^{(k)} \cdot \mathbf{y}^{(k)})\}\}$
  $k \leftarrow k+1$
return $(A^{(k)}_{\text{TRW}}, \tau = \text{marginals}\{\nabla^{(k)}_\Theta\})$ *; via relation betw. marginals and gradient*

$\sum_T \rho(T)\theta_\alpha(T) = \theta_\alpha$ must be satisfied. Among the admissible $\{\theta_\alpha(T)\}$ for a given index α, whose weighted sum must be θ_α, the sum of squares is minimized if $(\theta_\alpha(T) - \theta'_\alpha(T))^2$ is equal for all trees T with $\alpha \in \mathcal{I}(T)$. Consider now the distance from the target parameter $\delta_\alpha = (\sum_T \rho(T)\theta'_\alpha(T) - \theta_\alpha)$ and the accumulated probability mass $\sigma_\alpha = \sum_{T:\alpha\in\mathcal{I}(T)} \rho(T)$. It can be verified that the projection given by

$$\mathcal{P}_\theta(\Theta') = \begin{cases} \theta_\alpha(T) = 0 & \text{if } \alpha \notin \mathcal{I}(T) \\ \theta_\alpha(T) = \theta'_\alpha(T) - \frac{\delta_\alpha}{\sigma_\alpha} & \text{otherwise} \end{cases} \tag{139}$$

ensures satisfaction of all constraints while adhering to the optimality criterion discussed above. Hence, it provides a solution to (138) which can be computed in $O(|\mathcal{T}|d)$, i.e. in time linear in the dimensionality of Θ.

Tightening the Bound

For now, assume that $\rho(T) > 0$ for a small number of trees T only. The gradient of our objective in (134) can then be computed efficiently. Moreover, the constraint set $\mathcal{C}(\theta)$ is convex and can be projected onto at little cost. A principal method for optimization in such a setting is the projected gradient algorithm. However, this basic method can be improved on significantly.

Spectral projected gradient method. The main improvements of the spectral projected gradient (SPG) method[106] over classic projected gradient descent are a particular choice of the step size due to Barzilai and Borwein[107] and a non-monotone, yet convergent line search due to Grippo et al.[108] In the setting of unconstrained quadratics, the SPG algorithm has been observed to converge superlinearly towards the optimum.

[106] Ernesto G. Birgin, José M. Martínez, and Marcos Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000

[107] Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988

[108] Luigi Grippo, Francesco Lampariello, and Stefano Lucidi. A non monotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, 23:707–716, 1986

In Figure 19, we show the TIGHTENBOUND algorithm, which outlines the application of the spectral projected gradient method to optimization problem (134). Besides the mandatory input $\mathcal{T}$, $\boldsymbol{\rho}$ and $\boldsymbol{\theta}$, the meta parameters $[a_{\min}, a_{\max}]$ specify the interval of admissible step sizes, and history length h specifies how many steps may be taken without sufficient decrease of the objective. If the number of steps is exceeded, backtracking is performed and the step size is decremented until sufficient decrease has been established. In our implementation, we chose $a_{\min} = 10^{-10}$, $a_{\max} = 10^{10}$ and $h = 10$. In the backtracking step, we simply multiply by a factor of $\lambda = 0.3$. In practice, we found the TIGHTENBOUND algorithm to be very robust with respect to the choice of meta parameters.

Proposition 1 *For a given set of spanning trees $\mathcal{T}$, valid distribution over trees $\boldsymbol{\rho}$ and exponential parameters $\boldsymbol{\theta}$ of a discrete graphical model, the* TIGHTENBOUND *algorithm converges to the global optimum of* (134)*, or equivalently, the tightest tree-reweighted upper bound that can be achieved for the specific choice of trees.*

PROOF (SKETCH) Convergence is a simple consequence of the convergence of spectral project gradient methods, which was analyzed by Wang et al.[109]

[109] Changyu Wang, Qian Liu, and Xinmin Yang. Convergence properties of non monotone spectral projected gradient methods. *Journal of Computational and Applied Mathematics*, 182(1):51–66, 2005

Projected quasi-Newton method. The projected quasi-Newton (PQN) method was recently introduced by Schmidt et al.[110] and can be considered a generalization of L-BFGS[111] to constrained optimization. It is particularly suitable if the constraint set can be projected onto efficiently, and the objective is expensive to compute. At each iteration k, a feasible direction is found by minimizing a quadratic model subject to the original constraints:

$$\mathbf{d}^{(k)} = \operatorname*{arg\,min}_{\mathbf{d} \in \mathcal{C}(\boldsymbol{\theta})} \{ A^{(k)}_{\text{TRW}} + (\mathbf{d} - \boldsymbol{\Theta}^{(k)})^{\mathsf{T}} \boldsymbol{\nabla}^{(k)}_{\boldsymbol{\Theta}} + \tfrac{1}{2} (\mathbf{d} - \boldsymbol{\Theta}^{(k)})^{\mathsf{T}} \mathbf{B}^{(k)} (\mathbf{d} - \boldsymbol{\Theta}^{(k)}) \},$$

where $\mathbf{B}^{(k)}$ is a positive-definitive approximation to the Hessian that is maintained in compact form in terms of a fixed number of previous iterates and gradients.[112] The SPG algorithm can be used to perform this inner minimization effectively. We hypothesized that PQN might compensate for the larger per-iteration cost through improved asymptotic convergence and thus implemented a variant of the TIGHTENBOUND algorithm drawing on PQN, similar to the one shown in Figure 19. We do not give a complete specification here, as it only differs from Figure 19 in the choice of the direction and the use of a traditional line search.

[110] Mark Schmidt, Ewout Van den Berg, Michael P. Friedlander, and Kevin Murphy. Optimizing Costly Functions with Simple Constraints: A Limited-Memory Projected Quasi-Newton Algorithm. In *Artificial Intelligence and Statistics (AISTATS)*, 2009

[111] Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980

[112] Richard H. Byrd, Jorge Nocedal, and Robert B. Schnabel. Representations of quasi-Newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1):129–156, 1994

Choosing the Set of Trees

It is clear that the TIGHTENBOUND algorithm is only efficient for a reasonably small number of selected trees with $\rho(T) > 0$. We refer to this set as $\mathcal{S}$ and denote the corresponding vector of non-zero coefficients by $\boldsymbol{\rho}_s$. Subsequently, we discuss how to obtain $\mathcal{S}$ and $\boldsymbol{\rho}_s$ in a principled manner.

Uniform probabilities. According to the Laplacian principle of insufficient reasoning, one might choose uniform edge occurrence probabilities given by $\nu_{st} = (|V| - 1)/|E|$. However, in our formulation, we need to find a pair $(\mathcal{S}, \boldsymbol{\rho}_s)$ that *results* in these probabilities. The dual coupling between $(\mathcal{S}, \boldsymbol{\rho}_s)$

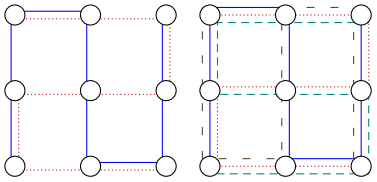
Figure 20: CoveringTrees algorithm. The algorithm determines sets of spanning trees resulting in uniform edge occurrence probabilities. If terminated early, it can be used to establish sets of small cardinality that still cover every edge of the original graph.

input : graph G, stopping criterion
output: selected trees $\mathcal{S}$, valid $\boldsymbol{\rho}_s$
$\mathcal{S}^{(1)} \leftarrow \{\text{random spanning tree}\}$, $\boldsymbol{\rho}_s^{(1)} \leftarrow [1]$, $k \leftarrow 1$
while not criterion **do**
  $\boldsymbol{\nu}^{(k)} \leftarrow \boldsymbol{\nu}(\mathcal{S}^{(k)}, \boldsymbol{\rho}_s^{(k)})$ *; compute edge probabilities*
  $\mathcal{S}^{(k+1)} \leftarrow \mathcal{S}^{(k)} \cup \mathrm{MST}(G, \boldsymbol{\nu}^{(k)})$ *; minimum spanning tree for edge cost* $\boldsymbol{\nu}^{(k)}$
  $\boldsymbol{\rho}_s^{(k+1)} \leftarrow \mathbf{1}/(k+1)$ *; for* $\mathbf{1} \in \mathbb{R}^{k+1}$
  $k \leftarrow k+1$
end
return $(\mathcal{S}^{(k)}, \boldsymbol{\rho}_s^{(k)})$

and $\boldsymbol{\nu}$ is defined in terms of the mapping $\boldsymbol{\nu}(\mathcal{S}, \boldsymbol{\rho}_s) = \sum_{T \in \mathcal{S}} \rho_s(T) \boldsymbol{\nu}(T)$, where $\boldsymbol{\nu}(T) \in \mathbb{R}^{|E|}$ indicates the edges contained in T, such that $\nu_{st}(T) = [\![(s,t) \in \mathcal{E}_T]\!]$. The CoveringTrees algorithm shown in Figure 20 establishes a suitable pair $(\mathcal{S}, \boldsymbol{\rho}_s)$ in a greedy manner. At each step, we add a minimum spanning tree (MST) for weights given by the current edge probabilities. We stop when $\boldsymbol{\nu}(\mathcal{S}, \boldsymbol{\rho}_s)$ is sufficiently uniform, which allows to trade off the number of resulting trees against uniformity.

Proposition 2 *Given any graph G, the* CoveringTrees *algorithm determines a sequence $\{\boldsymbol{\nu}(\mathcal{S}^{(k)}, \boldsymbol{\rho}_s^{(k)})\}$ converging to a vector $\mathbf{u}$ of uniform edge occurrence probabilities, all given by $u_{st} = (|V|-1)/|E|$, as $k \to \infty$.*

A proof will be given at the end of this section; the CoveringTrees algorithm can be seen to take conditional gradient steps that seek to minimize $\|\boldsymbol{\nu}(\mathcal{S}, \boldsymbol{\rho}_s) - \mathbf{u}\|_2^2$.



(a) (b)

Figure 21: (a) Two "snakes" cover any grid; (b) Two more mirrored replicas achieve symmetric edge probabilities.

Snake-based strategy. For grid-structured graphs, we also found that fairly uniform edge occurrence probabilities could be obtained using four "snake"-shaped trees that in sum cover all edges. This is best seen in terms of an illustration, which we provide in Figure 21. If we choose $\boldsymbol{\rho}_s = \mathbf{1}/|\mathcal{S}|$, the edges in the interior assume $\nu_{st} = 1/2$, whereas those on the boundary are given by $\nu_{st} = 3/4$.

Constructing an almost minimal set. If we choose a different stopping criterion, namely $\nu_{st} > 0 \ \forall (s,t)$, the CoveringTrees algorithm can also be used to greedily establish a set of trees that is almost minimal, in the sense that its cardinality is close to the minimum number of spanning trees required to cover all edges of G. Note that there is no guarantee of optimality in this respect. However, in practice, we found that the CoveringTrees algorithm was very effective at establishing sets of small cardinality.

Optimal sets of trees. Wainwright et al.[113] show that one can obtain even tighter upper bounds by optimizing (135) over the edge occurrence probabilities $\boldsymbol{\nu}$. This is achieved using conditional gradient steps, where each such outer iteration involves solution of (135) for the current iterate $\boldsymbol{\nu}^{(k)}$ and a subsequent minimum spanning tree (MST) search with edge weights given by the negative mutual information of the current edge pseudo-marginals, denoted by $I(\boldsymbol{\tau}_{st})$. The resulting bound is jointly optimal over $\boldsymbol{\nu}$

[113] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

input : graph G, target parameters θ
output: selected trees $\mathcal{S}$, valid ρ_s, τ, $A_{\text{TRW}} \geq A(\theta)$
$(\mathcal{S}^{(l)}, \rho_s^{(l)}) \leftarrow \text{CoveringTrees}(G, \nu_{st} > 0 \; \forall (s,t))$
$k \leftarrow l$
while not converged **do**
  $(A_{\text{TRW}}^{(k)}, \tau^{(k)}) \leftarrow \text{TightenBound}(\mathcal{S}^{(k)}, \rho_s^{(k)}, \theta)$
  $\mathbf{i}^{(k)} \leftarrow [-I(\mu_{st}^{(k)})]_{(s,t)\in E}$ *; negative mutual information per edge*
  $\mathcal{S}^{(k+1)} \leftarrow \mathcal{S}^{(k)} \cup \text{MST}(G, \mathbf{i}^{(k)})$ *; minimum spanning tree for edge cost* $\mathbf{i}^{(k)}$
  $\rho_s^{(k+1)} \leftarrow \mathbf{1}/(k+1)$ *; for* $\mathbf{1} \in \mathbb{R}^{k+1}$
  $k \leftarrow k+1$
end
return $(\mathcal{S}^{(k)}, \rho^{(k)}, \text{TightenBound}(\mathcal{S}^{(k)}, \rho_s^{(k)}, \theta))$

Figure 22: OptimalTrees algorithm. The algorithm converges to a set of spanning trees and tree probabilities yielding a jointly optimal (over trees and parameterizations) bound on the log-partition function.

and τ. Our OptimalTrees agorithm, shown in Figure 22, defines a similar procedure for the primal space we are operating in. It successively establishes pairs $(\mathcal{S}, \rho_s)$ resulting in increasingly tighter upper bounds A_{TRW}. The invocation of CoveringTrees$(\cdot)$ in the initialization phase ensures that we start from a valid distribution ρ_s and a small set $\mathcal{S}$ such that each edge is covered with non-zero probability and our TightenBound algorithm can be applied.

In practice, the biggest gains are achieved in the first few iterations. Hence, although it is expensive to find a suitable tree at each iteration, the number of trees stays relatively small, and we approach the joint optimum in the process. The fact that the set of trees stays small is crucial, since it allows us to employ our TightenBound algorithm.

Proposition 3 *The* OptimalTrees *algorithm determines a sequence* $\{A_{\text{TRW}}^{(k)}\}$ *converging to an upper bound* $A_{\text{TRW}} \geq A(\theta)$ *that is jointly optimal over the choice of spanning trees* $\mathcal{S}$*, the distribution over spanning trees* ρ_s*, and the tractable parameterization* Θ*, as* $k \to \infty$*.*

A proof will be given towards the end of the section; again, the key here is that the algorithm takes conditional gradient steps with respect to a particular objective function.

Parallelized Computation

The computational cost of the TightenBound algorithm is dominated by computation of $\sum_T \rho(T) A(\theta^{(k+1)}(T))$, which requires sum-product belief propagation on each tree $T \in \mathcal{S}$. One might then assume that compared to traditional message passing algorithms, this incurs an overhead that is asymptotically linear in the number of selected trees. However, observe that the terms $\{A(\theta^{(k+1)}(T))\}$ are completely independent of each other. Hence, as long as the number of CPU cores is greater than or equal to the number of trees, we can avoid the additional execution time by scheduling each run of belief propagation on a different core. As our experiments will demonstrate, this works very well in practice.

Generally, the SPG variant of our TightenBound algorithm is preferable from a parallelization point of view, since PQN incurs quite some overhead while solving the inner direction finding problem.

	GRID ISINGGAUSS		GRID ISINGUNIFORM		REGULAR ISINGGAUSS		COMPLETE EXPGAUSS	
	$e(A_{\text{TRW}})$	$e(\boldsymbol{\tau})$	$e(A_{\text{TRW}})$	$e(\boldsymbol{\tau})$	$e(A_{\text{TRW}})$	$e(\boldsymbol{\tau})$	$e(A_{\text{TRW}})$	$e(\boldsymbol{\tau})$
4SNAKES	0.085 ± 0.01	0.112 ± 0.01	0.104 ± 0.01	0.087 ± 0.00	$\sim$		$\sim$	
MINIMAL	0.088 ± 0.01	0.113 ± 0.01	0.109 ± 0.01	0.090 ± 0.00	0.833 ± 0.10	0.308 ± 0.05	0.397 ± 0.07	0.074 ± 0.01
UNIFORM	0.084 ± 0.01	0.110 ± 0.01	0.102 ± 0.01	0.085 ± 0.00	0.833 ± 0.10	0.308 ± 0.05	0.394 ± 0.07	0.074 ± 0.01
*UNIFORM	0.083 ± 0.01	0.110 ± 0.01	0.101 ± 0.01	0.085 ± 0.00	0.833 ± 0.10	0.308 ± 0.05	0.394 ± 0.07	0.074 ± 0.01
OPTIMAL	0.031 ± 0.01	0.091 ± 0.02	0.053 ± 0.01	0.079 ± 0.01	0.832 ± 0.10	0.308 ± 0.05	0.377 ± 0.07	0.075 ± 0.01

Table 1: Impact of the set of spanning trees on the approximation error

Experiments

Let us now assess several aspects of our algorithms empirically. Towards this end, we will consider four types of random graphs with varying structure and exponential parameters θ. All graph instances we will discuss in the sequence were generated using libDAI.[114]

[114] Joris M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, 2010

GRID ISINGGAUSS*:* an $n_g \times n_g$ grid of binary variables ($\mathcal{Y}_s = \{-1, +1\}$), with potentials chosen as $\theta_s(y_s) = \theta_s y_s$ and $\theta_{st}(\mathbf{y}_{st}) = \theta_{st} y_s y_t$, where θ_s and θ_{st} were drawn independently for each node and edge according to a $\mathcal{N}(0, 1)$ distribution.

GRID ISINGUNIFORM*:* Equal to the above, except that θ_s and θ_{st} were drawn from $\mathcal{U}(-1, +1)$.

REGULAR ISINGGAUSS*:* A random regular graph with n_r binary variables, each connected to n_d others, and potentials akin to GRID ISINGGAUSS.

COMPLETE EXPGAUSS*:* A complete graph with n_c variables ($\mathcal{Y}_s = \{0, 1, 2, 3\}$) and potentials independently drawn as $\theta_s(y_s) = 0$ and $\theta_{st}(\mathbf{y}_{st}) \sim \mathcal{N}(0, 1)$.

These graphs cover a broad spectrum, ranging from rather benign (grid) to almost pathological (complete).

	$e(A_{\text{TRW}})$
GRID ISINGGAUSS	0.096 ± 0.0018
GRID ISINGUNIFORM	0.112 ± 0.0019
REGULAR ISINGGAUSS	0.866 ± 0.0003
COMPLETE EXPGAUSS	0.355 ± 0.0023

Table 2: Standard deviation of the approximation error for 30 runs over the *same* graphs and potentials, using different MINIMAL sets of trees.

Impact of tree selection. For our experiments, we are going to consider four ways of decomposing the cyclic graphs into spanning trees:

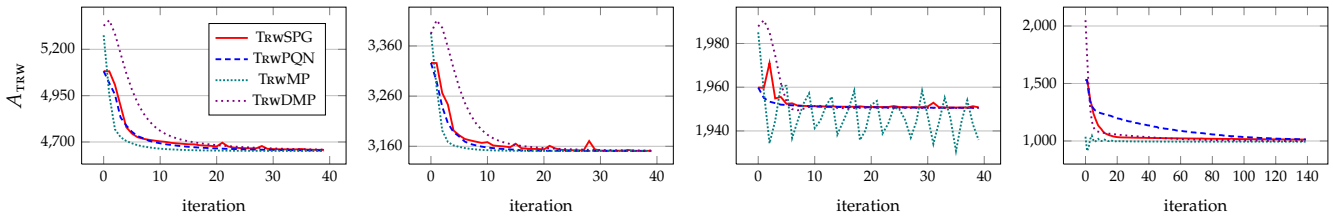
4SNAKES*:* Using a set of four "snakes", as illustrated in Figure 21; this decomposition is only applicable to grids.

MINIMAL*:* Using our greedy algorithm to establish an almost minimal set of covering trees.

UNIFORM*:* Drawing on the same algorithm, but allowing more iterations to achieve approximately uniform edge occurrence probabilities, stopping once $\min_{(s,t)} \nu_{st} \geq 0.9 \max_{(s,t)} \nu_{st}$.

OPTIMAL*:* Based on our algorithm for establishing optimal sets of trees.

First, let us assess the impact of the decomposition scheme on the approximation error of the bound on the log-partition function, which we compute as $e(A_{\text{TRW}}) = |A_{\text{TRW}} - A(\theta)|/A(\theta)$, and the error of the corresponding pseudo-marginals, given by $e(\boldsymbol{\tau}) = \|\boldsymbol{\tau} - \mathbb{E}[\boldsymbol{\phi}(\mathbf{y})]\|_1/d$.



(a) GRID ISINGGAUSS, MINIMAL (b) GRID ISINGUNIFORM, MINIMAL (c) REG. ISINGGAUSS, MINIMAL (d) COMP. EXPGAUSS, MINIMAL

Figure 23: Asymptotic efficiency—only a single run is depicted to highlight convergence characteristics.

For each type of graph, we generated 30 random instances (with $n_g = 15, n_r = 30, n_d = 10$ and $n_c = 10$) and solved the corresponding instance of (134) to a tolerance of $\varepsilon = 10^{-5}$ using our TIGHTENBOUND algorithm. The tree decomposition was computed anew for each instance. For the OPTIMAL scheme, we used 50 outer iterations; gains were minuscule beyond this point. The reference values $A(\theta)$ and $\mathbb{E}[\phi(\mathbf{y})]$ were computed using junction trees or brute force, depending on the graph.

Table 1 shows the average and the standard deviation ($\pm$) of the error over the 30 instances of each type of graph. As expected, the OPTIMAL scheme performs best almost universally, with large gains in some instances. More interestingly, the other three schemes are rather closely tied, with only a slight edge for the UNIFORM decomposition. For comparison, we also computed the approximation errors resulting from analytically determined uniform edge probabilities (*UNIFORM), which corresponds to an infinite number of iterations of our COVERINGTREES algorithm; the gains over the UNIFORM scheme are negligible.

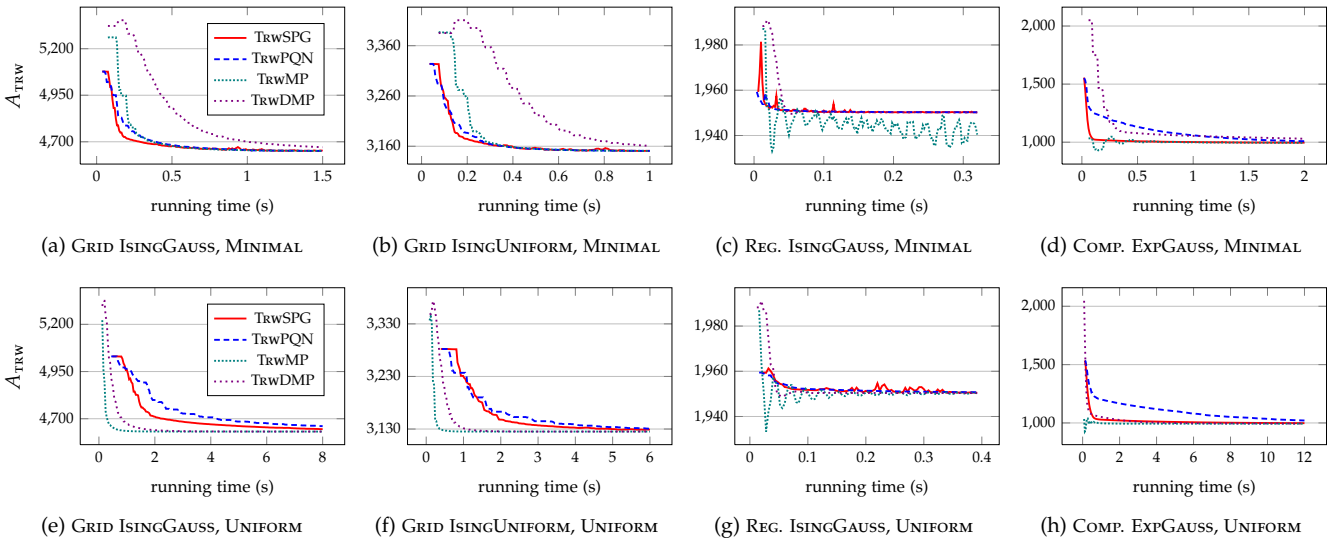
Let us now consider how deterministically the MINIMAL scheme behaves on a single given graph with fixed exponential parameters. Given the random nature of the decomposition, this is an important aspect. Table 2 confirms that the standard deviation of the approximation error over 30 independently computed decompositions is very low.

Effectiveness of solvers. Let us now compare our own solvers TRWSPG, outlined in Figure 19, and TRWPQN, its projected quasi-Newton variant (with $p = 4$), to the message passing algorithm (TRWMP) of Wainwright et al.[115] and a variant thereof (TRWDMP) that employs "damping" ($\alpha = 0.5$). In our implementation of the latter, we update the messages by iterating over the edges uniformly at random. For comparison, we use the same types of graphs as previously, but with $n_g = 50$, $n_r = 100$, $n_d = 10$ and $n_c = 50$.

[115] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

Asymptotic Efficiency. Let us first compare the asymptotic behavior of the competing solvers. To this end, we ran them on the same randomly generated instances of each type of graph. Figure 23 shows the progress of the objective as a function of iterations of the respective algorithm. The plot displays only a single run of each solver (rather than an average over multiple runs), so as not to "average out" the convergence characteristics. We only show the curves for a particular set of trees obtained using the MINIMAL scheme; the others triggered similar asymptotic behavior.

As one can see from Figure 23, the message updates performed by TRWMP decrease the objective very rapidly. However, this comes at a price.



(a) Grid IsingGauss, Minimal (b) Grid IsingUniform, Minimal (c) Reg. IsingGauss, Minimal (d) Comp. ExpGauss, Minimal

(e) Grid IsingGauss, Uniform (f) Grid IsingUniform, Uniform (g) Reg. IsingGauss, Uniform (h) Comp. ExpGauss, Uniform

Figure 24: Computational efficiency of competing solvers—average over 10 runs is shown here.

In some cases, e.g. panel (c), the process diverges. We also note that the iterates produced by TrwMP need not lie within $\mathbb{L}(G)$. Feasibility is only guaranteed for the optimal solution; hence, the curve can fluctuate about the optimum, see panel (d). This can even happen for TrwDMP, which generally improves smoothness of convergence considerably, but decreases the objective more slowly. In contrast, the iterates of TrwSPG and TrwPQN are always guaranteed to yield an upper bound. In terms of smoothness of convergence, TrwPQN exposes the most desirable behavior. On the other hand, TrwSPG implements a compromise between smoothness and rapid decrease; while its non-monotone line search can yield sporadic "bumps", it ultimately converges to the global optimum.

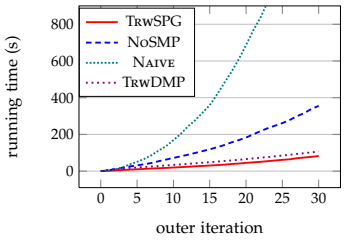
Computational Efficiency. Let us now assess the solvers in terms of their computational efficiency. For this purpose, we measure the progress of the objective as a function of running time, rather than iterations. The curve of each solver is averaged over 10 runs in order to smooth any effects caused by the random nature of the updates performed by TrwMP, or scheduling of the multiple CPU threads used by TrwSPG and TrwPQN. All results were obtained on a machine with 8 Intel Xeon CPU cores at 2.4 GHz.

Figure 24 shows the resulting plots for two decomposition schemes at opposing ends of the spectrum, Minimal (top row), and Uniform (bottom row). As expected, the results vary significantly with the number of trees in use. For Minimal sets, TrwSPG approaches the optimum even more quickly than TrwMP, and much more so than TrwDMP. TrwPQN is also competitive in some cases, but is generally dominated by TrwSPG due to the lower per-iteration cost. On the other hand, TrwMP and its damped variant are more efficient for the larger Uniform sets, since they only depend on the edge occurrence probabilities. This is particularly apparent in panels (e) and (f); over 50 spanning trees are required to achieve uniform edge probabilities, outnumbering the available CPU cores by far. However, previously, we saw that there is only limited gain in establishing Uniform

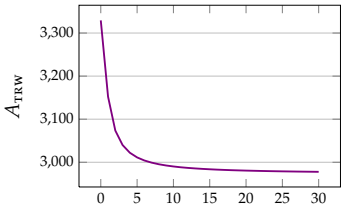
sets. Hence, one should definitely opt for a MINIMAL or 4SNAKES strategy with TRWSPG and TRWPQN. In this regime, TRWSPG outperforms both TRWMP and TRWDMP while guaranteeing convergence.

Scalability of optimal tree selection. Let us now consider OPTIMAL tree selection. Here, at each iteration, the set of trees grows. One might then expect the running time of the TIGHTENBOUND algorithm to increase at each iteration, such that the accumulated running time grows superlinearly. We draw on two strategies in order to suppress this effect. First, by parallelizing computation, the computation time of each iteration can be kept constant until the number of trees exceeds the number of cores. Second, by warm-starting the TIGHTENBOUND algorithm, almost-constant execution time can be maintained up to a relevant number of iterations: At each outer iteration, we start from the previous solution $\hat{\boldsymbol{\Theta}}$; the additional parameters $\theta(\hat{T})$ of the newly added MST are obtained from the weighted average over the other trees, $\boldsymbol{\theta}(\hat{T}) = \sum_{T' \in \mathcal{S} \setminus \hat{T}} \rho_s(T') \hat{\boldsymbol{\theta}}(T')$. All parameters are then projected to obtain an initial feasible point.

Figure 25 shows a run of the OPTIMALTREES algorithm. We compared our actual implementation (TRWSPG) to an implementation that does not use multi-processing (NOSMP) and a naive implementation that uses neither warm-starting nor multi-processing (NAIVE). As one can see, the differences are dramatic. Using the two strategies presented above, TRWSPG becomes an attractive choice as the inner solver, as it is guaranteed to converge. Finally, we assessed an implementation (TRWDMP) that uses damped ($\alpha = 0.5$) message passing to solve the inner problem, as in the original algorithm of Wainwright et al.[116] Figure 25 shows that up to a relevant number of iterations, this is less efficient than the TRWSPG-based scheme. Moreover, one does not know in advance which damping factor ensures convergence—a crucial aspect in this scenario.



(a) accumulated running time



(b) progress of objective

Figure 25: Constructing OPTIMAL sets of spanning trees for GRID ISINGUNIFORM: Note that TRWSPG scales linearly until after the number of trees exceeds the CPU cores.

[116] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

Related Work

As we noted in the introduction, it recently came to our notice that an approach very similar to ours was followed independently by Domke[117] and published shortly after ours.

Other than that, our formulation is most closely related to the dual decomposition scheme of Komodakis et al.,[118] who optimize an upper bound on the MAP score. As opposed to our setting, there is no strong duality between the (discrete) primal MAP problem and minimization of the convex upper bound, hence primal solutions must be generated heuristically. Moreover, the upper bound on the MAP score is non-differentiable, which has recently been dealt with using proximal regularization.[119] On the other hand, the upper bound on the log-partition function depends on the choice of trees, a different source of complication.

Several independent lines of work have focused on convergent algorithms for convex free energies. Heskes[120] derives convergent double-loop algorithms. He also argues that given sufficient damping, the original algorithm of Wainwright et al. should converge. Globerson and Jaakkola[121] provide a convergent algorithm for tree-reweighted free energies that solves an

[117] Justin Domke. Dual Decomposition for Marginal Inference. In *Conference on Artificial Intelligence (AAAI)*, 2011

[118] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF Optimization via Dual Decomposition: Message-Passing Revisited. In *International Conference on Computer Vision (ICCV)*, 2007

[119] Vladimir Jojic, Stephen Gould, and Daphne Koller. Accelerated dual decomposition for MAP inference. In *International Conference on Machine Learning (ICML)*, 2010

[120] Tom Heskes. Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies. *Journal of Artificial Intelligence Research*, 26:153–190, 2006

[121] Amir Globerson and Tommi S. Jaakkola. Convergent propagation algorithms via oriented trees. In *Uncertainty in Artificial Intelligence (UAI)*, 2007

unconstrained geometric program. However, the authors note their work is mostly of theoretical interest, since "damped" message passing converges more rapidly. Hazan and Shashua[122] devise a convergent algorithm for general convex energies by imposing strict non-negativity constraints on certain counting numbers of the entropy approximation. Meltzer et al.[123] provide a unifying view that relates convergence to the order in which message updates are performed.

Concerning parallelization, Gonzalez et al.[124] devise an efficient concurrent implementation of belief propagation. They show that synchronous schedules, which are naturally parallel, converge less rapidly—both empirically and theoretically. Hence, the authors parallelize a residual-based asynchronous schedule, which requires locking and considerable engineering effort. Moreover, their algorithm is not guaranteed to converge. On the other hand, some schemes that *do* guarantee convergence—such as that of Meltzer et al.—rely on the *order* of updates, which makes it inherently hard to gainfully employ parallelization. Our algorithms avoid these problems naturally.

[122] Tamir Hazan and Amnon Shashua. Convergent message-passing algorithms for inference over general graphs with convex free energies. In *Uncertainty in Artificial Intelligence (UAI)*, 2008

[123] Talya Meltzer, Amir Globerson, and Yair Weiss. Convergent message passing algorithms - A unifying view. In *Uncertainty in Artificial Intelligence (UAI)*, 2009

[124] Joseph E. Gonzalez, Yucheng Low, and Carlos Guestrin. Residual splash for optimally parallelizing belief propagation. In *Artificial Intelligence and Statistics (AISTATS)*, 2009

Conclusion

In this section, we derived convergent optimization schemes for computation of approximate marginal probabilities in cyclic graphical models. For tree-reweighted energies arising from a small number of spanning trees, our SPG-based solver was shown to be more efficient than the original message passing algorithm for this problem, while guaranteeing convergence. Moreover, we found empirically that such energies provide approximations of reasonable quality. If more accurate approximations are desired, one can additionally optimize over the choice of trees. Towards this end, we outlined an efficient algorithm that draws on our convergent solvers at each iteration to establish the joint global optimum. In this context, the convergence guarantees of our solvers are particularly valuable. We described how to avoid linear growth of the cost at each outer iteration, which improved computational efficiency by several orders of magnitude over a naive implementation.

Proofs Regarding Tree Selection

We start with a general discussion, as Proposition 2 and Proposition 3 are both based on the same framework. In particular, both algorithms seek the solution to a convex optimization problem

$$\min_{\boldsymbol{\nu} \in \mathbb{T}(G)} f(\boldsymbol{\nu}),$$

where $f(\cdot)$ is a convex function of the edge occurrence probabilities $\boldsymbol{\nu}$ and $\mathbb{T}(G)$ is the so-called *spanning tree polytope* of a graph G.[125] The latter is described by a number of inequalities that is exponential in the size of G. Nonetheless, one can optimize efficiently over $\boldsymbol{\nu}$ using the *conditional gradient* or Frank-Wolfe method.[126] Here, at each iteration k, we determine a feasible descent direction $\mathbf{p}^{(k)}$ through the solution of the first-order Taylor

[125] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971

[126] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999

expansion of $f(\cdot)$ around $\boldsymbol{\nu}^{(k)}$,

$$\min_{\boldsymbol{\nu}\in\mathbb{T}(G)} \left\{ f(\boldsymbol{\nu}^{(k)}) + \nabla f(\boldsymbol{\nu}^{(k)}) \cdot (\boldsymbol{\nu} - \boldsymbol{\nu}^{(k)}) \right\}.$$

We use $\hat{\boldsymbol{\nu}}^{(k)}$ to denote the minimizer of the above. The feasible descent direction is then given by $\mathbf{p}^{(k)} = \hat{\boldsymbol{\nu}}^{(k)} - \boldsymbol{\nu}^{(k)}$, and the next iterate is obtained as

$$\boldsymbol{\nu}^{(k+1)} = \boldsymbol{\nu}^{(k)} + \alpha^{(k)}\mathbf{p}^{(k)}, \quad \alpha^{(k)} \in [0,1].$$

Observe that this is equivalent to

$$\boldsymbol{\nu}^{(k+1)} = \alpha^{(k)}\hat{\boldsymbol{\nu}}^{(k)} + (1-\alpha^{(k)})\boldsymbol{\nu}^{(k)}, \quad \alpha^{(k)} \in [0,1],$$

i.e. , the new iterate is obtained as a convex combination of the previous iterate and the extreme point $\hat{\boldsymbol{\nu}}^{(k)}$, which can be found efficiently using the minimum spanning tree (MST) algorithm with edge weights given by $\nabla f(\boldsymbol{\nu}^{(k)})$. Hence, the MST algorithm solves the linear program $\min_{\boldsymbol{\nu}\in\mathbb{T}(G)} \langle \nabla f(\boldsymbol{\nu}^{(k)}), \boldsymbol{\nu} \rangle$ over the spanning tree polytope.

Lemma 1 *The steps taken by the* CoveringTrees *algorithm, as well as the* OptimalTrees *algorithm, are exactly of the form described above.*

Proof To see this, observe that at each step, the current edge occurrence probabilities $\boldsymbol{\nu}^{(k)}$ are maintained through the mapping

$$\boldsymbol{\nu}^{(k)} = \boldsymbol{\nu}(\mathcal{S}^{(k)}, \boldsymbol{\rho}_s^{(k)}) = \sum_{T\in\mathcal{S}^{(k)}} \rho_s^{(k)}(T)\boldsymbol{\nu}(T),$$

where $\boldsymbol{\nu}(T) \in \mathbb{R}^{|E|}$ indicates which edges are contained in T, that is, $\nu_{st}(T) = [\![(s,t) \in E(T)]\!]$. Each step chooses $\boldsymbol{\rho}_s^{(k+1)}$ as $\mathbf{1}/(k+1)$. Equivalently, we can develop the iterate as $\boldsymbol{\rho}_s^{(k+1)} = [\alpha^{(k)}, (1-\alpha^{(k)})\boldsymbol{\rho}_s^{(k)}]$ with $\alpha^{(k)} = 1/(k+1)$. Moreover, we note that the extreme point $\hat{\boldsymbol{\nu}}^{(k)}$ corresponds to a particular tree $\hat{T}^{(k)}$ via the relation $\hat{\boldsymbol{\nu}}^{(k)} = \boldsymbol{\nu}(\hat{T}^{(k)})$. It is precisely this tree $\hat{T}^{(k)}$ that both of the above-mentioned algorithms add to $\mathcal{S}^{(k)}$ with associated probability $\alpha^{(k)}$ at each step. But then, through the mapping between $(\mathcal{S}, \boldsymbol{\rho}_s)$ and $\boldsymbol{\nu}$, we obtain

$$\begin{aligned}
\boldsymbol{\nu}^{(k+1)} &= \boldsymbol{\nu}(\mathcal{S}^{(k+1)}, \boldsymbol{\rho}_s^{(k+1)}) \\
&= \alpha^{(k)}\boldsymbol{\nu}(\hat{T}^{(k)}) + \boldsymbol{\nu}(\mathcal{S}^{(k)}, (1-\alpha^{(k)})\boldsymbol{\rho}_s^{(k)}) \\
&= \alpha^{(k)}\hat{\boldsymbol{\nu}}^{(k)} + (1-\alpha^{(k)})\boldsymbol{\nu}^{(k)},
\end{aligned}$$

which is what we wanted to show.

To guarantee convergence of the framework, we also need the following lemma.

Lemma 2 *For the sequence of step sizes $\{\alpha^{(k)}\}$ chosen as $\alpha^{(k)} = 1/(k+1)$, the conditional gradient algorithm converges to the global minimum of $f(\cdot)$.*

Proof (Sketch) We do not give an explicit proof here. Global convergence of a conditional gradient algorithm with $\{\alpha^{(k)}\}$ chosen as a $1/(k+1)$ has been shown by Nedic and Subramanian,[127] among others.

Note that it is also possible to choose $\alpha^{(k)}$ such that sufficient decrease is obtained at each step by imposing the Armijo condition.[128] It remains to discuss the objective functions $f(\cdot)$ optimized by the CoveringTrees and OptimalTrees algorithms.

[127] Angelia Nedic and Vijay G. Subramanian. Approximately optimal utility maximization. In *IEEE Information Theory Workshop on Networking and Information Theory*, pages 206–210, 2009

[128] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999

Proof of Proposition 2. We wish to show that the CoveringTrees algorithm determines a sequence $\{\boldsymbol{\nu}(\mathcal{S}^{(k)}, \boldsymbol{\rho}_s^{(k)})\}$ that converges to a vector $\mathbf{u}$ with components given by $u_{st} = (|V| - 1)/|E|$. To see this, consider the optimization problem

$$\min_{\boldsymbol{\nu} \in \mathbb{T}(G)} f_{\text{uni}}(\boldsymbol{\nu}), \quad f_{\text{uni}}(\boldsymbol{\nu}) \stackrel{\text{def}}{=} \|\boldsymbol{\nu} - \mathbf{u}\|_2^2.$$

To apply the conditional gradient algorithm, we require the gradient of the objective, which we develop as $\nabla f_{\text{uni}}(\boldsymbol{\nu}) = 2(\boldsymbol{\nu} - \mathbf{u})$. At each iteration k, to determine the extreme point $\hat{\boldsymbol{\nu}}^{(k)}$, we thus solve a minimum spanning tree problem with edge weights given by $2(\boldsymbol{\nu}^{(k)} - \mathbf{u})$. The constant factor 2 does not affect the solution, nor does the constant vector $\mathbf{u}$, the components of which are all equal. Consequently, we can solve the MST problem with edge weights given by $\boldsymbol{\nu}^{(k)}$. This is exactly what the CoveringTrees algorithm does. Finally, we note that $\mathbf{u} \in \mathbb{T}(G)$ such that $\boldsymbol{\nu} = \mathbf{u}$ can be achieved. Proposition 2 then follows from Lemma 1 and Lemma 2.

Proof of Proposition 3. We wish to show that the OptimalTrees algorithm determines a sequence $\{A_{\text{TRW}}^{(k)}\}$ converging to a bound $A_{\text{TRW}} \geq A(\boldsymbol{\theta})$ that is jointly optimal over the choice of trees $\mathcal{S}$, the distribution over trees $\boldsymbol{\rho}_s$, and the tractable parameterization $\boldsymbol{\Theta}$. To see this, consider the problem

$$\min_{\boldsymbol{\nu} \in \mathbb{T}(G)} f_{\text{opt}}(\boldsymbol{\nu}), \quad f_{\text{opt}} \text{ given by (135).}$$

[129] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

It can be shown[129] that $f_{\text{opt}}(\cdot)$ is convex and differentiable in $\boldsymbol{\nu}$, and that its partial derivatives are given by $\frac{\partial f_{\text{opt}}}{\partial \nu_{st}} = -I(\hat{\boldsymbol{\tau}}_{st})$, where $I(\hat{\boldsymbol{\tau}}_{st})$ denotes the mutual information of (s, t) given the pseudo-marginals $\hat{\boldsymbol{\tau}}$ that maximize the objective in (135). Note that at each iteration k, the term $\hat{\boldsymbol{\tau}}^{(k)}$ depends on the solution of (135) given the current iterate $\boldsymbol{\nu}^{(k)}$. At each step, the conditional gradient algorithm first determines $\hat{\boldsymbol{\tau}}^{(k)}$, and then finds the minimum spanning tree with the weight of edge (s, t) given by $-I(\hat{\boldsymbol{\tau}}_{st}^{(k)})$. Wainwright et al. show that by minimizing $f_{\text{opt}}(\cdot)$ over $\boldsymbol{\nu}$, one obtains a bound $A_{\text{TRY}} \geq A(\boldsymbol{\theta})$ that is jointly optimal over $\boldsymbol{\nu}$ and $\boldsymbol{\tau}$. Now, from Lemma 1, we conclude that the OptimalTrees takes the same steps as the conditional gradient algorithm of Wainwright et al. The only difference lies in the fact that the edge occurrence probabilities $\boldsymbol{\nu}^{(k)}$ are implicitly maintained in terms of the mapping $\boldsymbol{\nu}(\mathcal{S}^{(k)}, \boldsymbol{\rho}_s^{(k)})$, and that the pseudo-marginals $\hat{\boldsymbol{\tau}}^{(k)}$ are (equivalently) computed using the TightenBound algorithm, which at each step determines the parameterization $\hat{\boldsymbol{\Theta}}$ that minimizes the upper bound for the current iterate $(\mathcal{S}^{(k)}, \boldsymbol{\rho}_s^{(k)})$. Furthermore, Lemma 2 guarantees convergence for our choice of step sizes. It then follows that the sequence of upper bounds $\{A_{\text{TRW}}^{(k)}\}$ converges to a jointly optimal upper bound A_{TRW}.

An Incremental Subgradient Algorithm for MAP Estimation

In this section, we investigate minimization of the linear programming relaxation over the local polytope we introduced in the previous chapter. Our goals in doing so are twice-fold: First of all, we want to be able to obtain approximate integral solutions of the maximum-a-posteriori (MAP) problem (for *predicting*), and second, we want to be able to solve the relaxation itself to obtain a well-motivated bound on $\mathring{A}(\theta)$, which will be useful for approximate discriminative training in the chapter to follow.

Towards this end, we present an incremental subgradient algorithm for approximate computation of maximum-a-posteriori (MAP) states in cyclic graphical models. Its most striking property is its immense simplicity: each iteration requires only the solution of a sequence of trivial optimization problems. The algorithm can be equally understood as a dual decomposition scheme or as minimization of a degenerate tree-reweighted upper bound and assumes a form that is reminiscent of message-passing. Despite (or due to) its conceptual simplicity, it is equipped with important theoretical guarantees and exposes strong empirical performance.

Motivation

In recent years, machine learning has given rise to a number of very-large-scale optimization problems. In this regime, conceptually simple algorithms can have an edge over their mathematically involved counterparts, in particular if accuracy is not at a premium. One reason for this phenomenon is that constant overhead matters a lot in the very-large-scale setting. The purpose of our investigation is to assess whether the principle of simplicity extends to maximum-a-posteriori (MAP) estimation in cyclic graphical models. As we saw previously, this is an NP-hard combinatorial optimization problem in general. There have been several attempts to solve the task approximately by optimizing a continuous relaxation, most prominently the so-called first-order linear programming (LP) relaxation, first considered by Schlesinger,[130] which we discussed previously.

Industrial-strength general purpose solvers can be quite ineffective given the sheer size of the resulting programs.[131] Message passing algorithms, on the other hand, can exploit the special problem structure but are still a topic of ongoing research. For instance, the original tree-reweighted message passing algorithm of Wainwright et al.[132] is not guaranteed to converge at all, while later improvements by Kolmogorov[133] and a related formulation by Globerson and Jaakkola[134] establish convergence, but not necessarily to the global optimum (except for binary variables). This undesirable behavior is due to the non-differentiability of the dual of the linear programming relaxation, which in turn results from non-strict convexity of the primal.

Consequently, recent work has focused on smoothing a dual formulation of the LP relaxation[135] and on obtaining strict convexity in the primal formulation.[136] These approaches provide global convergence and improved asymptotic convergence rates at the cost of greater complexity. However, it is not immediate that solving a relaxation extremely accurately should result in better solutions of the discrete problem.

[130] Michail I. Schlesinger. Syntactic analysis of two-dimensional visual signals in the presence of noise. *Cybernetics and Systems Analysis*, 12(4):612–628, 1976

[131] Chen Yanover, Talya Meltzer, and Yair Weiss. Linear Programming Relaxations and Belief Propagation - An Empirical Study. *Journal of Machine Learning Research*, 7:1887–1907, 2006

[132] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. MAP Estimation via Agreement on Trees: Message-Passing and Linear Programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005

[133] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568 – 1583, 2006

[134] Amir Globerson and Tommi Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *Advances in Neural Information Processing Systems*, 2007

[135] Jason K. Johnson, Dmitry Malioutov, and Alan S. Willsky. Lagrangian relaxation for MAP estimation in graphical models. In *Allerton Conference on Communication, Control and Computing*, 2007; and Vladimir Jojic, Stephen Gould, and Daphne Koller. Accelerated dual decomposition for MAP inference. In *International Conference on Machine Learning (ICML)*, 2010

[136] Pradeep Ravikumar, Alekh Agarwal, and Martin J. Wainwright. Message-passing for Graph-structured Linear Programs: Proximal Methods and Rounding Schemes. *Journal of Machine Learning Research*, 11:1043–1080, 2010

A different approach was taken by Komodakis et al.,[137] who solve a dual decomposition formulation using the projected subgradient algorithm. Global convergence is guaranteed, although at a sublinear rate. At each iteration, their scheme involves max-product belief propagation on spanning trees of the graph. We found that in practice, this comes at a considerable cost. Depending on the structure of a graph, a substantial number of spanning trees can be required in order to cover all edges. The number of dual parameters is exceedingly large in these cases. Moreover, since each iteration requires repeated belief propagation, a significant computational overhead can accumulate.

Interestingly, the dual formulation of Komodakis et al. is equivalent to minimization of the tree-reweighted upper bounds of Wainwright et al. Whereas the first authors directly minimize this bound, the latter go on to determine a Lagrangian reformulation and devise message passing algorithms in order to solve it. An important finding in this context is that the choice of spanning trees does not matter as long as all edges are covered with non-zero probability. Moreover, as Kolmogorov[138] later noted, the trees need not be spanning. We exploit this freedom in the choice of trees to define a lightweight iterative scheme that solves a dual formulation of the first-order LP relaxation. The resulting algorithm is easy to implement, efficient in practice, and guaranteed to converge to the global optimum of the relaxation.

Subsequent to publication of our algorithm, augmented Lagrangian approaches have become popular.[139] These approaches are interesting alternatives to our algorithm, since they are also guaranteed to converge; however, they seem to carry some (constant) overhead as compared to simpler block coordinate updates or our incremental subgradient scheme.

[137] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF Optimization via Dual Decomposition: Message-Passing Revisited. In *International Conference on Computer Vision (ICCV)*, 2007

[138] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568 – 1583, 2006

[139] André F. T. Martins, Mário A. T. Figueiredo, Pedro M. Q. Aguiar, Noah A. Smith, and Eric P. Xing. An Augmented Lagrangian Approach to Constrained MAP Inference. In *International Conference on Machine Learning (ICML)*, 2011; and Ofer Meshi and Amir Globerson. An Alternating Direction Method for Dual MAP LP Relaxation. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2011

Preliminaries

As in the previous section, we will consider undirected graphical models G with vertex set V and edge set E defined over discrete random variables with at most pairwise interactions. The potential of a joint variable state $\mathbf{y} \in \mathcal{Y}$ thus decomposes as

$$P(\mathbf{y}; \boldsymbol{\theta}) = \sum_{s \in V} \theta_s(y_s) + \sum_{(s,t) \in E} \theta_{st}(y_s, y_t), \tag{140}$$

where the exponential parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ (consisting of node potentials $\boldsymbol{\theta}_s$ and edge potentials $\boldsymbol{\theta}_{st}$) are considered given. In the following, we will be concerned with computation of approximations to the maximum-a-posteriori (MAP) value and state,

$$\mathring{A}(\boldsymbol{\theta}) = \max_{\mathbf{y} \in \mathcal{Y}} \Big\{ \textstyle\sum_s \theta_s(y_s) + \sum_{(s,t)} \theta_{st}(y_s, y_t) \Big\} \tag{141}$$

and

$$\hat{\mathbf{y}}(\boldsymbol{\theta}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \Big\{ \textstyle\sum_s \theta_s(y_s) + \sum_{(s,t)} \theta_{st}(y_s, y_t) \Big\}. \tag{142}$$

Tree-reweighted upper bounds

We next discuss the tree-reweighted upper bounds on $\mathring{A}(\theta)$ introduced by Wainwright et al.[140] and show that minimization of these bounds is equivalent to the dual decomposition formulation by Komodakis et al.[141]. Consider the set $\mathcal{T} = \{T\}$ of all spanning trees of a cyclic graph G. As in the bound on the log-partition function, each of the spanning trees is associated with a parameterization $\theta(T)$ that is tractable by the structural assumption; that is, the components of $\theta(T)$ corresponding to configurations y_s of vertices and (y_s, y_t) of edges that do not belong to a particular tree T are implicitly constrained to be zero. A convex combination $\sum_T \rho(T)\mathring{A}(\theta(T))$ over trees then yields a natural upper bound on $\mathring{A}(\theta)$ if the tractable parameters $\{\theta(T)\}_{T\in\mathcal{T}}$ and the distribution over trees $\boldsymbol{\rho}$ lie in the respective convex sets

$$\mathcal{C}(\theta) = \Big\{ \{\theta(T)\} \mid \textstyle\sum_T \rho(T)\theta(T) = \theta \Big\} \tag{143}$$

and

$$\Delta = \Big\{ \boldsymbol{\rho} \mid \textstyle\sum_T \rho(T) = 1, \quad \rho(T) \geq 0, \forall T \Big\}. \tag{144}$$

Note that the definition of $\mathcal{C}(\theta)$ implies that each edge (s,t) must be covered with non-zero probability unless the potentials θ_{st} are all zero. The upper bound now follows from Jensen's inequality:

$$\mathring{A}(\theta) \stackrel{\text{def}}{=} \mathring{A}\big(\textstyle\sum_T \rho^T \theta(T)\big) \leq \sum_T \rho(T)\mathring{A}\big(\theta(T)\big).$$

A natural question is then how to obtain the tighest upper bound possible within this framework. For a given distribution $\boldsymbol{\rho}$ over spanning trees, and given target parameters $\boldsymbol{\theta}$, we want to find the minimum over the set of tractable parameterizations $\{\theta(T)\}$,

$$\min_{\{\theta(T)\}\in\mathcal{C}(\boldsymbol{\theta})} \sum_{T\in\mathcal{T}} \rho(T)\mathring{A}\big(\theta(T)\big). \tag{145}$$

This is a convex optimization problem. For any feasible $\boldsymbol{\rho} \in \Delta$, the optimum attained in (145) will be the same, the reason being that the Lagrangian duals are all equivalent to the same LP relaxation.[142] Hence, we can choose $\boldsymbol{\rho}$ such that most coefficients $\rho(T)$ are zero, whereas the coefficients for a few selected trees needed to cover the edges, which we denote by $\mathcal{S}$, are equal to a common constant $\rho = 1/|\mathcal{S}|$. But then, the formulation in (145) reduces to

$$\min_{\{\boldsymbol{\lambda}(T)\}\in\mathcal{C}'(\theta)} \sum_{T\in\mathcal{S}} \mathring{A}\big(\boldsymbol{\lambda}(T)\big) \tag{146}$$

with

$$\mathcal{C}'(\boldsymbol{\theta}) = \Big\{ \{\boldsymbol{\lambda}(T)\} \;\Big|\; \textstyle\sum_{T\in\mathcal{S}} \boldsymbol{\lambda}(T) = \theta \;\Big\}, \tag{147}$$

where we moved the common constant ρ into the parameters by defining $\boldsymbol{\lambda}(T) = \rho\theta(T)$. Due to the linearity of $\mathring{A}(\cdot)$, this changes neither the solution nor the corresponding optimum. But now the equivalence to the formulation obtained by Komodakis et al. is apparent.

[140] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. MAP Estimation via Agreement on Trees: Message-Passing and Linear Programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005

[141] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF Optimization via Dual Decomposition: Message-Passing Revisited. In *International Conference on Computer Vision (ICCV)*, 2007

[142] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

Approach

How can the above problems be solved most efficiently? One might choose to exploit convex duality; indeed, one of the principal motivations of Wainwright et al.[143] in deriving the Lagrangian dual of (145) is the associated reduction in dimensionality; specifically, the number of parameters of the optimization problem is reduced from $|\mathcal{S}|d$ to d. However, as we shall point out, a similar dimensionality reduction is possible while maintaining the upper bound formulation in (145)–(146).

[143] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

Re-stating the Problem

We start out with the more convenient formulation given by (146). It is apparent that the objective does not depend on $\boldsymbol{\rho}$; moreover, as mentioned before, the optimum is independent of the choice of trees as long as each edge with non-zero potentials θ_{st} is covered. Our main idea is now to choose each tree as a single edge, such that $\mathcal{S}$ equals E. To make this choice explicit, we will use $F = (s,t) \in E$ to refer to such a degenerate tree consisting of a single edge from now on. An important consequence of our limitation to single-edge trees is that the edge potentials $\boldsymbol{\lambda}_{st}(F)$ of degenerate tree $F = (s,t)$ must equal the edge potentials θ_{st} of the target parameters $\boldsymbol{\theta}$, otherwise we have that $\{\boldsymbol{\lambda}(F)\} \notin \mathcal{C}'(\boldsymbol{\theta})$. Hence, the parameters $\boldsymbol{\lambda}_{st}(F)$ are fully specified. Moreover, those components of $\boldsymbol{\lambda}(F)$ corresponding to variables that are not part of F or edges other than F are implicitly constrained to be zero. Hence, the only remaining parameters of an edge $F = (s,t)$ are the node parameters $\boldsymbol{\lambda}_s(F)$ and $\boldsymbol{\lambda}_t(F)$. For notational convenience, we will refer to these parameters as $\boldsymbol{\lambda}(F) = \{\boldsymbol{\lambda}_s^F, \boldsymbol{\lambda}_t^F\}$ in the following.

The MAP value of each degenerate tree $F = (s,t)$ is easily obtained as

$$\mathring{A}^F\big(\boldsymbol{\lambda}(F);\boldsymbol{\theta}\big) = \max_{(y_s,y_t)\in\mathcal{Y}_s\times\mathcal{Y}_t} \Big\{\lambda_s^F(y_s) + \lambda_t^F(y_t) + \theta_{st}(y_s,y_t)\Big\}. \tag{148}$$

[144] Remember that the maximum need not be attained uniquely.

Subsequently, we use $(\hat{y}_s^F, \hat{y}_t^F)$ to denote any maximizing edge state[144] of the above, which can be found by maximizing over $|\mathcal{Y}_s \times \mathcal{Y}_t|$ sums of three scalar values. In contrast, computation of $\mathring{A}(\boldsymbol{\lambda}(T))$ in (146) requires max-product belief propagation, which is a fairly elaborate procedure. Next, observe that the constraint set simplifies to

$$\mathcal{Q}(\boldsymbol{\theta}) = \Big\{\{\boldsymbol{\lambda}(F)\} \;\Big|\; \textstyle\sum_{F:s\in F} \boldsymbol{\lambda}_s^F = \boldsymbol{\theta}_s, \quad \forall s \in V\Big\}. \tag{149}$$

By defining $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}^F\}_{F\in E}$ and putting things together, we obtain the final formulation:

$$\begin{aligned} \text{minimize} \quad & D(\boldsymbol{\Lambda};\boldsymbol{\theta}) \stackrel{\text{def}}{=} \textstyle\sum_{F\in E} \mathring{A}^F\big(\boldsymbol{\lambda}(F);\boldsymbol{\theta}\big) \\ \text{s.t.} \quad & \boldsymbol{\Lambda} \in \mathcal{Q}(\boldsymbol{\theta}). \end{aligned} \tag{150}$$

Importantly, this new problem is defined in terms of the parameters $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}(F)\} \in \mathbb{R}^{(|\mathcal{Y}_s|+|\mathcal{Y}_t|)|E|}$, the dimensionality of which is significantly lower than that of $\{\boldsymbol{\lambda}(T)\}$. In particular, since $\boldsymbol{\Lambda}$ only involves parameters corresponding to node potentials, the number of parameters is not on the order of $|\mathcal{Y}_s \times \mathcal{Y}_t|$. In terms of memory consumption, formulation (150) is thus on par with message passing algorithms. Indeed, as we shall see,

each $\boldsymbol{\lambda}_s^F$ can be understood to carry messages between a node s and its containing edge F. Furthermore, comparing (150) and (146), we observe that the new objective replaces $|\mathcal{S}|$ elaborate optimization problems by a large number $|E|$ of primitive optimization problems.

Minimization of the Upper Bound

We next inspect the mathematical properties of (150). While this optimization problem is convex, the objective function $D(\boldsymbol{\lambda};\boldsymbol{\theta})$ is non-differentiable. Nonetheless, one can obtain a subgradient $\mathbf{g} \in \mathbb{R}^{(|\mathcal{Y}_s|+|\mathcal{Y}_t|)|E|}$ with respect to $\boldsymbol{\Lambda}$. Its components are given by

$$g_s^F(x_s) = [\![y_s = \hat{y}_s^F]\!], \;\; g_t^F(y_t) = [\![y_t = \hat{y}_t^F]\!], \qquad \forall F = (s,t), y_s, y_t, \tag{151}$$

where $\hat{y}_s^F$ and $\hat{y}_t^F$ belong to an edge MAP state $(\hat{y}_s^F, \hat{y}_t^F)$ that maximizes (148) for edge E, and $[\![\cdot]\!]$ evaluates to 1 if the condition inside the brackets is true and 0 otherwise. This subgradient is trivially bounded since we have $\|\mathbf{g}_s^F\|_2^2 = 1$ and hence $\|\mathbf{g}\|_2^2 = 2|E|$.

Consider now the constraint set $\mathcal{Q}(\boldsymbol{\theta})$. It turns out that there is an efficient way of projecting an infeasible point $\boldsymbol{\Lambda}'$ onto this set. Formally, we search the solution to the following problem:

$$\mathcal{P}_{\boldsymbol{\theta}}(\boldsymbol{\Lambda}') = \operatorname*{arg\,min}_{\boldsymbol{\Lambda} \in \mathcal{Q}(\boldsymbol{\theta})} \|\boldsymbol{\Lambda} - \boldsymbol{\Lambda}'\|_2^2. \tag{152}$$

It is easily seen that among the admissible $\{\lambda_s^F(y_s)\}$ for a given variable s and state y_s, which must sum to $\theta_s(y_s)$, the sum of squares is minimized if $(\lambda_s^F(x_s) - \lambda_s'^F(y_s))^2$ is equal for all containing edges $E_s = \{F \in E \mid s \in F\}$. For these components, we need to subtract a common constant

$$\delta_s(y_s) = \left(\textstyle\sum_{F' \in E_s} \lambda_s'^{F'}(y_s) - \theta_s(y_s)\right) / |E_s| \tag{153}$$

to restore feasibility. Hence, the optimal projection in the sense of (152) is given by

$$\mathcal{P}_{\boldsymbol{\theta}}(\boldsymbol{\Lambda}') = \left\{\lambda_s^F(y_s) \leftarrow \lambda_s'^F(y_s) - \delta_s(y_s), \quad \forall s \in V, F \in E_s, y_s \in \mathcal{Y}_s\right\}. \tag{154}$$

Equivalently, after each modification of a component $\lambda_s^F(y_s)$ of a feasible point, feasibility can be restored by distributing the amount of change uniformly over all components $\{\lambda_s^{F'}(y_s) \mid F' \in E_s\}$.

Incremental subgradient algorithm. Equipped with efficient ways of computing the subgradient and projecting onto the feasible set, we could use the projected subgradient algorithm to solve (150), analogously to Komodakis et al.[145]. However, our problem differs from theirs in that the number of component functions can be expected to be significantly larger. Hence, the incremental subgradient method[146] is an attractive option. Here, at each inner iteration, only the subgradient of a single component function is subtracted, after which feasibility is restored using projection. The subgradient of the next component function is then computed using the adapted parameters. When the parameters of the component functions overlap or are coupled through the constraints, this can result in significantly faster initial convergence.

[145] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF Optimization via Dual Decomposition: Message-Passing Revisited. In *International Conference on Computer Vision (ICCV)*, 2007

[146] Angelia Nedic and Dimitri P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12:109–138, 2001

Figure 26: IncMP algorithm for MAP estimation

Input : Graph G, target parameters θ, initial feasible point Λ
Output: Feasible primal solution $\tilde{\mathbf{y}}$ that is an approximation to $\hat{\mathbf{y}}(\theta)$
choose initial feasible primal solution $\tilde{\mathbf{y}}$ arbitarily ;
repeat
  pick next step size α and shuffle the set of edges E;
  foreach $F = (s,t) \in E$ **do**
    find edge MAP state: $(\hat{y}_s^F, \hat{y}_t^F)$;
    subtract scaled subgradient: $\lambda_s^F(\hat{y}_s^F) \leftarrow \lambda_s^F(\hat{y}_s^F) - \alpha$;
    $\lambda_t^F(\hat{y}_t^F) \leftarrow \lambda_t^F(\bar{y}_t^F) - \alpha$;
    foreach $F' \in E_s$ **do** project: $\lambda_s^{F'}(\hat{y}_s^F) \leftarrow \lambda_s^{F'}(\hat{y}_s^F) + \alpha/|E_s|$;
    foreach $F' \in E_t$ **do** project: $\lambda_t^{F'}(\hat{y}_t^F) \leftarrow \lambda_t^{F'}(\hat{y}_t^F) + \alpha/|E_t|$;
  foreach $s \in V$ **do**
    construct candidate $\mathbf{c}$: $c_s \leftarrow$ choose at random from $\{\hat{y}_s^F \mid F \in E_s\}$;
  if $P(\mathbf{c};\theta) > P(\tilde{\mathbf{y}};\theta)$ **then**
    accept best primal solution so far: $\tilde{\mathbf{y}} \leftarrow \mathbf{c}$;
  if $D(\Lambda;\theta) = P(\tilde{\mathbf{y}};\theta)$ **then**
    optimal primal solution found: **return** $\tilde{\mathbf{y}}$;
until converged;
approximate primal solution found: **return** $\tilde{\mathbf{y}}$;

Our IncMP algorithm, shown in Figure 26, outlines the application of this method to optimization problem (150). Each component subgradient is very sparse in our case; only two indices are ever non-zero, namely those corresponding to the variable states $\hat{y}_s^F$ and $\hat{y}_t^F$ of the edge MAP state that maximizes (148) for edge F. Consequently, each inner update only affects a small number of parameters. The parameters of edges other than F are affected through the projection step, which only involves adjacent edges. Hence, the structure of a graph determines how quickly parameter updates propagate through the graph, which mirrors the situation in message passing algorithms.

Two choices impact the convergence behavior of the IncMP algorithm significantly. The first one is the order in which edges F are selected for the inner updates. We found that a random update order (implemented using a Fisher-Yates shuffle) consistently gives good results over a variety of graph structures. This is also supported by findings of Nedic and Bertsekas,[147] who show improved convergence rates for updates in random order.

[147] Angelia Nedic and Dimitri P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12:109–138, 2001

The second choice concerns the sequence of step sizes $\{\alpha^{(k)}\}$. Several sequences are known for which convergence to the global optimum is guaranteed; however, these can be rather slow in practice. We implemented the following practical variant: Initially, $\alpha^{(0)}$ is set to the sample standard deviation of the potentials in θ. At each outer iteration, if $D(\cdot)$ has decreased, we opt for a moderate decrease, say, $\alpha^{(k+1)} = 0.95\alpha^{(k)}$; otherwise, we decrease aggressively, e.g. $\alpha^{(k+1)} = 0.5\alpha^{(k)}$. Once $\alpha^{(k)}$ drops below a tiny number ε in iteration k, we adopt a static schedule and choose $\alpha^{(k')} = \varepsilon/(k'-k)$ in subsequent iterations $k' = k+1, k+2, \ldots$ of the algorithm.

Proposition 4 *With the sequence of step sizes $\{\alpha^{(k)}\}$ chosen as described above, the* IncMP *algorithm, shown in Figure 26 converges to the global optimum of optimization problem* (150) *as k approaches infinity.*

PROOF (SKETCH) Initially, the sequence of step sizes $\{\alpha^{(k)}\}$ decreases such that we reach ε in a finite number of iterations. Subsequently, $\{\alpha^{(k)}\}$ is chosen as a nonsummable diminishing sequence, guaranteeing global convergence for bounded subgradients by the results of Nedic and Bertsekas.[148]

[148] Angelia Nedic and Dimitri P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12:109–138, 2001

Obtaining Primal Solutions

In general, we do not have strong duality between (141) and (150). Hence, it is not always possible to extract the exact MAP state $\hat{\mathbf{y}}(\theta)$ from an optimal solution $\hat{\boldsymbol{\Lambda}}$ of (150). However, "good" feasible points can be expected to be obtained from the edge MAP states $(\hat{y}_s^F, \hat{y}_t^F)$. Specifically, at each iteration of the INCMP algorithm, for each node s, we choose the component c_s of a candidate primal solution uniformly at random from the set $\{\hat{y}_s^F \mid F \in E_s\}$. Those MAP states that appear in more edges adjacent to s thus have a higher chance of being picked. We keep track of the best primal solution $\tilde{\mathbf{y}}$ generated so far, and in some cases, a certificate of optimality can be obtained for $\tilde{\mathbf{y}}$ this way.

Proposition 5 *Assume that at a given outer iteration of the* INCMP *algorithm, we have* $P(\tilde{\mathbf{y}}; \theta) = D(\boldsymbol{\Lambda}; \theta)$. *It then follows that* $\boldsymbol{\Lambda}$ *minimizes* $D(\cdot)$ *and* $\tilde{\mathbf{y}}$ *maximizes* $P(\cdot)$. *This happens precisely if there is a set of edge MAP states* $\{(\hat{y}_s^F, \hat{y}_t^F)\}$ *that for each node* s *agrees on the current solution* $\tilde{y}_s$. *In that case,* $\tilde{\mathbf{y}} = \hat{\mathbf{y}}(\theta)$.

PROOF (SKETCH) By construction, $P(\tilde{\mathbf{y}}; \theta)$ gives a lower bound on $\mathring{A}(\theta)$, whereas $D(\boldsymbol{\Lambda}; \theta)$ gives an upper bound. For the bounds to coincide, $\tilde{\mathbf{y}}$ and $\boldsymbol{\Lambda}$ must both be optimal. Agreement of the edge MAP states at the joint optimum follows from Wainwright et al.,[149] Proposition 1.

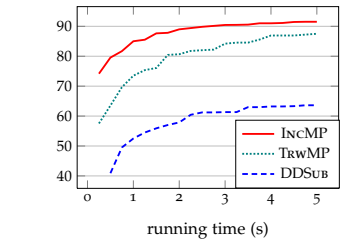
[149] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. MAP Estimation via Agreement on Trees: Message-Passing and Linear Programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005

Experiments

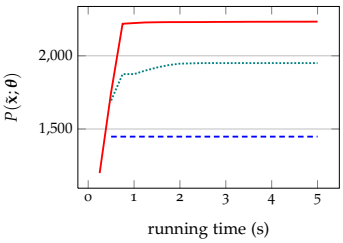
Let us consider three random graphs of varying structure and exponential parameters θ. First, GRIDISINGUNI is a 50×50 grid with binary variables ($\mathcal{Y}_s = \{-1, +1\}$) and potentials given by $\theta_s(y_s) = \gamma_s y_s$ and $\theta_{st}(y_s, y_t) = \gamma_{st} y_s y_t$ with γ_s and γ_{st} drawn from a $\mathcal{U}(-1, +1)$ distribution independently for each node and edge. Second, GRIDMULTIGAUSS is a 20×20 grid with variables of arity $|\mathcal{Y}_s| = 16$ and potentials chosen as $\theta_s(y_s) = 0$ and $\theta_{st}(y_s, y_t) \sim \mathcal{N}(0, 15)$ independently. Finally, COMPISINGUNI is a complete graph of 50 binary variables with potentials chosen akin to GRIDISINGUNI.

We compare INCMP, our own algorithm, to our implementations of two competing algorithms. We did not tune INCMP individually for each graph, but rather use the general step size schedule we previously described. By construction, a choice of trees is not required by INCMP. For DDSUB, the dual decomposition scheme of Komodakis et al.,[150] the greedy algorithm we described in the previous section was used to establish small sets of trees covering all edges and obtained the primal solutions similarly to INCMP. The step size schedule is similar to the one presented here and performed well in previous experiments. For TRWMP, the tree-reweighted message passing algorithm of Wainwright et al., the edge occurrence probabilities are obtained analogously to DDSUB, and the primal solutions are constructed from the maximizers of the node beliefs at each iteration. The messages are updated by iterating over factors in random order, akin to

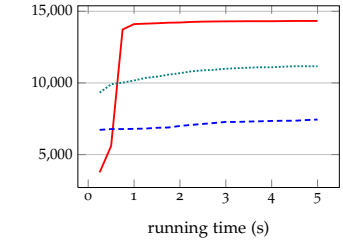
[150] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF Optimization via Dual Decomposition: Message-Passing Revisited. In *International Conference on Computer Vision (ICCV)*, 2007



(a) Comp. IsingUni



(b) Grid IsingUni



(c) Grid MultiGauss

Figure 27: Best primal solution found by the solvers as a function of time.

IncMP. For each graph and solver, we plot the best primal function value found as a function of running time, averaged over 20 runs. We exclude the time for generation of the set of trees needed by TrwMP and DDSub.

Figure 27 shows that our IncMP algorithm dominates its competitors on the three graphs discussed above. Interestingly, all algorithms were able to minimize the dual $D(\cdot)$ very effectively (not shown in the figure), but the quality of primal solutions found by the methods varies significantly. In particular, DDSub suffers from this phenomenon. One explanation is that the LP relaxation need not be very tight for graphs of substantial size and complexity. In this regime, the low per-iteration cost of IncMP allows for guided construction and evaluation of a large number of candidate solutions, which clearly pays off.

Conclusion and Outlook

We derived an efficient algorithm for approximate MAP estimation in cyclic graphical models that is reminiscent of message passing. It is characterized by the following properties: (a) guaranteed convergence to the global optimum of the first-order LP relaxation of the MAP problem; (b) by construction, we obtain both an upper bound and a lower bound on the exact MAP value; (c) if the LP relaxation is tight, the bounds coincide and we obtain the exact MAP state; (d) the memory requirements are equal to those of belief propagation. In future work, it may be interesting to employ the algorithm as the computational core in a branch-and-bound scheme. The above properties of the algorithm, along with the potential for warm-starting, render it an attractive choice in this setting.

Exploiting Duality in Discriminative Training

In the previous chapter, we have already seen examples of the various duality relations arising from relaxations of the log-partition function and the maximum a-posteriori function, and how these can be exploited for efficient inference in discrete graphical models. It is our restriction to *convex* relaxations that enables this flexibility. In this chapter, we will have an in-depth look at how these duality relations can be exploited for *training* of discrete graphical models. Interestingly, one can obtain a wide variety of equivalent optimization problems, each characterized by different strengths in terms of computational efficiency and memory requirements.

Conditional Random Fields and Max-Margin Markov Networks

In the sequence, we will be discussing discriminative parameter estimation in discrete graphical models, in particular maximum conditional likelihood estimation (a.k.a. conditional random fields)[151] and max-margin learning (a.k.a. max-margin Markov networks).[152]

[151] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *International Conference on Machine Learning (ICML)*, 2001

[152] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-Margin Markov Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2003

In the first part of the thesis, we already saw the objective functions these approaches seek to minimize. Since we restrict our attention to discrete graphical models in this part of thesis, we can re-consider these functions in terms of the marginal polytope. We obtain

$$\mathcal{O}_{\text{CRF}}(\mathbf{w}) = \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \big[E(\mathbf{y} \mid \mathbf{x};\mathbf{w}) + \underbrace{A(\boldsymbol{\theta}(\mathbf{x};\mathbf{w}))}_{\max_{\boldsymbol{\mu}\in\mathbb{M}^\circ(\mathbf{x})}\{\langle\boldsymbol{\theta}(\mathbf{x};\mathbf{w}),\boldsymbol{\mu}\rangle + H(p_{\boldsymbol{\theta}(\boldsymbol{\mu})})\}}\big] \tag{155}$$

and

$$\mathcal{O}_{\text{M3N}}(\mathbf{w}) = \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \big[E(\mathbf{y} \mid \mathbf{x};\mathbf{w}) + \underbrace{\mathring{A}(\boldsymbol{\theta}(\mathbf{x};\mathbf{w}) + \mathbf{e}(\mathbf{y}))}_{\max_{\boldsymbol{\mu}\in\mathbb{M}(\mathbf{x})}\{\langle\boldsymbol{\theta}(\mathbf{x};\mathbf{w}),\boldsymbol{\mu}\rangle + \langle\mathbf{e}(\mathbf{y}),\boldsymbol{\mu}\rangle\}}\big]. \tag{156}$$

Several comments are in order here: First of all, since for discrete random fields, the exponential parameters $\boldsymbol{\theta}(\mathbf{x};\mathbf{w}) \in \mathbb{R}^{d(\mathbf{x})}$ are *unconstrained*, we need not empose any constraints on the model parameters $\mathbf{w} \in \mathbb{R}^p$ to ensure feasibility of the exponential parameters.

Second, while the corresponding optimization problems are convex and unconstrained, the principal difficulty comes from the log-partition function A or the maximum a-posteriori function $\mathring{A}$, which require optimization over the marginal polytope.

Finally, the variational representation of the inner problem allows us to recognize the CRF objective as a particular special case of an M3N, where the loss term $\mathbf{e}(\mathbf{y})$ does not decompose over factors but is rather chosen as the entropy of the posterior distribution.

CRF Recipe 1 (Tightened Free Energy Formulation)
Choose a concave region-based entropy approximation $\tilde{H}$ and solve the convex optimization problem

$$\underset{\mathbf{w}}{\text{minimize}} \quad \tilde{\mathcal{O}}_{\text{CRF}}(\mathbf{w}) \stackrel{\text{def}}{=} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \left[E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) + \max_{\boldsymbol{\tau} \in \mathbb{L}(\mathbf{x})} \{ \langle \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}), \boldsymbol{\tau} \rangle + \tilde{H}(\boldsymbol{\tau}) \} \right]$$

by running convex message passing to solve the inner problem and obtain the maximizing pseudo-marginals $\boldsymbol{\tau}(\mathbf{x}; \mathbf{w})$ required to compute the gradient for each example, at each step of an iterative optimization algorithm.

The Relaxation Viewpoint

Let us now establish *tractable* approximations to the exact objective functions. Our starting point will be the notion of *relaxations* we developed over the previous chapters: Remember that a relaxation optimizes over a simpler set of feasible points (a superset of the original set), and that the relaxed objective function forms an upper bound on the original objective function (for all points in the original set).

The formulations we obtain from this perspective are particularly intuitive: We simply replace A or $\mathring{A}$ by a suitable relaxation (which maintains convexity), and solve for the model parameters, which is now tractable.

Training via Convex Entropy Approximations

As we discussed previously, for CRFs, a whole class of suitable relaxations is provided by convex free energies over the local polytope $\mathbb{L}(G)$. Specifically, the negative convex free energy, characterized by a particular region-based entropy approximation $\tilde{H}$, forms a relaxation of the log-partition function A. This approach is illustrated in CRF Recipe 1. For each training example, the original variational problem over the marginal polytope—computation of $A(\boldsymbol{\theta}(\mathbf{x}; \mathbf{w}))$—is simply replaced by a tractable optimization problem over the local polytope.

We already discussed two particular convex region-based entropy approximations that qualify for use in a relaxation,

$$H_{\text{TRW}}(\boldsymbol{\tau}) = \textstyle\sum_s H(\boldsymbol{\tau}_s) - \sum_{(s,t)} \nu_{st} I(\boldsymbol{\tau}_{st}), \tag{157}$$

and

$$H_{\text{TRIV}}(\boldsymbol{\tau}) = \textstyle\sum_F H(\boldsymbol{\tau}_F), \tag{158}$$

the *tree-reweighted* approximation of Wainwright et al.[153] and the *trivial* approximation of Weiss et al.[154], respectively. Either of those two approximations are suitable for our purposes.

Assuming that a choice regarding the entropy approximation has been made, the most important point is how the relaxed inference problem can be solved. This is of particular importance since the problem must be solved repeatedly, for each training example.

[153] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

[154] Yair Weiss, Chen Yanover, and Talya Meltzer. MAP Estimation, Linear Programming and Belief Propagation with Convex Free Energies. In *Uncertainty in Artificial Intelligence (UAI)*, 2007

In principle, tree-reweighted variational problems can be solved using the original message passing algorithm of Wainwright et al. However, problems can arise, since this algorithm is not guaranteed to converge. The authors suggest to "damp" the updates to avoid this effect. Since it is crucial to be able to find the exact optimum of the inner problem, it is advisable to use a convergent message passing scheme, such as the one introduced by Meltzer et al.[155] instead.

For the variational problem arising from the trivial entropy approximation, the *norm-product belief propagation* scheme has recently been introduced[156], which also guarantees convergence to the exact optimum.

In both cases, one obtains an approximation $\tilde{A}(\mathbf{x};\mathbf{w})$ to the log-partition function and a set of corresponding pseudo-marginals $\boldsymbol{\tau}(\mathbf{x};\mathbf{w})$. These are crucial in obtaining a gradient with respect to the model parameters,

$$\nabla\tilde{\mathcal{O}}_{\text{CRF}}(\mathbf{w}) = C\mathbf{w} + \sum_{(\mathbf{x},\mathbf{y})} [\mathbf{B}(\mathbf{x})]^{\mathsf{T}}[\boldsymbol{\tau}(\mathbf{x};\mathbf{w}) - \boldsymbol{\phi}(\mathbf{y})]. \tag{159}$$

This gradient can then be employed in a wide variety of iterative algorithms for unconstrained convex optimization. If exact solutions are desired, the L-BFGS[157] method has emerged as the algorithm of choice for such problems. For exceedingly large datasets, stochastic gradient methods have also emerged as viable choices.[158] Here, the gradient is only computed on a subset of training examples at each step, approximating the true gradient.

Either way, the main weakness of this approach is that at each step of an outer iterative optimization algorithm, several inner variational problems must be solved—one for each training example. Each such variational problem is a large-scale optimization problem, and a veritable challenge in its own right. Moreover, the inner optimization must be carried out to rather high precision, to ensure that the objective function remains convex (remember that maximization preserves the convexity), and that the line search of the outer optimization algorithm can make sufficient progress.

Use in previous work. Despite its drawbacks, this approach enjoys some popularity. For instance, it has successfully been followed by Levin and Weiss[159] in conditional random field learning for natural image segmentation. Moreover, Yanover et al.[160] have followed this route in optimizing the CRF parameters for side-chain prediction, a prominent problem from computational biology.

In both cases, the authors used a tree-reweighted entropy approximation. It is important to point out that a choice regarding the edge occurrence probabilities must be made in this case. As we saw in the previous chapter, the choice of edge occurrence probabilities can significantly affect the tightness of the approximation. The most popular choice seems to be to choose the edge occurrence probabilities as *uniform*. Our previous experiments confirm that this is a reasonable choice in principle. Nonetheless, it is somewhat unsettling to know that tighter approximations could exist within the same class of free energies.

An approach that suggests itself is then to consider the edge occurrence probabilities as part of the model parameters over which we optimize. Can such optimization be carried out efficiently? As we show next, this is indeed possible and a viable option.

[155] Talya Meltzer, Amir Globerson, and Yair Weiss. Convergent message passing algorithms - A unifying view. In *Uncertainty in Artificial Intelligence (UAI)*, 2009

[156] Tamir Hazan and Amnon Shashua. Norm-product belief propagation: Primal-dual message-passing for approximate inference. *IEEE Transactions on Information Theory*, 56(12):6294–6316, 2010

[157] Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980

[158] Léon Bottou and Olivier Bousquet. The Tradeoffs of Large Scale Learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2008

[159] Anat Levin and Yair Weiss. Learning to Combine Bottom-Up and Top-Down Segmentation. In *European Conference on Computer Vision (ECCV)*, 2006

[160] Chen Yanover, Ora Schueler-Furman, and Yair Weiss. Minimizing and Learning Energy Functions for Side-Chain Prediction. In *International Conference on Research in Computational Molecular Biology (RECOMB)*, 2007

CRF Recipe 2 (Tightened Tree-Reweighted Free Energy Formulation)
Solve the jointly convex optimization problem

$$\begin{aligned} \underset{\mathbf{w},\{\boldsymbol{\nu}\}}{\text{minimize}} \quad & \tilde{\mathcal{O}}_{\text{CRF}}(\mathbf{w},\{\boldsymbol{\nu}\}) \stackrel{\text{def}}{=} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \left[E(\mathbf{y}\,|\,\mathbf{x};\mathbf{w}) + \max_{\boldsymbol{\tau}\in\mathbb{L}(\mathbf{x})} \{\langle \boldsymbol{\theta}(\mathbf{x};\mathbf{w}),\boldsymbol{\tau}\rangle + \textstyle\sum_s H(\boldsymbol{\tau}_s) - \sum_{(s,t)} \nu_{st} I(\boldsymbol{\tau}_{st})\}\right] \\ \text{s.t.} \quad & \{\boldsymbol{\nu}\} \in \mathbb{T}^n \end{aligned}$$

by running convex message passing to solve the inner problem, obtaining the maximizing pseudo-marginals $\boldsymbol{\tau}(\mathbf{x};\mathbf{w})$ and the corresponding mutual information $I(\boldsymbol{\tau}_{st}(\mathbf{x};\mathbf{w}))$ required to compute the gradient for each example, at each step of an iterative optimization algorithm, ensuring feasibility of the edge occurrence probabilities $\boldsymbol{\nu}$ of each example by projecting onto the spanning tree polytope $\mathbb{T}(\mathbf{x})$.

The Special Case of Tree-Reweighted Approximations

In the previous chapter, when we introduced our TIGHTENBOUND algorithm, we already saw that it is possible to tighten the tree-reweighted upper bound over the edge occurrence probabilities in a double-loop algorithm. However, in conditional random field training, this approach would actually result in a triple-loop method, rendering optimization infeasible.

Our main observation is that the tree-reweighted approximation to the log-partition function,

$$A_{\text{TRW}}(\boldsymbol{\theta},\boldsymbol{\tau}) = \max_{\boldsymbol{\tau}\in\mathbb{L}(G)} \{\langle \boldsymbol{\theta},\boldsymbol{\tau}\rangle + \textstyle\sum_s H(\boldsymbol{\tau}_s) - \sum_{(s,t)} \nu_{st} I(\boldsymbol{\tau}_{st})\}, \tag{160}$$

is *jointly convex* both in $\boldsymbol{\theta}$ and the edge occurrence probabilities $\boldsymbol{\nu}$.[161] One can thus move the $\boldsymbol{\nu}$ parameters associated with each example into the overall objective function and use the gradients

$$\nabla_{\boldsymbol{\theta}} A_{\text{TRW}}(\boldsymbol{\theta},\boldsymbol{\tau}) = \hat{\boldsymbol{\tau}} \tag{161}$$

and

$$\nabla_{\boldsymbol{\tau}} A_{\text{TRW}}(\boldsymbol{\theta},\boldsymbol{\tau}) = -[I(\hat{\boldsymbol{\tau}}_{st})]_{(s,t)\in E} \tag{162}$$

to obtain the gradient with respect to the overall objective function.

The main complication arises from the fact that the edge occurrence probabilities of each example must belong to the spanning tree polytope,[162] which we denote by $\mathbb{T}(G)$.

Assuming we could project onto this polytope, the projected gradient methods[163] we already discussed would be suitable for handling this type of constraint. To do so, we need to be able to solve

$$\arg\min_{\boldsymbol{\tau}\in\mathbb{T}(G)} \|\boldsymbol{\tau} - \boldsymbol{\tau}'\|_2^2, \tag{163}$$

starting out from an infeasible point $\boldsymbol{\tau}'$. In fact, as we saw in the context of our COVERINGTREES algorithm, this can be done efficiently by solving a sequence of minimum spanning tree problems with edge weights given by $2(\boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}')$, effectively taking conditional gradient steps.

We are not aware of this approach having been followed in the literature before, but it certainly seems worthwhile.

[161] Observe that the objective function of the variational problem is linear in $\boldsymbol{\theta}$ and $\boldsymbol{\nu}$.

[162] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971

[163] Mark Schmidt, Ewout Van den Berg, Michael P. Friedlander, and Kevin Murphy. Optimizing Costly Functions with Simple Constraints: A Limited-Memory Projected Quasi-Newton Algorithm. In *Artificial Intelligence and Statistics (AISTATS)*, 2009; and Ernesto G. Birgin, José M. Martínez, and Marcos Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000

M3N Recipe 1 (Zero-Temperature Limit Formulation)
Choose a decomposing loss term $\mathbf{e}$, as well as an arbitrary concave region-based entropy approximation $\tilde{H}$ and a close-to-zero temperature $\mathfrak{T}$, and solve the convex optimization problem

$$\underset{\mathbf{w}}{\text{minimize}} \quad \tilde{\mathcal{O}}_{\text{M3N}}(\mathbf{w}) \stackrel{\text{def}}{=} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \left[E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) + \max_{\boldsymbol{\tau} \in \mathbb{L}(\mathbf{x})} \{ \langle \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) + \mathbf{e}(\mathbf{y}), \boldsymbol{\tau} \rangle + \mathfrak{T}\tilde{H}(\boldsymbol{\tau}) \} \right]$$

by running convex message passing to solve the inner problem and obtain the maximizing pseudo-marginals $\boldsymbol{\tau}(\mathbf{x}; \mathbf{w})$ required to compute the gradient for each example, at each step of an iterative optimization algorithm.

Free Energies in the Limit of the Temperature

Let us now consider training of M3Ns. Here, the maximum-a-posteriori function $\mathring{A}$ is the main source of complication. One promising approach, suggested by the close similarity to the log-partition function A, is to proceed as previously for CRFs, but in the "zero temperature" limit.

In particular, we already saw that

$$\overset{\mathfrak{T}}{A}(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \mathfrak{T}\, A(\boldsymbol{\theta}/\mathfrak{T}) \tag{164}$$

forms a smooth approximation to $\mathring{A}$, with equality in the limit of $\mathfrak{T} \to 0$. The same relationship holds between relaxations of the log-partition function and the linear programming relaxation of $\mathring{A}$, which optimizes over $\mathbb{L}(G)$ instead of $\mathbb{M}(G)$.

For $\mathfrak{T} > 0$, one can evaluate the smoothed relaxation at $\theta(\mathbf{x}; \mathbf{w}) + \mathbf{e}(\mathbf{y})$ in the M3N objective and obtain the gradient

$$\nabla \tilde{\mathcal{O}}_{\text{M3N}}(\mathbf{w}) = C\mathbf{w} + \sum_{(\mathbf{x},\mathbf{y})} [\mathbf{B}(\mathbf{x})]^{\mathsf{T}} [\boldsymbol{\tau}(\mathbf{x}; \mathbf{w}) - \boldsymbol{\phi}(\mathbf{y})], \tag{165}$$

where $\boldsymbol{\tau}(\mathbf{x}; \mathbf{w})$ refers to the pseudo-marginals that solve the smooth, loss-augmented variational problem.

Exactly how the inner variational problem is best solved depends on the entropy approximation in use. Since the quality of the entropy approximation does not matter in this case, it is perhaps best to use a simple approximation like H_{TRIV}. The variational problem can then again be solved using norm-product belief propagation,[164] with the difference to the CRF-case being that the counting number of each factor is $\mathfrak{T}$, rather than 1.

Due to smoothing, the overall objective function is differentiable in $\mathbf{w}$, so almost any solver for unconstrained convex problems can be applied.

Relation to previous work. The relationship between A and $\mathring{A}$ has repeatedly been noted in the literature. For instance, Pletscher et al.[165] introduce a "continuum" of approximations for discriminative training based on the temperature parameter. The special case of $\mathfrak{T} = 1$ is referred to as the "Softmax Margin CRF" by Gimpel and Smith.[166]

In both case, the authors restrict their consideration to the *exact* case. Recently, Hazan and Urtasun[167] extended the approach to approximations over the local polytope.

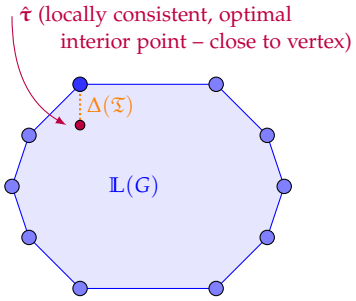


Figure 28: As $\mathfrak{T}$ decreases, the optimal pseudo-marginals are gradually drawn towards a vertex, which can either be integral or fractional.

[164] Tamir Hazan and Amnon Shashua. Norm-product belief propagation: Primal-dual message-passing for approximate inference. *IEEE Transactions on Information Theory*, 56(12):6294–6316, 2010

[165] Patrick Pletscher, Cheng Soon Ong, and Joachim M. Buhmann. Entropy and Margin Maximization for Structured Output Learning. In *European Conference on Machine Learning (ECML)*, 2010

[166] Kevin Gimpel and Noah A. Smith. Softmax-Margin CRFs: Training Log-Linear Models with Cost Functions. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010

[167] Tamir Hazan and Rachel Urtasun. A Primal-Dual Message-Passing Algorithm for Approximated Large Scale Structured Prediction. In *Advances in Neural Information Processing Systems*, 2010

M3N Recipe 2 (LP Relaxation Formulation)
Pick a decomposing loss term $\mathbf{e}$ and solve the convex optimization problem

$$\underset{\mathbf{w}}{\text{minimize}} \quad \tilde{\mathcal{O}}_{\text{M3N}}(\mathbf{w}) \stackrel{\text{def}}{=} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \left[E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) + \max_{\boldsymbol{\tau} \in \mathbb{L}(\mathbf{x})} \{\langle \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) + \mathbf{e}(\mathbf{y}), \boldsymbol{\tau} \rangle\}\right]$$

by running an off-the-shelf LP solver or a message passing algorithm yielding a fractional primal-optimal solution $\boldsymbol{\tau}(\mathbf{x}; \mathbf{w})$ required to compute a subgradient for each example, at each step of an iterative optimization algorithm for non-differentiable problems.

Training via Linear Programming Relaxations

In the context of max-margin Markov networks, perhaps the most obvious and most well-studied[168] approach is to estimate the parameters directly using linear programming relaxations of *Å*.

Unlike the log-partition function, there is no need for an additional approximation of the entropy term. In exchange, the variational problem is not strictly convex, so the maximum need not be attained uniquely. As a consequence, the overall objective is not differentiable in the model parameters. Nonetheless, one can obtain a subgradient

$$\tilde{\mathbf{g}}_{\text{M3N}}(\mathbf{w}) = C\mathbf{w} + \sum_{(\mathbf{x},\mathbf{y})} [\mathbf{B}(\mathbf{x})]^{\mathsf{T}} [\boldsymbol{\tau}(\mathbf{x}; \mathbf{w}) - \boldsymbol{\phi}(\mathbf{y})], \tag{166}$$

where $\boldsymbol{\tau}(\mathbf{x}; \mathbf{w})$ denotes one of possibly many solutions of the linear program. If the LP relaxation is tight, $\boldsymbol{\tau}(\mathbf{x}; \mathbf{w})$ will be a *vertex*, such that $\boldsymbol{\tau}(\mathbf{x}; \mathbf{w}) = \boldsymbol{\phi}(\mathbf{y})$ for some $\mathbf{y} \in \mathcal{Y}$. In general, the solution will be fractional.

The options for solving the inner problem are somewhat limited. Of course, one can in principle use an off-the-shelf linear programming solver, but in practice, it is desirable to exploit the problem structure.

This requires a message passing algorithm that allows to retrieve the *primal* solution of the LP relaxation. Algorithms that are guaranteed to find the optimum *and* yield the primal solution have recently been introduced by Ravikumar et al.[169] and Martins et al.,[170] but compared to typical message passing algorithms that only yield *dual* solutions, they carry a large constant overhead. Learning using the LP relaxation is perhaps most practical for problems with binary variables; in this special case, an optimal primal solution can be recovered trivially from a dual solution.

To solve the outer problem, the venerable subgradient method is in principle applicable, but provides only a sub-linear convergence rate. If high precision is desired, *bundle* methods specifically designed for regularized risk functionals[171] are a better choice. Analogously to stochastic gradient descent, stochastic sub-gradient methods[172] have also been developed that provide faster initial convergence on large, redundant datasets and are applicable to non-differentiable problems of the above kind.

Learning using linear programming relaxations has been considered in numerous application areas, for instance *dependency parsing* in natural language processing.[173]

[168] Alex Kulesza and Fernando Pereira. Structured Learning with Approximate Inference. In *Advances in Neural Information Processing Systems (NIPS)*, 2007; and Thomas Finley and Thorsten Joachims. Training structural SVMs when exact inference is intractable. In *International Conference on Machine Learning (ICML)*, 2008

[169] Pradeep Ravikumar, Alekh Agarwal, and Martin J. Wainwright. Message-passing for Graph-structured Linear Programs: Proximal Methods and Rounding Schemes. *Journal of Machine Learning Research*, 11:1043–1080, 2010

[170] André F. T. Martins, Mário A. T. Figueiredo, Pedro M. Q. Aguiar, Noah A. Smith, and Eric P. Xing. An Augmented Lagrangian Approach to Constrained MAP Inference. In *International Conference on Machine Learning (ICML)*, 2011

[171] Choon Hui Teo, S. V. N. Vishwanathan, Alex Smola, and Quoc V. Le. Bundle Methods for Regularized Risk Minimization. *Journal of Machine Learning Research*, 11:311–365, 2010

[172] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *International Conference on Machine Learning (ICML)*, 2007

[173] André F. T. Martins, Noah A. Smith, and Eric P. Xing. Polyhedral outer approximations with application to natural language parsing. In *International Conference on Machine Learning (ICML)*, 2009

Related Approaches

All "recipes" we considered so far had in common that they solve a well-motivated convex relaxation of the original problem, yielding an upper bound on the exact optimum. Various other approximations have been suggested in the literature that are closely connected but expose undesirable properties. In the following, we will review two of the most common approaches and discuss their shortcomings.

Training via loopy belief propagation. Prior to the advent of convex message passing algorithms, perhaps the most common approach was to obtain approximate marginals,[174] or approximate MAP states using loopy belief propagation. As we already intimated, loopy belief propagation (in its sum-product variant) seeks to minimize the Bethe free energy.

The problem with

$$A_{\text{BETHE}}(\theta) \stackrel{\text{def}}{=} \arg\max_{\tau \in \mathbb{L}(G)} \{\langle \theta, \tau \rangle + H_{\text{BETHE}}(\tau)\} \tag{167}$$

is that in principle, the function is convex in θ, but only if the variational problem can be solved exactly. Alas, this is not the case, since the objective function is not convex in the pseudo-marginals τ over which it is optimized. Even if we could find the optimum, A_{BETHE} need not be smooth, since the maximum need not be attained uniquely (the Bethe approximation to the entropy does not guarantee strict convexity).

From a practical perspective, this renders optimization very difficult. From a theoretical viewpoint, recent results of Heinemann and Globerson[175] characterize the implications of these problems on learning (although in the generative setting): In particular, unlike for the exact case or convex relaxations, there exists data for which moment-matching[176] does not happen: even at the optimum, the marginals under the model will not be equal to the empirical marginals.

Training M3Ns by running (max-product) loopy belief propagation to obtain approximate MAP states, and using these instead of exact maximizers to obtain a subgradient with respect to the model parameters is even less well-motivated. In particular, as Finley and Joachims[177] show, this approach is *undergenerating*, in that the model cannot realize all vertices of the marginal polytope during training. As a consequence, vertices that cannot be realized cannot be *penalized* either. A problem in particular is incompatibility with approximate maximum a-posteriori prediction at test time: models trained in the above way tend to promote fractional solutions.

In contrast, training using LP relaxations is *overgenerating*, in that the model can realize a superset of the feasible points. This leads to fractional solutions being penalized during training and increases the chances of integral solutions being found by approximate MAP prediction at test time.

Training via the mean field approximation. Discriminative training using *mean field* approximations has also been suggested by several authors, for instance by Vishwanathan et al., and more recently Weinman et al.[178].

In mean field approximations, one optimizes over a tractable *subset* of the marginal polytope. In the *naïve* mean field approximation in particular,

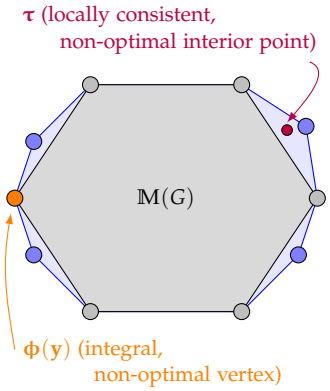


Figure 29: In training via sum-product loopy belief propagation, one uses approximate marginals obtained from the Bethe approximation. Since the global optimum cannot be found, convexity of the overall objective breaks down. Using max-product belief propagation, one employs non-optimal integral vertices. This approach is *undergenerating* and does not perform well.

[174] S. V. N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy. Accelerated Training of Conditional Random Fields with Stochastic Gradient Methods. In *International Conference on Machine Learning (ICML)*, 2006; and Charles Sutton and Andrew McCallum. An Introduction to Conditional Random Fields for Relational Learning. In Lise Getoor, editor, *Introduction to Statistical Relational Learning*, pages 93–128. MIT Press, 2007

[175] Uri Heinemann and Amir Globerson. What Cannot be Learned with Bethe Approximations. In *Uncertainty in Artificial Intelligence (UAI)*, 2011

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

[177] Thomas Finley and Thorsten Joachims. Training structural SVMs when exact inference is intractable. In *International Conference on Machine Learning (ICML)*, 2008

[178] Jerod J. Weinman, Lam Tran, and Christopher J. Pal. Efficiently Learning Random Fields for Stereo Vision with Sparse Message Passing. In *European Conference on Computer Vision (ECCV)*, 2008

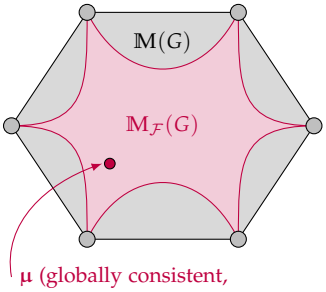


Figure 30: Training via approximate marginals obtained from the naive mean field approximation: The set of fully factorized marginals is non-convex, so the optimal marginals cannot in general be found. This causes convexity of the overall learning objective to break down.

this subset corresponds to *fully* factorized distributions, such that

$$\mu_{st}(y_s, y_t) \stackrel{!}{=} \mu_s(y_s)\mu_t(y_t), \quad \forall (s,t), (y_s, y_t). \tag{168}$$

A sketch of the set of $\mathbb{M}_{\mathcal{F}}(G)$ of fully factorized marginals is shown in Figure 30. In general, this set is not convex.[179] One may choose to explicitly enforce the constraints, moving them into the objective function of the variational problem; this overcomes non-convexity of the constraint set, but in turn results in a non-concave objective function.

The implications for conditional random field training are two-fold: First, as in the case of the Bethe approximation, while the mean field approximation to the log-partition function $A_{\mathcal{F}}(\theta)$ is in principle convex in θ, convexity only holds up if the maximization over μ can be carried out exactly, which is intractable. Second, somewhat analogously to the concept of *undergeneration* set forth by Finley and Jochaims[180] for training using approximate MAP states, a model trained using mean field approximations can only realize, and hence penalize, a subset of the valid marginals during training. While the implications of this undesirable property have not to our knowledge been investigated empirically in the context of conditional random field training, the results of Finley and Joachims still suggest that one should be cautious of any potential detrimental effects.

The Reparameterization Viewpoint

So far, we discussed the natural approach of *relaxing* the inner inference problems that must be solved during training of conditional random fields and max-margin Markov networks. We also saw other approaches that are commonly followed in the literature and discussed undesirable properties of these approaches.

We will now follow up with a different approach that builds on the *convexity* of the relaxations we previously used. For inference, we already saw that optimization of free energies over the local polytope and optimization of LP relaxations is dually coupled to a certain class of *bound tightening* problems. This duality can also be exploited for discriminative training, as we are going to point out shortly.

In particular, these bounds are constructed as follows: Decompose a given cyclic graph G into tractable subgraphs or pieces $R \in \mathcal{R}(G)$, and approximate the log-partition function $A(\theta)$ by a weighted sum of the log-partition functions $A_R(\lambda(R))$ of the pieces,

$$\tilde{A}(\theta) = \textstyle\sum_R c(R) A_R(\lambda(R)), \qquad \sum_R \lambda(R) \stackrel{!}{=} \theta, \tag{169}$$

where the tractable set of parameters $\{\lambda(R)\}_{R \in \mathcal{R}(G)}$ associated with the pieces must add up the original vector $\theta \in \mathbb{R}^d$ of exponential parameters. For a piece R to be *tractable*, a subset of its parameters $\lambda(R)$ is bound to be zero. We use the index set $\mathcal{I}(R) \subset \{1, 2, \ldots, d\}$ to denote the indices of non-zero components of a piece R.

Reparameterization. We say that the tractable parameters $\{\lambda(R)\}$ induce a reparameterization of θ if the condition

$$\textstyle\sum_R \langle c(R)\lambda(R), \phi(\mathbf{y})\rangle = \langle \theta, \phi(\mathbf{y})\rangle, \quad \forall \mathbf{y} \in \mathcal{Y}, \tag{170}$$

[179] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008

[180] Thomas Finley and Thorsten Joachims. Training structural SVMs when exact inference is intractable. In *International Conference on Machine Learning (ICML)*, 2008

holds, that is, the sum over pieces assigns the same *energy* to each joint state that it receives under the original parameter vector. It can easily be seen that this is the case if the $\{\boldsymbol{\lambda}(R)\}$ parameters add up to $\boldsymbol{\theta}$.

Duality. For a suitable choice of pieces R and associated weights $c(R)$, (169) forms an upper bound on the exact log-partition function. An obvious attempt is then to find the *tightest* such bound, by minimizing over $\{\boldsymbol{\lambda}(R)\}$, subject to the reparameterization constraint. Indeed, this problem is related to the variational problems we previously considered through convex duality, that is,

$$\min_{\substack{\{\boldsymbol{\lambda}(R)\}:\\ \sum_R c(R)\boldsymbol{\lambda}(R)=\boldsymbol{\theta}}} \sum_R c(R) A_R(\boldsymbol{\lambda}(R)) = \max_{\boldsymbol{\tau}\in\mathbb{L}(G)\supseteq\mathbb{M}(G)} \{\langle\boldsymbol{\theta},\boldsymbol{\tau}\rangle + \tilde{H}(\boldsymbol{\tau})\} \tag{171}$$

for some region-based entropy approximation $\tilde{H}$, the precise form of which arises from weighted entropy terms contributed by the tractable pieces. For typical choices of pieces, the constraint set $\mathbb{L}(G)$ is precisely the *first-order* local polytope we discussed previously.

Approximate MAP. A similar approach can be followed to obtain an approximation to $\mathring{A}(\boldsymbol{\theta})$, the maximum a-posteriori function. Since the entropy term is absent in this case, the choice of weights $c(R)$ does not matter and we can simply assume that they are all 1, obtaining

$$\min_{\substack{\{\boldsymbol{\lambda}(R)\}:\\ \sum_R \boldsymbol{\lambda}(R)=\boldsymbol{\theta}}} \sum_R A_R(\boldsymbol{\lambda}(R)) = \max_{\boldsymbol{\tau}\in\mathbb{L}(G)\supseteq\mathbb{M}(G)} \langle\boldsymbol{\theta},\boldsymbol{\tau}\rangle. \tag{172}$$

The choice of pieces only affects the outer approximation $\mathbb{L}(G)$, and for typical choices, this will again be the first-order local polytope.

Examples. Our claims are best understood by means of concrete examples. The *tree-reweighed* bounds of Wainwright et al.[181] map into this framework as follows: The pieces are chosen as spanning trees T of the graph, each associated with a probability $\rho(T)$. As the authors show, the dual relation

$$\min_{\substack{\{\boldsymbol{\lambda}(T)\}:\\ \sum_T \rho(T)\boldsymbol{\lambda}(T)=\boldsymbol{\theta}}} \sum_T \rho(T) A(\boldsymbol{\lambda}(T)) = \max_{\boldsymbol{\tau}\in\mathbb{L}(G)} \{\langle\boldsymbol{\theta},\boldsymbol{\tau}\rangle + H_{\text{TRW}}(\boldsymbol{\tau})\} \tag{173}$$

holds, where the approximate entropy $H_{\text{TRW}}(\boldsymbol{\tau}) = \sum_s H(\boldsymbol{\tau}_s) - \sum_{(s,t)} \nu_{st} I(\boldsymbol{\tau}_{st})$ depends on the coefficients $\rho(T)$ via the edge occurrence probabilities ν_{st}.

As another example, consider the node-splitting piecewise upper bound of Sutton and McCallum,[182] which approximates the log-partition function by a sum of per-factor log-partition functions, i.e. $A_{\text{PW}}(\boldsymbol{\theta}) = \sum_F A_F(\boldsymbol{\theta})$. As we show in the appendix of this section, by tightening over the reparameterizations, one obtains

$$\min_{\substack{\{\boldsymbol{\lambda}(F)\}:\\ \sum_F \boldsymbol{\lambda}(F)=\boldsymbol{\theta}}} \sum_F A_F(\boldsymbol{\lambda}(F)) = \max_{\boldsymbol{\tau}\in\mathbb{L}(G)} \{\langle\boldsymbol{\theta},\boldsymbol{\tau}\rangle + \sum_F H(\boldsymbol{\tau}_F)\}, \tag{174}$$

a variational problem involving the *trivial* entropy approximation of Weiss et al.,[183] providing an interesting interpretation of piecewise training.

By substituting $\mathring{A}$ for A, both cases reduce to the linear programming relaxation over the first-order local polytope, so for M3N training, the choice of pieces matters only for computational reasons.

[181] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005; and Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005

[182] Charles Sutton and Andrew McCallum. Piecewise training for structured prediction. *Machine learning*, 77(2):165–194, 2009

[183] Yair Weiss, Chen Yanover, and Talya Meltzer. MAP Estimation, Linear Programming and Belief Propagation with Convex Free Energies. In *Uncertainty in Artificial Intelligence (UAI)*, 2007

CRF Recipe 3 (Consecutively Tightened Reparameterization Formulation)
Choose a decomposition of each example into tractable subgraphs $R \in \mathcal{R}(\mathbf{x})$ with associated weights $c(R)$ and index set $\mathcal{I}(R) \subset \{1, 2, \ldots, d(\mathbf{x})\}$, and solve the convex optimization problem

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad \tilde{\mathcal{O}}_{\text{CRF}}(\mathbf{w}) \stackrel{\text{def}}{=} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \Big[E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) + \min_{\{\boldsymbol{\lambda}(R)\}} \; & \textstyle\sum_{R \in \mathcal{R}(\mathbf{x})} c(R) A_R(\boldsymbol{\lambda}(R)) \Big] \\ \text{s.t.} \; & \textstyle\sum_R c(R) \boldsymbol{\lambda}(R) = \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) \\ & \lambda_\alpha(R) = 0, \; \forall R, \alpha \notin \mathcal{I}(R) \end{aligned}$$

by solving, for each example, the inner reparameterization problem subject to the reparameterization and structural constraints (handled explicitly or implicitly) to obtain the marginals $\boldsymbol{\mu}_R(\mathbf{x}; \mathbf{w})$ of each tractable subgraph, needed to compute the full gradient at each step of an iterative optimization algorithm.

Consecutive Tightening of the Log-Partition Function

We can exploit the duality relations between *reparameterization* and the variational problems we previously considered in CRF training.

In particular, as shown in CRF Recipe 3, instead of solving a variational problem over the local polytope in order to approximate the log-partition function, we can equivalently solve a reparameterization problem for suitably chosen pieces and associated weights.

One feasible way of doing so is to solve the reparameterization problem for each example at each iteration. Note that the overall objective is convex in the $\{\boldsymbol{\lambda}(R)\}$ parameters. By standard results in convex optimization,[184] partial minimization over these parameters then preserves convexity of the problem. Each such inner problem can be solved efficiently, for instance, using projected gradient methods, as in our TightenBound algorithm.

[184] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004

How can the outer problem, the minimization with respect to $\mathbf{w}$, be solved? A slight complication is that—as presented—$\mathbf{w}$ enters the inner problem only through the reparameterization constraints. Nonetheless, the solution to this problem is actually a function of $\mathbf{w}$.

To obtain the gradient with respect to $\mathbf{w}$, we use the fact the reparameterization constraints can be eliminated altogether. In particular, observe that only *equality* constraints are present. For each index $\alpha \in \mathcal{I}$, we can thus simply pick any piece R_α with $\alpha \in \mathcal{I}(R_\alpha)$, and solve for the associated parameter $\lambda_\alpha(R_\alpha)$, obtaining

$$\lambda_\alpha(R_\alpha) = \frac{1}{c(R_\alpha)} \theta_\alpha(\mathbf{x}; \mathbf{w}) - \sum_{R \in \mathcal{R} \setminus R_\alpha} \frac{c(R)}{c(R_\alpha)} \lambda_\alpha(R), \tag{175}$$

which allows us to move $\boldsymbol{\theta}(\mathbf{x}; \mathbf{w})$ into the objective function. Each component $\boldsymbol{\theta}_\alpha$ enters the objective only through the single chosen piece R_α, which we can now differentiate:

$$\frac{c(R_\alpha) \partial A_{R_\alpha}(\cdot)}{\partial \theta_\alpha} = \frac{\cancel{c(R_\alpha)}}{\cancel{c(R_\alpha)}} \mu_\alpha(R_\alpha), \tag{176}$$

where $\mu_\alpha(R_\alpha)$ denotes the marginal probability of a particular state in the

reparameterized piece R_α. Interestingly, as we have already seen in the context of our TIGHTENBOUND algorithm, at optimum, the reparameterized pieces satisfy the stationary condition

$$\mu_\alpha(R) \stackrel{!}{=} \tau_\alpha(\mathbf{x};\mathbf{w}), \quad \forall R\colon \alpha \in \mathcal{I}(R), \tag{177}$$

that is, the marginals of the reparameterized pieces are all equal to a single pseudo-marginal $\tau_\alpha(\mathbf{x};\mathbf{w})$, where $\boldsymbol{\tau}(\mathbf{x};\mathbf{w})$ denotes the solution of the dually coupled variational inference problem.

The comforting—and expected—result is that the gradient with respect to the model parameters is equivalent to the gradient we previously obtained for CRF Recipe 1,

$$\nabla\tilde{\mathcal{O}}_{\text{CRF}}(\mathbf{w}) = C\mathbf{w} + \sum_{(\mathbf{x},\mathbf{y})} [\mathbf{B}(\mathbf{x})]^\mathsf{T}[\boldsymbol{\tau}(\mathbf{x};\mathbf{w}) - \boldsymbol{\phi}(\mathbf{y})]. \tag{178}$$

The gradient can then be employed in an off-the-shelf iterative algorithm for unconstrained convex optimization, as previously.

One may wonder what has been gained over CRF Recipe 1. For CRF training, the practical gains are little, since the inner variational problem is strictly convex and the pseudo-marginals can be trivially obtained by solving either the bound-tightening or the free energy minimization problem. However, the importance of this formulation lies in the fact that the training objective is actually *jointly* convex in the model and the reparameterization parameters, which we are going to exploit in a subsequent recipe.

Moreover, as we shall see, the dual coupling between equivalent formulations of the inference sub-problem is more subtle for M3Ns, such that the reparameterization formulation *does* have practical benefits.

Relation to previous work. From a theoretical viewpoint, the reparameterization formulation also enables us to see the close connection to related approaches, which is interesting in its own right.

For instance, by choosing the tractable pieces as spanning trees, one obtains a tightened version of the spanning-tree approximation that has been suggested by Pletscher et al.[185] for conditional random field training. If a loser approximation is sufficient, one can simply omit minimization over the $\{\boldsymbol{\lambda}(R)\}$ parameters in CRF Recipe 3 to obtain precisely such an approach, which is computationally more attractive.

If, on the other hand, one chooses the pieces as single factors, one obtains a tightened variant of the piecewise training approach suggested by Sutton and McCallum.[186] As we already intimated, this approach is equivalent to learning with the *trivial* approximation to the log-partition function suggested by Weiss et al.[187]. The close connection between piecewise training and this particular approximation was already pointed out by Ganapathi et al.,[188] but motivated from the dual perspective to ours, namely in terms of adding local polytope constraints to the per-factor log-partition functions to enforce consistency of their pseudo-marginals. In contrast, we arrived at this result by *reparameterizing* the per-factor log-partition functions—a different perspective, which yields, however, a mathematically equivalent approximation.

[185] Patrick Pletscher, Cheng Soon Ong, and Joachim M. Buhmann. Spanning Tree Approximations for Conditional Random Fields. In *Artificial Intelligence and Statistics (AISTATS)*, 2009

[186] Charles Sutton and Andrew McCallum. Piecewise training for structured prediction. *Machine learning*, 77(2):165–194, 2009

[187] Yair Weiss, Chen Yanover, and Talya Meltzer. MAP Estimation, Linear Programming and Belief Propagation with Convex Free Energies. In *Uncertainty in Artificial Intelligence (UAI)*, 2007

[188] Varun Ganapathi, David Vickrey, John Duchi, and Daphne Koller. Constrained Approximate Maximum Entropy Learning. In *Uncertainty in Artificial Intelligence (UAI)*, 2008

M3N Recipe 3 (Consecutively Tightened Reparameterization Formulation)
Choose a decomposition of each example into tractable subgraphs $R \in \mathcal{R}(\mathbf{x})$ with associated index set $\mathcal{I}(R) \subset \{1, 2, \ldots, d(\mathbf{x})\}$, and solve the convex optimization problem

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad \tilde{\mathcal{O}}_{\text{M3N}}(\mathbf{w}) \stackrel{\text{def}}{=} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \left[E(\mathbf{y} \,|\, \mathbf{x}; \mathbf{w}) + \min_{\{\boldsymbol{\lambda}(R)\}} \; \textstyle\sum_{R \in \mathcal{R}(\mathbf{x})} \mathring{A}_R(\boldsymbol{\lambda}(R)) \right] \\ \text{s.t.} \; \textstyle\sum_R \boldsymbol{\lambda}(R) = \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) + \mathbf{e}(\mathbf{y}) \\ \lambda_\alpha(R) = 0, \; \forall R, \alpha \notin \mathcal{I}(R) \end{aligned}$$

by solving, for each example, the inner reparameterization problem subject to reparameterization and structural constraints (handled explicitly or implicitly) to obtain a maximizing state $\hat{\mathbf{y}}_R(\mathbf{x}; \mathbf{w})$ of each tractable subgraph, needed to compute a subgradient at each step of a non-differentiable optimization scheme.

Consecutive Tightening of the Maximum A-Posteriori Function

A similar reparameterization approach may be followed for M3N training. As we intimated in the beginning of this section, any choice of pieces will result in a linear programming relaxation of the MAP function, and both for single factor pieces and spanning trees, the resulting constraint set is precisely the local polytope $\mathbb{L}(G)$.

The reparameterization approach has a practical advantage over solving a linear programming over the local polytope: In fact, most message passing algorithms belong to the family of *bound tightening* algorithms[189] that implicitly solve the reparameterization problems. As such, one cannot easily obtain the solution of the dually coupled linear programming relaxation from them, except for special cases.

For instance, the IncMP algorithm we previously introduced can be used to solve the inner reparameterization problem very efficiently. Another suitable and efficient message passing algorithm is the augmented Lagrangian approach by Meshi et al.[190]

However, note that it is *not* a sound approach to minimize the bound using coordinate descent schemes such as MPLP[191] or TRW-S[192], which get stuck at sub-optimal solutions. In order to maintain convexity, it is crucial that the partial minimization is carried out exactly. Otherwise, one needs to maintain the $\{\boldsymbol{\lambda}(R)\}$ variables over iterations, which we are going to consider in the next recipe.

As was the case for reparameterization of the log-partition function, the inner problem is a function of $\mathbf{w}$. To see this, we can again move $\boldsymbol{\theta}(\mathbf{w}; \mathbf{x})$ into the objective via the relation

$$\lambda_\alpha(R_\alpha) = \theta_\alpha(\mathbf{x}; \mathbf{w}) + e_\alpha(\mathbf{y}) - \sum_{R \in \mathcal{R} \setminus R_\alpha} \lambda_\alpha(R). \tag{179}$$

However, as is to be expected, the function is non-smooth in $\boldsymbol{\theta}(\mathbf{x}; \mathbf{w})$ and hence $\mathbf{w}$. Nonetheless, after solving the reparameterization problem to optimality, one can obtain a subgradient with respect to $\boldsymbol{\theta}(\mathbf{x}; \mathbf{w})$ in terms of

[189] Talya Meltzer, Amir Globerson, and Yair Weiss. Convergent message passing algorithms - A unifying view. In *Uncertainty in Artificial Intelligence (UAI)*, 2009

[190] Ofer Meshi and Amir Globerson. An Alternating Direction Method for Dual MAP LP Relaxation. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2011

[191] Amir Globerson and Tommi Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *Advances in Neural Information Processing Systems*, 2007

[192] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568 – 1583, 2006

the containing pieces via

$$\mathring{g}_\alpha(\theta_\alpha(\mathbf{x};\mathbf{w})) = \phi_\alpha(\hat{\mathbf{y}}_R(\mathbf{x};\mathbf{w})), \qquad \alpha \in \mathcal{I}(R), \tag{180}$$

where $\hat{\mathbf{y}}_R(\mathbf{x};\mathbf{w})$ is a maximizing state (after reparameterization) of any piece R containing index α, and $\boldsymbol{\phi}(\cdot)$ denotes the sufficient statistics of that state. Hence, the component of the subgradient is 1 if θ_α belongs to the maximizing state, and 0 otherwise.

Note that this leaves several degrees of freedom: first of all, the maximum need not be attained uniquely in a piece, and second, in constructing the subgradient, one can choose among the pieces that contain index α. Unlike the previous case of the log-partition function, where the marginals of the pieces must agree at the optimum, the MAP states of the pieces need not agree. However, if they *do* agree, the LP relaxation was *tight*, and a subgradient constructed as above corresponds to the sufficient statistics of an exact joint MAP state $\hat{\mathbf{y}}(\mathbf{x},\mathbf{w})$, as in the case of the exact M3N objective.

Solving the outer problem. Using the above recipe to obtain a subgradient of the inner problem with respect to $\boldsymbol{\theta}(\mathbf{x};\mathbf{w})$, and by applying the chain rule, one obtains a subgradient of the overall objective with respect to $\mathbf{w}$,

$$\tilde{\mathbf{g}}_{\text{M3N}}(\mathbf{w}) = C\mathbf{w} + \sum_{(\mathbf{x},\mathbf{y})} [\mathbf{B}(\mathbf{x})]^\top [\mathring{\mathbf{g}}(\boldsymbol{\theta}(\mathbf{x};\mathbf{w})) - \boldsymbol{\phi}(\mathbf{y})]. \tag{181}$$

Using this subgradient, one can again employ a solver for non-differentiable regularized risk functionals,[193] or a stochastic subgradient method.[194]

Related work. We are unaware of this approach having been followed in the literature. As we pointed out, its convenience lies in the fact that one need not obtain primal solutions of the linear programming relaxation. Instead, it is sufficient to tighten the dual formulation, which is an easier task in practice, and indeed the route that is followed by most message passing algorithms.

[193] Choon Hui Teo, S. V. N. Vishwanathan, Alex Smola, and Quoc V. Le. Bundle Methods for Regularized Risk Minimization. *Journal of Machine Learning Research*, 11:311–365, 2010

[194] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *International Conference on Machine Learning (ICML)*, 2007

Exploiting Joint Convexity in All Parameters

The two previous recipes still rely on consecutive tightening of a reparameterization of the log-partition function or the MAP function as an *inner problem*, which must be solved repeatedly during the course of optimizing the overall objective function.

However, we already alluded to the fact that the overall objective function is actually *jointly convex* in both the model parameters $\mathbf{w}$ and the reparameterization parameters $\{\boldsymbol{\lambda}(R)\}$. An important consequence of this fact is that the inner problem need not actually be solved at each step, but rather can we include its parameters in the overall optimization process. That is, instead of defining our objective function in terms of partial minimization with respect to the $\{\boldsymbol{\lambda}(R)\}$, these parameters become part of the outer objective function and are optimized along with $\mathbf{w}$ at each step.

This transformation avoids the need to repeatedly run inference using a specialized solver as part of training. On the other hand, it results in a single very-large-scale convex optimization problem, the number of parameters of which grows linearly in the number of training examples.

CRF Recipe 4 (Jointly Convex Reparameterization Formulation)
Choose a decomposition of each example into tractable subgraphs $R \in \mathcal{R}(\mathbf{x})$ with associated weights $c(R)$ and index set $\mathcal{I}(R) \subset \{1, 2, \ldots, d(\mathbf{x})\}$, and solve the jointly convex optimization problem

$$\begin{aligned} \underset{\mathbf{w},\{\boldsymbol{\lambda}(R)\}}{\text{minimize}} \quad & \tilde{\mathcal{O}}_{\text{CRF}}\big(\mathbf{w}, \{\boldsymbol{\lambda}(R)\}\big) \stackrel{\text{def}}{=} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \big[E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) + \textstyle\sum_{R \in \mathcal{R}(\mathbf{x})} c(R) A_R(\boldsymbol{\lambda}(R))\big] \\ \text{s.t.} \quad & \textstyle\sum_R c(R)\boldsymbol{\lambda}(R) = \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}), \;\; \forall (\mathbf{x}, \mathbf{y}) \\ & \lambda_\alpha(R) = 0, \qquad\quad \forall (\mathbf{x}, \mathbf{y}), R, \alpha \notin \mathcal{I}(R) \end{aligned}$$

by computing, at each step of an iterative optimization algorithm and for each example, the marginals $\boldsymbol{\mu}_R(\mathbf{x}; \mathbf{w}, \boldsymbol{\lambda}(R))$ of each tractable subgraph for the current parameters to obtain the gradient, handling the reparameterization constraints implicitly by moving them into the objective function.

Exploiting Joint Convexity in CRF training

Let us make this strategy concrete for CRF training, where the $\{\boldsymbol{\lambda}(R)\}$ parameters serve the purpose of tightening a piecewise approximation of the log-partition function.

The formulation as shown in CRF Recipe 4 is explicit and most easily comprehensible, but for practical optimization purposes we shall again find it more convenient to eliminate the reparameterization constraints by moving them into the objective function. Again, for each index $\alpha \in \mathcal{I}$, we solve for the parameter of a chosen piece R_α to obtain

$$\lambda_\alpha(R_\alpha) = \frac{1}{c(R_\alpha)}\theta_\alpha(\mathbf{x}; \mathbf{w}) - \sum_{R \in \mathcal{R} \setminus R_\alpha} \frac{c(R)}{c(R_\alpha)}\lambda_\alpha(R), \tag{182}$$

and substitute into the objective function. For the model parameters, we then obtain the familiar (partial) gradient

$$\nabla_{\mathbf{w}} \tilde{\mathcal{O}}_{\text{CRF}}(\mathbf{w}, \{\boldsymbol{\lambda}(R)\}) = C\mathbf{w} + \sum_{(\mathbf{x},\mathbf{y})} [\mathbf{B}(\mathbf{x})]^{\mathsf{T}} [\boldsymbol{\tau}(\mathbf{x}; \mathbf{w}, \{\boldsymbol{\lambda}(R)\}) - \boldsymbol{\phi}(\mathbf{y})], \tag{183}$$

where the vector of pseudo-marginals is given component-wise by

$$\tau_\alpha\big(\mathbf{x}; \mathbf{w}, \{\boldsymbol{\lambda}(R)\}\big) \stackrel{\text{def}}{=} \frac{c(R_\alpha)\partial A_{R_\alpha}(\cdot)}{\partial \theta_\alpha} = \mu_\alpha(R_\alpha). \tag{184}$$

In other words, the components of $\boldsymbol{\tau}(\cdot)$ correspond to marginal probabilities of the designated pieces R_α, for the current parameters. At the joint global optimum, all pieces again yield (locally) consistent marginal probabilities, so it does not matter to which piece θ_α is assigned.

The gradient with respect to the tightening parameters is slightly more involved. Observe that each $\lambda_\alpha(R)$ occurs in two pieces: once in its own piece R with positive sign, and by (182) once in the designated piece R_α of index α, with negative sign. The partial derivative is then

$$\frac{\partial \tilde{\mathcal{O}}_{\text{CRF}}(\mathbf{w}, \{\boldsymbol{\lambda}(R)\})}{\partial \lambda_\alpha(R)} = c(R)[\mu_\alpha(R) - \mu_\alpha(R_\alpha)], \tag{185}$$

exposing the intuitive stationary condition of marginal consistency among the reparameterized pieces.

M3N Recipe 4 (Jointly Convex Reparameterization Formulation)
Choose a decomposition of each example into tractable subgraphs $R \in \mathcal{R}(\mathbf{x})$ with associated index set $\mathcal{I}(R) \subset \{1, 2, \ldots, d(\mathbf{x})\}$, and solve the jointly convex optimization problem

$$\begin{aligned} \underset{\mathbf{w}, \{\boldsymbol{\lambda}(R)\}}{\text{minimize}} \quad & \tilde{\mathcal{O}}_{\text{M3N}}\big(\mathbf{w}, \{\boldsymbol{\lambda}(R)\}\big) \stackrel{\text{def}}{=} \frac{C}{2}\|\mathbf{w}\|_2^2 + \sum_{(\mathbf{x},\mathbf{y})} \big[E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) + \textstyle\sum_{R \in \mathcal{R}(\mathbf{x})} \mathring{A}_R(\boldsymbol{\lambda}(R))\big] \\ \text{s.t.} \quad & \textstyle\sum_R \boldsymbol{\lambda}(R) = \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) + \mathbf{e}(\mathbf{y}), \quad \forall (\mathbf{x}, \mathbf{y}) \\ & \lambda_\alpha(R) = 0, \qquad\qquad \forall (\mathbf{x}, \mathbf{y}), R, \alpha \notin \mathcal{I}(R) \end{aligned}$$

by computing, at each step of an iterative optimization algorithm and for each example, a maximizing state $\hat{\mathbf{y}}_R(\mathbf{x}; \mathbf{w}, \boldsymbol{\lambda}(R))$ of each tractable subgraph for the current parameters to obtain a subgradient, handling the reparameterization constraints implicitly by moving them into the objective function.

From the above ingredients, one can construct a gradient with respect to all parameters. The problem is then unconstrained and convex. The main challenge is posed by the large number of parameters. It is thus advisable to either use a first-order gradient based method, or a limited-memory method such L-BFGS[195] that works using a fixed memory budget.

[195] Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980

Related work. Hazan and Urtasun[196] recently proposed a similar formulation, which can be understood as a special case of ours where the pieces consist of single factors. In their development, the tightening parameters arise as Lagrange multipliers corresponding to (first-order) marginalization constraints, which provides yet another viewpoint. Our approach is more general in that (for suitably chosen pieces) higher-order marginalization constraints can be enforced, at the cost of a larger number of parameters.

[196] Tamir Hazan and Rachel Urtasun. A Primal-Dual Message-Passing Algorithm for Approximated Large Scale Structured Prediction. In *Advances in Neural Information Processing Systems*, 2010

To actually solve the problem, Hazan and Urtasun suggest to perform stochastic gradient descent on the model parameters, and block coordinate updates (as in message passing) on the tightening parameters. Whether this is preferable to solving the whole problem over all parameters using a single iterative algorithm depends on a wide variety of characteristics of the task, such as the number of features, the redundancy in the dataset, and the cardinality of the variables. In any case, the possibility to construct a multitude of convergent optimization strategies is a major benefit of joint convexity over the previous "relaxation" perspective.

Exploiting Joint Convexity in M3N Training

The same strategy we used for CRF training can be used to exploit joint convexity in the context of M3N training (M3N Recipe 4).

The main difference is that, as previously, the sufficient statistics of a MAP state of a reparameterized piece must be used instead of the marginals of that piece, due to non-differentiability. In terms of these sufficient statistics, a subgradient with respect to the model and tightening parameters can then be constructed. A problem in this context is that unlike the model parameters $\mathbf{w}$, the tightening parameters $\{\boldsymbol{\lambda}(R)\}$ are not regularized, so

BMRM,[197] the solver for regularized risk functionals we previously suggested, is not immediately applicable.

A possible remedy is to include the tightening parameters $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}(R)\}$ in the regularization, scaled by a small number C_λ, such that the regularization term turns into

$$\frac{C_w}{2}\|\mathbf{w}\|_2^2 + \frac{C_\lambda}{2}\|\boldsymbol{\Lambda}\|_2^2. \tag{186}$$

Another practical approach is to approximate $\mathring{A}_R(\boldsymbol{\theta}) \approx \mathfrak{T} A_R(\boldsymbol{\theta}/\mathfrak{T})$ for a small number $\mathfrak{T}$, such that the objective function becomes smooth and L-BFGS[198] can be used. In both cases, if the constant is chosen small enough, the difference over the original formulation becomes negligible—and in any case, we are working with an approximation of $\mathring{A}(\boldsymbol{\theta})$ already.

Related work. Similar approaches have recently been followed independently by Hazan and Urtasun,[199] as well as Meshi et al.,[200] using slightly different, but mathematically equivalent, dual formulations. In both cases, the authors suggest to perform coordinate descent on the tightening parameters $\{\boldsymbol{\lambda}(R)\}$, and stochastic subgradient steps, akin to Pegasos,[201] on the $\mathbf{w}$ parameters. However, unlike the strictly convex case of the CRF objective, block coordinate updates are not guaranteed to converge to the optimum in this setting. This problem could be rectified by regularizing the tightening parameters, as we suggested above, such that the objective function becomes strictly convex.

As in the case of jointly convex CRF training, our formulation is more general than the above approaches, but equivalent for typical choices of decomposing the graphs into pieces.

Appendix: Dual of the piecewise reparameterization problem

Towards the beginning of this section, we claimed that the node-splitting piecewise training approach of Sutton and McCallum,[202] if tightened over reparameterizations, is equivalent to learning with the *trivial* approximation of Weiss et al.,[203]

$$A_{\text{TRIV}}(\boldsymbol{\theta}) = \max_{\boldsymbol{\tau}\in\mathbb{L}(G)} \big\{\langle\boldsymbol{\theta},\boldsymbol{\tau}\rangle + \textstyle\sum_F H(\boldsymbol{\tau}_F)\big\}. \tag{187}$$

We now wish to show that this is indeed the case. Remember that piecewise training approximates the intractable log-partition function via

$$A_{\text{PW}}(\boldsymbol{\theta}) = \sum_F A_F(\boldsymbol{\theta}), \tag{188}$$

with

$$A_F(\boldsymbol{\theta}) = \log\big(\textstyle\sum_{\mathbf{y}_F} \exp\langle\boldsymbol{\theta},\boldsymbol{\phi}(\mathbf{y}_F)\rangle\big). \tag{189}$$

This is a special case of our framework, where each tractable piece $R \in \mathcal{R}$ corresponds to a factor $F \in \mathcal{F}$ along with its variables $s \in F$.

As Sutton and McCallum note, the piecewise approximation provides an upper bound on the exact log-partition function. How can this bound be tightened in our framework? Observe that the exponential parameters $\boldsymbol{\theta}_F$ corresponding to a factor F occur precisely in a single piece, given precisely

[197] Choon Hui Teo, S. V. N. Vishwanathan, Alex Smola, and Quoc V. Le. Bundle Methods for Regularized Risk Minimization. *Journal of Machine Learning Research*, 11:311–365, 2010

[198] Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980

[199] Tamir Hazan and Amnon Shashua. Norm-product belief propagation: Primal-dual message-passing for approximate inference. *IEEE Transactions on Information Theory*, 56(12):6294–6316, 2010

[200] Ofer Meshi, David Sontag, Tommi Jaakkola, and Amir Globerson. Learning Efficiently with Approximate Inference via Dual Losses. In *International Conference on Machine Learning (ICML)*, 2010

[201] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *International Conference on Machine Learning (ICML)*, 2007

[202] Charles Sutton and Andrew McCallum. Piecewise training for structured prediction. *Machine learning*, 77(2):165–194, 2009

[203] Yair Weiss, Chen Yanover, and Talya Meltzer. MAP Estimation, Linear Programming and Belief Propagation with Convex Free Energies. In *Uncertainty in Artificial Intelligence (UAI)*, 2007

by that factor. Hence, in a reparameterization $\{\boldsymbol{\lambda}(F)\}$ of $\boldsymbol{\theta}$, the parameters $\{\boldsymbol{\lambda}_F(F)\}$ are already fully determined by $\boldsymbol{\lambda}_F(F) = \boldsymbol{\theta}_F$. However, variables are shared among possibly multiple factors, hence the exponential parameters corresponding to these can be varied.

We use $\boldsymbol{\lambda}_s(F)$ to denote the tightening parameters associated with variable $s \in F$. Note that there are no components corresponding to variables in $\boldsymbol{\theta}$, hence their exponential parameters are zero in the original parameterization, and we obtain the reparameterization constraint

$$\sum_{F\colon s\in F} \boldsymbol{\lambda}_s(F) = \mathbf{0}, \quad \forall s \in V. \tag{190}$$

Our claim is now that

$$\begin{aligned} \tilde{A}(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \min_{\{\boldsymbol{\lambda}(F)\}} \quad & \textstyle\sum_F A_F(\boldsymbol{\lambda}(F)) \\ \text{s.t.} \quad & \textstyle\sum_{F:s\in F} \boldsymbol{\lambda}_s(F) = \mathbf{0},\ \forall s \in V \end{aligned} \tag{191}$$

is in fact equivalent to $A_{\text{TRIV}}(\boldsymbol{\theta})$, defined in (187). To establish equivalence, we proceed by forming the Lagrangian of the above problem, using suggestively named Lagrange multipliers $\boldsymbol{\tau}_s$ associated with the equality constraints, and obtain

$$L\big(\{\boldsymbol{\lambda}(F)\}, \{\boldsymbol{\tau}_s\}\big) = \sum_F A_F(\boldsymbol{\lambda}(F)) - \sum_s \big\langle \boldsymbol{\tau}_s, \textstyle\sum_{F:s\in F} \boldsymbol{\lambda}_s(F)\big\rangle \tag{192}$$

$$= \sum_F A_F(\boldsymbol{\lambda}(F)) - \sum_F \textstyle\sum_{s\in F} \langle \boldsymbol{\lambda}_s(F), \boldsymbol{\tau}_s\rangle. \tag{193}$$

Next, by taking the partial derivative of L with respect to a single $\boldsymbol{\lambda}_s(F)$, we obtain the stationary conditions

$$\boldsymbol{\mu}_{F\to s} \stackrel{!}{=} \boldsymbol{\tau}_s, \quad \forall F\colon s \in F, \tag{194}$$

where we use $\boldsymbol{\mu}_{F\to s}(y_s) = \sum_{\mathbf{y}_F \sim y_s} \boldsymbol{\mu}_F(\mathbf{y}_F)$ to denote the marginals $\boldsymbol{\mu}_F$ resulting from the optimal $\hat{\boldsymbol{\lambda}}(F)$, marginalized for s. Consequently, at the optimum, the marginal probabilities of the single-factor pieces all agree.

We continue by inspecting the variational representation of A_F,

$$A_F(\boldsymbol{\lambda}(F)) = \max_{\boldsymbol{\mu}_F \in \mathbb{M}(F)} \Big\{ \langle \boldsymbol{\theta}_F, \boldsymbol{\mu}_F\rangle + \textstyle\sum_s \langle \boldsymbol{\lambda}_s(F), \boldsymbol{\mu}_{F\to s}\rangle + H(\boldsymbol{\mu}_F) \Big\}.$$

Exploiting the stationary conditions, we can hence develop the Lagrange dual function as

$$\begin{aligned} g(\{\boldsymbol{\tau}_s\}) &= \max_{\{\boldsymbol{\tau}_s\}} \sum_F \big(A_F(\hat{\boldsymbol{\lambda}}(F)) - \textstyle\sum_{s\in F} \langle \hat{\boldsymbol{\lambda}}_s(F), \boldsymbol{\tau}_s\rangle \big) \\ &= \cancel{\max_{\{\boldsymbol{\tau}_s\}}} \sum_F \Bigg(\max_{\substack{\boldsymbol{\mu}_F \in \mathbb{M}(F),\\ \boldsymbol{\mu}_{F\to s} = \boldsymbol{\tau}_s, \forall s}} \big\{ \langle \boldsymbol{\theta}_F, \boldsymbol{\mu}_F\rangle + \cancel{\textstyle\sum_s \langle \hat{\boldsymbol{\lambda}}_s(F), \boldsymbol{\mu}_{F\to s}\rangle} + H(\boldsymbol{\mu}_F) \big\} \\ &\qquad\qquad \cancel{- \textstyle\sum_s \langle \hat{\boldsymbol{\lambda}}_s(F), \boldsymbol{\tau}_s\rangle} \Bigg) \\ &= \max_{\{\boldsymbol{\mu}_F\} \in \mathbb{L}(G)} \{ \textstyle\sum_F \langle \boldsymbol{\theta}_F, \boldsymbol{\mu}_F\rangle + \sum_F H(\boldsymbol{\mu}_F) \}, \end{aligned}$$

Note that for a given $\boldsymbol{\tau}_s$, the term $\sum_s \langle \hat{\boldsymbol{\lambda}}_s(F), \boldsymbol{\mu}_{F\to s}\rangle$ is a constant that does not influence the optimum of the inner problem (due to the marginalization constraints obtained from the stationary conditions).

where we used the fact that all terms involving the $\boldsymbol{\tau}_s$ variables cancel, and that the intersection of the local normalization constraints and the marginalization constraints of (194) forms the first-order local polytope. We have thus shown equivalence to (187), which concludes our proof.

Dualizing the Entire Objective: Exponentiated Gradient Training

In this section, we will exploit yet another different kind of duality relation. Previously, we showed that decomposing the intractable log-partition function or maximum a-posteriori function into pieces, and tightening these pieces over all reparameterizations, yields formulations that are equivalent to the original variational problems for a suitable choice of pieces and associated weights.

Another possibility, which results from quadratic regularization, is to eliminate the model parameters $\mathbf{w}$ altogether and optimize directly over the marginal probabilities. For conditional random fields, this dual objective function was derived by Lebanon and Lafferty[204] and is given by

[204] Guy Lebanon and John Lafferty. Boosting and Maximum Likelihood for Exponential Models. In *Advances in Neural Information Processing Systems*, 2001

$$\mathcal{Q}_{\text{CRF}}(\{\boldsymbol{\mu}_{\mathbf{x}}\}) \stackrel{\text{def}}{=} \frac{1}{2C}\big\|\textstyle\sum_{(\mathbf{x},\mathbf{y})}[\mathbf{B}(\mathbf{x})]^{\mathsf{T}}[\boldsymbol{\phi}(\mathbf{y}) - \boldsymbol{\mu}_{\mathbf{x}}]\big\|_2^2 - \textstyle\sum_{(\mathbf{x},\mathbf{y})} H\big(p_{\boldsymbol{\theta}(\boldsymbol{\mu}_{\mathbf{x}})}\big), \quad (195)$$

where the marginals $\boldsymbol{\mu}_{\mathbf{x}} \in \mathbb{M}(\mathbf{x})$ of each example are restricted to belong the marginal polytope.

A similar dual was derived by Taskar et al.[205] for max-margin Markov networks and is given by

[205] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-Margin Markov Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2003

$$\mathcal{Q}_{\text{M3N}}(\{\boldsymbol{\mu}_{\mathbf{x}}\}) \stackrel{\text{def}}{=} \frac{1}{2C}\big\|\textstyle\sum_{(\mathbf{x},\mathbf{y})}[\mathbf{B}(\mathbf{x})]^{\mathsf{T}}[\boldsymbol{\phi}(\mathbf{y}) - \boldsymbol{\mu}_{\mathbf{x}}]\big\|_2^2 - \textstyle\sum_{(\mathbf{x},\mathbf{y})}\langle \boldsymbol{\mu}_{\mathbf{x}}, \mathbf{e}(\mathbf{y})\rangle, \quad (196)$$

where the parameters associated with each example are again restricted to reside in the marginal polytope.

The dual formulations are then coupled to their primal form via

$$\min_{\{\boldsymbol{\mu}_{\mathbf{x}}\}\in\mathbb{M}^n} \mathcal{Q}_{\text{CRF}}(\{\boldsymbol{\mu}_{\mathbf{x}}\}) = -\min_{\mathbf{w}\in\mathbb{R}^p} \mathcal{O}_{\text{CRF}}(\mathbf{w}) \quad (197)$$

and

$$\min_{\{\boldsymbol{\mu}_{\mathbf{x}}\}\in\mathbb{M}^n} \mathcal{Q}_{\text{M3N}}(\{\boldsymbol{\mu}_{\mathbf{x}}\}) = -\min_{\mathbf{w}\in\mathbb{R}^p} \mathcal{O}_{\text{M3N}}(\mathbf{w}), \quad (198)$$

and the optimal primal parameters $\hat{\mathbf{w}}$ can be recovered from the dual solution $\{\hat{\boldsymbol{\mu}}\}$ via the relation

$$\mathbf{w}(\{\boldsymbol{\mu}_{\mathbf{x}}\}) = C\sum_{(\mathbf{x},\mathbf{y})}[\mathbf{B}(\mathbf{x})]^{\mathsf{T}}[\boldsymbol{\phi}(\mathbf{y}) - \boldsymbol{\mu}_{\mathbf{x}}] \quad (199)$$

in both cases. We refrain from a detailed derivation here since it is inconsequential to our approach, but note in passing that for the CRF objective, the duality relation can easily be seen from the stationary conditions at the optimal $\hat{\mathbf{w}}$. More generally, the coupling is a direct consequence of Lagrangian duality.

Relaxing the problems

The main complications are the same as previously: first of all, the marginal polytope is intractable, since it is defined by an exponential number of halfspaces; moreover, the CRF objective requires to evaluate the entropy of the distribution coupled to the current marginal parameters, which, as we saw, is also intractable.

Our idea is now to proceed as previously, by relaxing the marginalization constraints such that the parameters are only required to reside in the

CRF Recipe 5 (Maximum Approximate Entropy Formulation)
Pick a concave region-based entropy approximation $\tilde{H}$ and solve the convex optimization problem

$$\begin{aligned} \underset{\{\boldsymbol{\tau}_{\mathbf{x}}\}}{\text{minimize}} \quad & \tilde{\mathcal{Q}}_{\text{CRF}}\big(\{\boldsymbol{\tau}_{\mathbf{x}}\}\big) \stackrel{\text{def}}{=} \frac{1}{2C}\big\|\textstyle\sum_{(\mathbf{x},\mathbf{y})}[\mathbf{B}(\mathbf{x})]^{\mathsf{T}}[\boldsymbol{\phi}(\mathbf{y}) - \boldsymbol{\tau}_{\mathbf{x}}]\big\|_2^2 - \sum_{(\mathbf{x},\mathbf{y})} \tilde{H}\big(\boldsymbol{\tau}_{\mathbf{x}}\big) \\ \text{s.t.} \quad & \boldsymbol{\tau}_{\mathbf{x}} \in \mathbb{L}(\mathbf{x}), \quad \forall(\mathbf{x},\mathbf{y}) \end{aligned}$$

using exponentiated gradient steps (see main text for details).

M3N Recipe 5 (Approximate Quadratic Program Formulation)
Pick a decomposing loss term $\mathbf{e}$ and solve the convex optimization problem

$$\begin{aligned} \underset{\{\boldsymbol{\tau}_{\mathbf{x}}\}}{\text{minimize}} \quad & \tilde{\mathcal{Q}}_{\text{M3N}}\big(\{\boldsymbol{\tau}_{\mathbf{x}}\}\big) \stackrel{\text{def}}{=} \frac{1}{2C}\big\|\textstyle\sum_{(\mathbf{x},\mathbf{y})}[\mathbf{B}(\mathbf{x})]^{\mathsf{T}}[\boldsymbol{\phi}(\mathbf{y}) - \boldsymbol{\tau}_{\mathbf{x}}]\big\|_2^2 - \sum_{(\mathbf{x},\mathbf{y})} \langle \boldsymbol{\tau}_{\mathbf{x}}, \mathbf{e}(\mathbf{y})\rangle \\ \text{s.t.} \quad & \boldsymbol{\tau}_{\mathbf{x}} \in \mathbb{L}(\mathbf{x}), \quad \forall(\mathbf{x},\mathbf{y}) \end{aligned}$$

using exponentiated gradient steps (see main text for details).

local polytope $\mathbb{L}(\mathbf{x})$, and by choosing a concave region-based entropy approximation $\tilde{H}(\boldsymbol{\tau}_{\mathbf{x}})$ instead of the intractable exact entropy. The resulting optimization problems are shown in CRF Recipe 5 and M3N Recipe 5.

One may wonder if—just like their exact counterparts—these relaxed problems are also dually coupled to a primal form. Indeed, this is the case: As expected, the coupling is to the relaxation of the exact primal-form CRF (CRF Recipe 1), or the exact primal-form M3N objective (M3N Recipe 2), respectively. This can easily be verified by deriving the Lagrangian dual of these problems similarly to the exact case.

Exponentiated gradient algorithm

One advantage of the above dual formulations over the "partially dualized" reparameterization formulations is that the M3N objective function is differentiable. A clear disadvantage is that the number of parameters is even larger in general: As we previously saw, for a choice of pieces as individual factors, the reparameterization parameters $\boldsymbol{\lambda}(F)$ need only be associated with *variable* states and hence require $O(|\mathcal{Y}_s|)$ memory. In contrast, each dual parameter $\boldsymbol{\mu}_{\mathbf{x}}$ consumes $O(|\mathcal{Y}_F|)$. For higher-order factors, this difference can be significant. Moreover, the local polytope is a significantly more difficult constraint set than the simple reparameterization equality constraints.

One may then wonder about the utility of the dual formulations. Their importance lies in the fact that they allow for application of a particularly convenient optimization scheme. Specifically, the local polytope constraints can be handled *implicitly* by taking exponentiated gradient steps. Exponentiated gradient training for CRFs and M3Ns was introduced by Collins et

[206] Michael Collins, Amir Globerson, Terry Koo, Xavier Carreras, and Peter L. Bartlett. Exponentiated Gradient Algorithms for Conditional Random Fields and Max-Margin Markov Networks. *Journal of Machine Learning Research*, 9:1775–1822, 2008

al.[206], but—to our knowledge—has so far only been applied to problems where *exact* inference is feasible. Our contribution here is to point out that it can be applied equally well to relaxations.

The main idea of the algorithm, applied to structured prediction problems, is as follows: Instead of the pseudo-marginal parameters $\boldsymbol{\tau}_{\mathbf{x}}$, we maintain dually coupled exponential parameters $\boldsymbol{\theta}_{\mathbf{x}}$ that *result* in these pseudo-marginals. Since the exponential parameters are *unconstrained*, one can thus avoid handling the local polytope constraints in the overall objective function. On the other hand, as we will see, it becomes necessary to compute the forward map $\boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}})$ from exponential parameters to pseudo-marginals for the exponentiated gradient steps, requiring repeated (tractable) inference as in our original relaxation formulation.

Nonetheless, the exponentiated gradient algorithm is attractive, since it can be implemented in an *online* manner. In particular, the exponentiated gradient steps can be chosen to update the parameters associated with a *single* example $(\mathbf{x}, \mathbf{y})$ at a time,

$$\boldsymbol{\theta}_{\mathbf{x}}^{(k+1)} = \boldsymbol{\theta}_{\mathbf{x}}^{(k)} - \eta \frac{\partial Q(\{\boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}})\})}{\partial \boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}})}. \tag{200}$$

[207] Léon Bottou and Olivier Bousquet. The Tradeoffs of Large Scale Learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2008

Unlike stochastic gradient methods,[207] which operate on the primal $\mathbf{w}$ parameters using an estimated gradient, this online update guarantees descent in the dual objective Q for a suitable chosen learning rate η. Indeed, the progress in the objective can be evaluated purely in terms of the parameters of the single chosen example, enabling efficient linesearch (which however, requires inference to obtain the marginals $\boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}})$ coupled to the current parameters).

To make the form of the updates concrete, for the CRF objective, assuming the *trivial* entropy approximation of Weiss et al.,[208] we develop

[208] Yair Weiss, Chen Yanover, and Talya Meltzer. MAP Estimation, Linear Programming and Belief Propagation with Convex Free Energies. In *Uncertainty in Artificial Intelligence (UAI)*, 2007

$$\frac{\partial Q_{\text{CRF}}(\{\boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}})\})}{\partial \boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}})} = \mathbf{1} + \log \boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}}) - \mathbf{B}(\mathbf{x})\mathbf{w}(\{\boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}})\}), \tag{201}$$

where the logarithm is taken component-wise, and similarly, for the M3N objective,

$$\frac{\partial Q_{\text{M3N}}(\{\boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}})\})}{\partial \boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}})} = -\mathbf{e}(\mathbf{y}) - \mathbf{B}(\mathbf{x})\mathbf{w}(\{\boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}})\}). \tag{202}$$

The primary challenge in implementing these updates efficiently lies in computation of the map $\mathbf{w}(\{\boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}})\})$, defined in (199). This term involves a sum over all training examples, so it would be inefficient to re-compute it for each update of the parameters of a single example. However, since (199) decomposes into contributions of the individual examples, one can simply maintain a vector $\mathbf{w}^{(k)}$ in memory and adjust it by the new contribution of an example after its parameters have been updated. This operation is efficient and requires only the marginals $\boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}}^{(k)})$ of the example for its previous parameters, and the marginals $\boldsymbol{\tau}(\boldsymbol{\theta}_{\mathbf{x}}^{(k+1)})$ for its new parameters.

The ability to update parameters in an online fashion is certainly an attractive property, even more so since decrease in the objective can be guaranteed. For this reason, CRF Recipe 5 and M3N Recipe 5 are worth considering, depending on the application.

A Brief Empirical Comparison

So far, we have seen a wide range of approximate convex formulations that are all equivalent for an appropriate choice of their hyper parameters. While we pointed out the theoretical advantages of each approach as far as memory consumption and computational aspects are concerned, it is interesting to see how these properties hold up in practice.

Towards this end, we will compare three different ways of solving the relaxation of the CRF objective using the *trivial* region-based concave entropy approximation over the first-order local polytope. In particular, we compare the following options for solving one and the same relaxation of the exact CRF objective:

- CRF Recipe 1 (denoted *Relaxation, BP*), where we solve the inner inference problem using norm-product belief propagation[209] at each step of the L-BFGS[210] iterative optimization code;
- CRF Recipe 4 (denoted *Relaxation, Jointly*), where the jointly convex formulation is optimized both over the model parameters and the tightening parameters at the same time using L-BFGS;
- And finally CRF Recipe 5 (denoted *Relaxation, EG*), where the dual of the relaxed CRF objective is solved by taking online exponentiated gradient steps,[211] again computing the map from exponential parameters to marginals using norm-product belief propagation.

Out of these formulations, CRF Recipe 4 is the one that does not require repeated inference using a specialized solver.

We compare these different ways of solving one and the same objective function to two other established tractable ways of CRF training:

- *Piecewise training*,[212] where the objective function is minimized using L-BFGS. Remember that the relaxation based on the trivial entropy approximation can be interpreted as an estimator that tightens the piecewise approach over all possible reparameterizations.
- *Pseudolikelihood training*,[213] again using L-BFGS to optimize the objective function. This estimator attains tractability by conditioning each variable on its Markov blanket and can be shown to asymptotically consistent.

We emphasize that in general, these two approaches attain optimal objective values that differ from the above three approaches. We include them in our comparison to put their relative computational cost into perspective.

The CoNLL-2000 task[214] we consider for our comparison will be described in greater detail in the next section. Our goal is to jointly predict the part-of-speech tags and the phrase boundaries of natural language sentences. Our model includes pairwise factors between variables encoding the part-of-speech and phrase-boundary labels. The cardinality of the variables is either 44 or 23, depending on which information it contains, and variables are connected in a cyclic graph structure, necessitating approximate parameter estimation.

All training algorithms were run on a 2,000 examples subset of the training data. This restriction was necessary due to the excessive memory con-

[209] Tamir Hazan and Amnon Shashua. Norm-product belief propagation: Primal-dual message-passing for approximate inference. *IEEE Transactions on Information Theory*, 56(12):6294–6316, 2010

[210] Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980

[211] Michael Collins, Amir Globerson, Terry Koo, Xavier Carreras, and Peter L. Bartlett. Exponentiated Gradient Algorithms for Conditional Random Fields and Max-Margin Markov Networks. *Journal of Machine Learning Research*, 9:1775–1822, 2008

[212] Charles Sutton and Andrew McCallum. Piecewise training for structured prediction. *Machine learning*, 77(2):165–194, 2009

[213] Julian Besag. Statistical Analysis of Non-Lattice Data. *The Statistician*, 24(3):179—-195, 1975

[214] Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 shared task. In *4th Conference on Computational natural language learning (CoNLL)*, September 2000

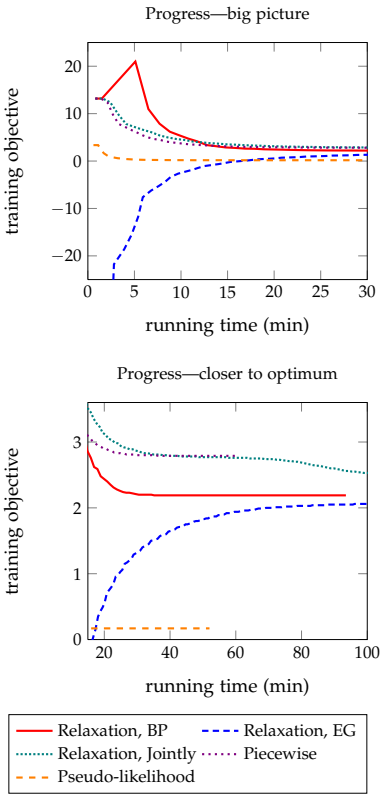


Figure 31: Objective (Y axis) vs. running time in minutes (X axis) on the CoNLL-2000 task.

sumption of the *Relaxation, EG* approach. Remember that the dual parameters require $O(|\mathcal{F}||\mathcal{Y}_F|)$ memory *per example*. In contrast, the *Relaxation, Jointly* approach only requires $O(|\mathcal{F}||\mathcal{Y}_s|)$ memory per example, and the *Relaxation, BP* approach does not require additional memory growing in the number of training examples. For large variable cardinalities in particular, the amount of training data that can be processed using the exponentiated gradient approach is clearly limited.

Results

The results are shown in Figure 31, where we plot for each approach the progress in the respective objective as a function of running time.

As one can see, the difference between the optimal piecewise objective value and the optimum of the relaxation is rather small, meaning that the piecewise approximation is rather tight already. The optimal objective value of the pseudolikelihood estimator is unrelated to the other approaches, so it is only the relative progress that is of interest to us.

Computationally, pseudolikelihood estimation is clearly the most efficient approach. From a bird's-eye view, all other approaches reach the vicinity of their respective optimum at comparable computational cost. In particular, all approaches that tighten the relaxation seem to be roughly comparable to piecewise training.

However, close to the optimum, the different approaches to solving the relaxation start differing. Interestingly, the *Relaxation, BP* approach, which repeatedly solves the inner inference problem using message passing, converges significantly faster than the *Relaxation, Jointly* approach, which solves a single jointly convex optimization problem. So while very attractive from a theoretical viewpoint, it seems that the large number of variables introduced into the jointly convex objective function inhibits rapid progress as we get closer to the optimum. The extra cost of solving an inference problem for each example at each outer step, as exercised by the *Relaxation, BP* approach, seems to pay off, as norm-product belief propagation can put the particular problem structure to good use. Although one should not rush to conclusions, this is a somewhat disappointing result—while we managed to eliminate the inference subproblem via convex reformulations, this does not seem to result in practical gains on this task: the jointly convex formulation requires more memory, and is slower to converge to the optimum.

The *Relaxation, EG* approach converges somewhat faster, but still slower than *Relaxation, BP*. This is to be expected, as it implements *online* parameter updates, affecting only the dual parameters of a single example at a time. However, in exchange, it should converge faster than batch methods initially, which is not clearly visible on this task. Besides, the ability of this approach to handle large, redundant datasets (the scenario which it is targeted at) is clearly limited due to its excessive memory requirements.

Computationally, out of the approaches that seek to solve the *relaxation*, the *Relaxation, BP* approach is the clear winner in this brief comparison. However, this finding can very dramatically depending on the properties of the task, so it is still important to be able to choose from a variety of alternatives, to select the most suitable approach for the problem at hand.

Applications and Results

Perhaps even more important than computational efficiency is the prediction accuracy achieved by different approaches to parameter estimation. As far as this aspect is concerned, all computational approaches are equivalent as long as they solve the same objective function. We are going to compare three different objective functions in the following:

- *Relaxation*, by which we denote the relaxation of the exact CRF objective obtained by optimizing over the local polytope instead of the marginal polytope, and the trivial concave region-based entropy approximation of Weiss et al.[215] instead of the true entropy;
- *Piecewise*, denoting the piecewise training approach of Sutton and McCallum[216], which can be seen as an untightened variety of the above;
- *Pseudolikelihood*, by which we denote the asymptotically consistent pseudo likelihood estimator of Besag.[217]

Note that these are the same objective functions we considered in the previous section; but this time, we are interested in their *generalization performance*, rather than computational efficiency. Towards this end, we consider several applications from different fields of research. Since all of the tasks involve cyclic graphs, we use the MPLP algorithm[218] for MAP prediction on test data, after the model parameters have been estimated. MPLP seeks to solve the first-order LP relaxation of the MAP problem; alternatively, we could have used the popular TRW-S code[219] or our own IncMP algorithm.

[215] Yair Weiss, Chen Yanover, and Talya Meltzer. MAP Estimation, Linear Programming and Belief Propagation with Convex Free Energies. In *Uncertainty in Artificial Intelligence (UAI)*, 2007

[216] Charles Sutton and Andrew McCallum. Piecewise training for structured prediction. *Machine learning*, 77(2):165–194, 2009

[217] Julian Besag. Statistical Analysis of Non-Lattice Data. *The Statistician*, 24(3):179—-195, 1975

[218] Amir Globerson and Tommi Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *Advances in Neural Information Processing Systems*, 2007

[219] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568 – 1583, 2006

Joint Part-of-Speech Tagging and Chunking

The CoNLL-2000 shared task[220] provides sentences that are labelled with Part-of-Speech and phrase boundary information. Sutton and McCallum[221] suggested to model the problem as a factorial CRF involving two label chains, see Figure 32 for an illustration. The dataset comprises a total of 10,947 sentences, which we split into 2000 test examples and 8,947 training examples. The label space consists of 44 PoS tags and 23 tags that mark phrase boundaries. The features of our CRF are based on the tokens in a window around the current position in the sentence. We use two unary factor types, one for each chain, as well as two pairwise factor types to model in-chain and between-chain interactions. The weights associated with the features are tied for factors of the same type; all factor types employ the same sparse feature vectors. These local observations are highly indicative of the true unary and pairwise states.

[220] Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 shared task. In *4th Conference on Computational natural language learning (CoNLL)*, September 2000

[221] Charles Sutton and Andrew McCallum. Piecewise training for structured prediction. *Machine learning*, 77(2):165–194, 2009

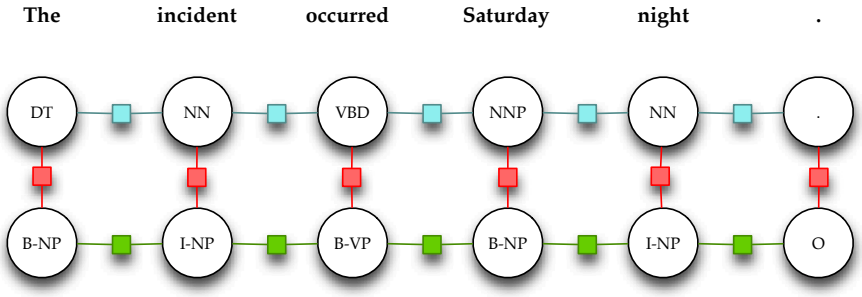


Figure 32: The CoNLL-2000 shared task dataset: Joint Part-Of-Speech tagging and phrase chunking.

This is precisely the scenario in which the piecewise estimator can be expected to perform well. The results of our experiment are shown in Figure 33. For each of the considered estimators, we plotted the per-label prediction accuracy on the test data and the training data. We trained using two different settings of the regularization parameter: $C = 10^{-4}$, which corresponds to strong regularization, and $C = 10^{-8}$, which is very moderate.[222] For each system, we plot the average final duality gap of MPLP (a gap of 0% indicates that the solution found by MPLP is indeed the discrete optimum), as well as the average number of iterations taken by MPLP. These numbers quantify the difficulty of the inference problem resulting from the estimators.

[222] In our implementation, we divide the loss by the number of variables, so the value of C we report here is scaled accordingly.

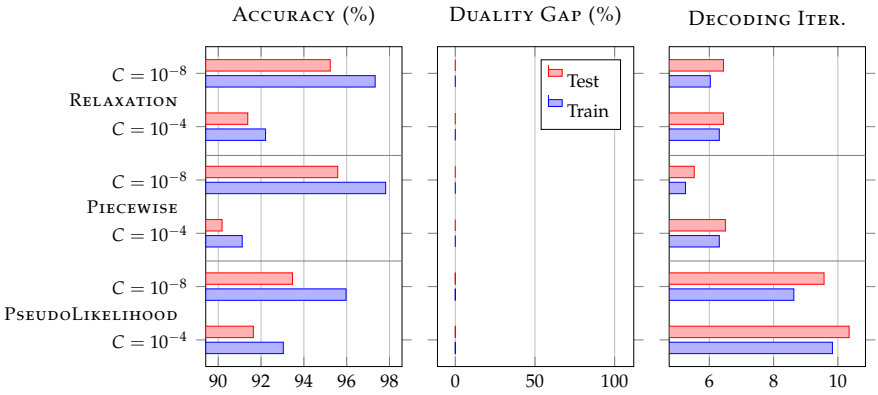


Figure 33: Results on the CoNLL-2000 dataset.

The piecewise estimator and its tightened variant, our relaxation, perform best at weak regularization settings. Both of these estimators outperform the pseudolikelihood estimator substantially. However, in this task, as was to be expected, there is no gain in using a tighter-than-piecewise relaxation since the local evidence is very strong. Approximate MAP prediction using MPLP works well (zero gap) for all systems, although the pseudolikelihood approximation results in a few more iterations of MPLP. Our results are slightly better than those reported by Sutton and McCallum[223], however we were able to use more training data. Finally, we also determined the *exact* ML estimator at $C = 10^{-4}$ by computing the marginals via junction trees, resulting in a test accuracy of 93.84%. This is somewhat better than the piecewise approximations at $C = 10^{-4}$, which however tend to require weaker regularization, so the result is only mildly conclusive.

[223] Charles Sutton and Andrew McCallum. Piecewise training for structured prediction. *Machine learning*, 77(2):165–194, 2009

Figure 34: The "Horse segmentation" task: The estimator drawing on the relaxation is more capable of reproducing characteristics that require global propagation of belief.

Segmentation of Horse Images

We now turn to a segmentation task from computer vision where propagation of belief is much more important, specifically the Weizmann Horse dataset used by Borenstein et al.[224] Here, the goal is to determine those pixels of an image that are part of a horse, as illustrated in Figure 34. The dataset consists of 328 images of horses, which we randomly split into 200 training images and 128 test images. Only two labels are required to model this task. We use a simple 4-connected CRF model involving three factor types—one for the unaries, and one each for the horizontal and vertical factors.

[224] Eran Borenstein, Eitan Sharon, and Shimon Ullmann. Combining top-down and bottom-up segmentation. In *IEEE Workshop on Perceptual Organization in Computer Vision*, June 2004

Unlike the previous task, it is very challenging to find good local observations that are highly indicative of the label of a pixel. Our approach to feature extraction is inspired by Tappen et al.[225] and works as follows: In a first step, we extract from the training data roughly a thousand 50×50 image patches from locations close to the segmentation boundary and store their corresponding segmentation masks. The patches are chosen at random in a greedy manner that ensures the patches are sufficiently dissimilar. Each such patch contains e.g. the head of a horse, and the segmentation mask specifies which pixels are part of the animal.

[225] Marshall Tappen, Kegan Samuel, Craig Dean, and David Lyle. The Logistic Random Field—A convenient graphical model for learning parameters for MRF-based labeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2008

In the second step, for each patch and each image, we determine the position where the absolute value of the normalized cross-correlation (NCC) is maximized. At this position, we overlay the segmentation mask, with values encoded as $\{+1, -1\}$, multiplied by the absolute NCC. Hence, for each pixel of an image, we obtain a sparse list of indices of the masks that were overlaid at this pixel, along with the corresponding scaled values. This step must also be performed for the test images.

We use the sparse vector obtained in the second step as the features of our unary factor type, associating with each component a weight. As a result, the system can learn which image patches are most informative by adjusting these weights. The patches typically cover most parts of the horse and so give a reasonable local hint. The pairwise factor types, on the other hand, are used to propagate connectivity, and to push the boundaries of the predicted segmentation towards meaningful positions in the image. Towards this end, we employ a constant bias term as well as a single feature that measures the absolute difference in intensity of the two pixels covered by a pairwise term.

Figure 35 shows the results obtained from the different system configurations. Again, we note that approximate MAP prediction via MPLP works surprisingly well – for all systems, the duality gap is negligible. However, there are substantial differences regarding the number of iterations

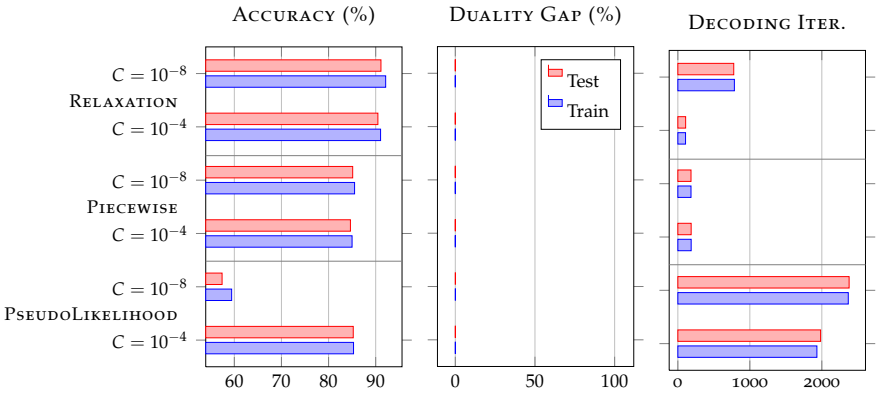


Figure 35: Results on the "Horse" dataset.

required by MPLP. In particular, the pseudolikelihood estimator gives rise to very hard inference problems at both settings of C. For the horse images, which are typically no larger than 300×200 pixels, this may not be much of a concern. However, for larger images, an order of magnitude difference in the number of iterations can effectively render approximate MAP decoding intractable.

Regarding the prediction accuracy of the competing systems, our tighter-than-piecewise relaxation is the clear winner at both settings of C. At over 91.1% test accuracy, it surpasses the other estimators by 5 percent points. This difference is also clearly visible in Figure 34. A worrisome result is that the pseudolikelihood estimator fails completely at $C = 10^{-8}$. It is unclear to us why this is the case. Compared to the 94.6% test accuracy achieved by Tappen et al.[226], our peak result of 91.1% still lags somewhat behind, however, we emphasize that our numbers were obtained on black & white input, while Tappen et al. seem to work on color input, which must be expected to be easier. The rest of the difference can be attributed to our rather simple feature engineering.

[226] Marshall Tappen, Kegan Samuel, Craig Dean, and David Lyle. The Logistic Random Field—A convenient graphical model for learning parameters for MRF-based labeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2008

Grapheme-to-Phoneme Prediction

The final task we consider is characterized by large label spaces and sparse higher-order factors. The goal is to predict the pronunciation of a German word given its orthographic input string, i.e. to learn a map from graphemes to phonemes. We also want to model stress and glottal stops jointly with the sequence of phonemes. The dataset we use has not been considered in the literature before; it consists of 8,000 training examples and 2,000 test examples.

The factor graph model we devised for this task can be seen in Figure 36. We use three chains, one for the word stress, one for the sequence of phonemes, and one for glottal stops. The label space of the phoneme variables is very large, encoding 51 phoneme symbols. The variables encoding word stress and glottal stops, on the other hand, are binary. We use dense pairwise factors within each chain. We use a binary feature vector consisting of observed grapheme n-grams relative to the current position for these pairwise factor types. To enforce the constraint that only one syllable can be stressed, we use a sparse global factor that prohibits all invalid

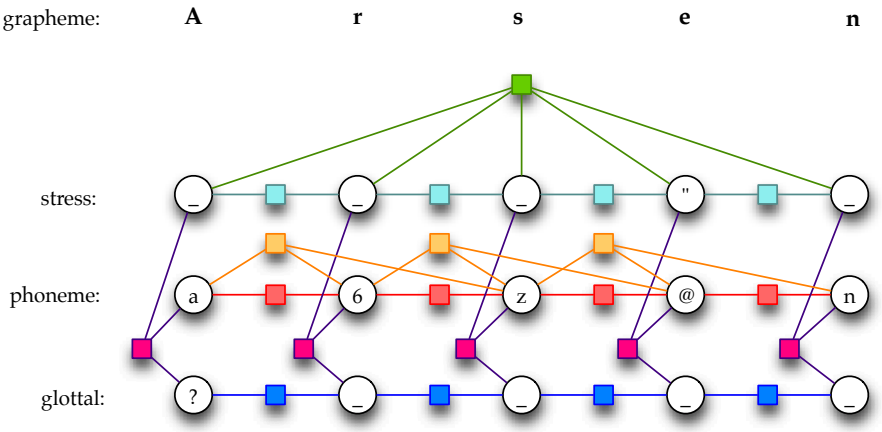


Figure 36: An exemplary factor graph in the grapheme-to-phoneme transcription task: Higher-order factors over the phonemes are not shown to reduce clutter.

states (see Figure 36, top factor). This factor is only instantiated during prediction at test time—as shown by Roth and Yi,[227] it is preferable not to incorporate hard constraints during training.

To model interactions between the chains, we use sparse ternary factors. Moreover, the higher-order interactions between phoneme variables are modelled using sparse factors involving up to six variables. For these sparse factor types, the only features we use are constant bias terms. A joint state of such a factor must occur in the training data at least once in order to receive a separate weight; all other states are merged into a single default state. As shown by Cohn,[228] most common factor operations can be performed efficiently in sparse factors with tied potentials.

[227] Dan Roth and Wen-tau Yih. Integer linear programming inference for conditional random fields. In *22nd International conference on Machine learning (ICML)*, pages 736–743, August 2005

[228] Trevor Cohn. Efficient Inference in Large Conditional Random Fields. In *17th European Conference on Machine Learning (ECML)*, pages 606–613, 2006

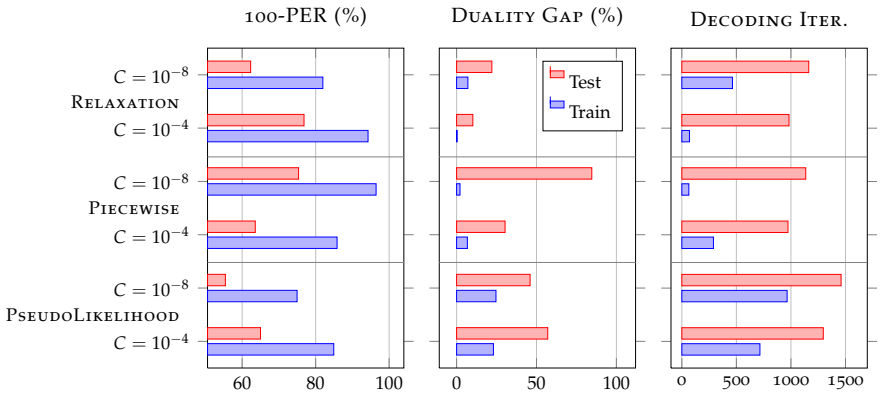


Figure 37: Results of the Grapheme-to-Phoneme task.

Since the grapheme and phoneme sequence are generally of different length, our approach requires a pre-processing step: We first merge some phoneme symbols based on mutual information statistics until all phoneme sequences are shorter than the corresponding grapheme sequences or of equal length. The two sequences are then aligned via dynamic programming, using mutual information between graphemes and phonemes as a distance measure. The task can then be formulated as a graph labelling problem and we proceed as usually.

Since the model is very complex, we expect the first-order local polytope to be very loose, and approximate inference to fail in many cases. We do not advocate the use of this model to achieve the best practical results; rather, the goal is to highlight the difference between the considered estimators.

Figure 37 shows the results. For each system, we report the accuracy in terms of the phoneme-error-rate (PER), defined as the string-edit-distance between the predicted phoneme sequence and the correct phoneme sequence, normalized by the total length of the correct phoneme sequence.

Figure 37 confirms that there are two sources of error in this task. Unlike the previous two tasks, approximate MAP prediction using MPLP does not work well, i.e. the approximate solutions are far from being optimal. We not only incur errors through the approximate estimators, but also through approximate prediction. The interplay of the two components becomes important. This can lead to seemingly paradox situations: For instance, the average duality gap during prediction is very large when training with the piecewise approximation at $C = 10^{-8}$; still, the overall system compares very favourably to the others. Interestingly, our tighter-than-piecewise relaxation performs best at $C = 10^{-4}$, while the piecewise estimator is better at $C = 10^{-8}$. Overall, use of our relaxation leads to the smallest duality gaps, and at $C = 10^{-4}$, it also achieves the highest test accuracy of all configurations. The pseudolikelihood estimator fails miserably again.

Conclusions and Future Work

In this part of the thesis, we contributed:

a) Convergent message passing algorithms for inference in discrete graphical models, solving well-motivated convex relaxations that provide an upper bound on the exact objective function that is solved during variational inference;

b) Based on these relaxations, a whole catalog of recipes for tractable training of CRFs and M3Ns, drawing on several convex reformulations, some of which remove the need for repeated inference during training;

c) A brief empirical comparison of our message passing algorithms to several competitors, as well as an empirical comparison of several tractable approaches to CRF training, both in terms of computational efficiency and predictive accuracy of the resulting models.

Compared to approaches such as loopy belief propagation and naïve mean field inference, as well as discriminative training approaches building on these, all approaches we presented in our thesis are well-motivated in the sense that they solve a tractable, convex objective function that forms an upper bound on the optimal value of the exact problem.

Compared to piecewise training,[229] which can also be considered a valid *relaxation* of the CRF training problem, the approaches we discussed can be considerably tighter (depending on the task), and often come at comparable computational cost. Compared to pseudolikelihood estimation,[230] training based on relaxations has proved to be significantly more robust, in the sense that models estimated for maximum pseudolikelihood tended to expose pathological predictive performance in some situations.

In the future, it would be gainful to conduct a large-scale study evaluating the different approaches and competing estimators in an exhaustive series of experiments to be able to draw even stronger conclusions.

[229] Charles Sutton and Andrew McCallum. Piecewise training for structured prediction. *Machine learning*, 77(2):165–194, 2009

[230] Julian Besag. Statistical Analysis of Non-Lattice Data. *The Statistician*, 24(3):179—195, 1975

Part III

Tractability through Gaussian Approximations

Exact Inference in Gaussian Models

In the second part of this thesis, we saw that exact inference in discrete graphical models is intractable, except for a few special cases (most prominently trees). Consequently, we discussed a variety of tractable relaxations as substitutes for the exact variational inference problems, and introduced algorithms to solve and make use of these relaxations.

A different approach, and indeed the one we are going to follow in this final part of the thesis, is to postulate a Gaussian model even when this assumption is unwarranted. Inference in Gaussian models is tractable (at least in theory), so one can work exactly within this restricted class of models. This opens up a variety of discriminative training approaches, allowing to handle both continuous and discrete structured prediction tasks.

In this first chapter of the final part of the thesis, we will start by recapitulating multivariate Gaussians and Gaussian Markov random fields. In doing so, we will make clear our notion of *inference*. While it is true that this problem has an exact algebraic solution, it is computationally more convenient to solve the inference problem using iterative algorithms when working in extremely high-dimensional Gaussian models. These algorithms allow to exploit the structure of the inverse covariance matrix, which, as we already saw at the beginning of this thesis, is defined by the underlying graphical model. Consequently, we will discuss a few of these approaches in some detail, as they are the computational backbone of the structured prediction methods we will devise in the chapters to follow.

The Normal Distribution and Gaussian Graphical Models

In this part of the thesis, we will model our structured prediction problem via v random variables $\mathbf{y}_s \in \mathbb{R}^\kappa, s \in V$ that are jointly Gaussian, i.e.

$$\mathbf{y} = (\mathbf{y}_1 \dots \mathbf{y}_v)^\mathsf{T} \tag{203}$$

with

$$\mathbf{y} \sim \mathcal{N}(\mathbf{u}, \mathbf{C}), \quad \mathbf{C} \succ 0. \tag{204}$$

Notably, when we speak of a variable, we actually refer to a *block* of κ components, so $\mathcal{Y}_s = \mathbb{R}^\kappa$ and $\mathcal{Y} = \mathbb{R}^{v\kappa}$. Put another way, we are interested in predicting v values, each of which is κ-dimensional.

Since the variables are jointly normal, one might equally think of this process as predicting $v\kappa$ scalar values. However, in the applications we will consider, the smallest units are naturally defined as low-dimensional vectors of fixed size κ. An additional benefit is that one can exploit this particular block structure computationally.

In the following, we are going to recapitulate important facts about Gaussian distributions and discuss some aspects in greater depth than in the short introduction at the beginning of the thesis.

Characterization of the Density

Typically, the density of multivariate normal distributions is parameterized in *standard form*. The density of the $v\kappa$-dimensional normal distribution $\mathcal{N}(\mathbf{u}, \mathbf{C})$ with mean $\mathbf{u}$ and covariance $\mathbf{C}$ in standard form is then given by

$$p(\mathbf{y}; \mathbf{u}, \mathbf{C}) = (2\pi)^{-\frac{v\kappa}{2}} \det(\mathbf{C})^{-\frac{1}{2}} \exp(-\tfrac{1}{2}(\mathbf{y}-\mathbf{u})^\mathsf{T}\mathbf{C}^{-1}(\mathbf{y}-\mathbf{u})), \quad \mathbf{C} \succ 0. \tag{205}$$

The covariance $\mathbf{C}$ must be positive-definite for the density to be valid. For later use, we record the entropy of this density, given in closed form by

$$H(\mathbf{u}, \mathbf{C}) = \tfrac{1}{2}\log\det(\mathbf{C}) + \tfrac{v\kappa}{2}\log 2\pi e. \tag{206}$$

We will prefer to work with an alternative parameterization of the same Gaussian density because it better reflects the *factorization* of the density. The so-called *canonical* or *information* form,[231] denoted $\mathcal{C}(\mathbf{h}, \mathbf{J})$, is defined as

[231] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009

$$p(\mathbf{y}; \mathbf{h}, \mathbf{J}) = \exp(-\tfrac{1}{2}\mathbf{y}^\mathsf{T}\mathbf{J}\mathbf{y} + \mathbf{y}^\mathsf{T}\mathbf{h} - A(\mathbf{h}, \mathbf{J})), \quad \mathbf{J} \succ 0. \tag{207}$$

As we already intimated in the beginning of the thesis, the canonical form maps to standard form using the equalities

$$\mathbf{J} = \mathbf{C}^{-1}, \qquad \mathbf{C} = \mathbf{J}^{-1}, \tag{208}$$

$$\mathbf{h} = \mathbf{C}^{-1}\mathbf{u}, \qquad \mathbf{u} = \mathbf{J}^{-1}\mathbf{h}, \tag{209}$$

and the normalization constant A can be computed as

$$A(\mathbf{h}, \mathbf{J}) = \tfrac{1}{2}\mathbf{h}^\mathsf{T}\mathbf{J}^{-1}\mathbf{h} + \tfrac{v\kappa}{2}\log(2\pi) + \tfrac{1}{2}\log\det(\mathbf{J}^{-1}). \tag{210}$$

It is worth pointing out that A is precisely the log-partition function we studied in great detail already; the fact that it can be computed efficiently using linear algebra primitives requiring time at most polynomial in the number of variables demonstrates the tractability of *inference* in Gaussian models. We will make this point more concrete shortly.

Gaussian Graphical Models

We already alluded to the fact that the canonical form of the Gaussian density directly reflects the factorization of the distribution, which we will again describe by means of a graph G with vertex set V and edge set E. In particular, this connection is evident from the symmetric, positive-definite $v\kappa \times v\kappa$ matrix $\mathbf{J} \succ 0$. This matrix, the inverse of covariance matrix $\mathbf{C}$, goes by the names *precision*, *information*, or *concentration* matrix. Notably, the $\kappa \times \kappa$ block corresponding to a pair (s, t) of variables is required to be zero unless $(s, t) \in E$.

By the Hammersley-Clifford theorem, $\mathbf{y} \sim p(\mathbf{y}; \mathbf{h}, \mathbf{J})$ is then Markov with respect to graph G, that is, any two non-adjacent variables are conditionally independent given all other variables, formally

$$\mathbf{y}_s \perp\!\!\!\perp \mathbf{y}_t \mid \mathbf{y}_{V\setminus\{s,t\}}, \quad \forall s, t \in V. \tag{211}$$

Importantly, this allows us to factor the distribution in terms of fully connected subgraphs $F \in \mathcal{F}$ of G via

$$p(\mathbf{y}) = \frac{1}{Z} \prod_{F \in \mathcal{F}} \psi_F(\mathbf{y}_F), \tag{212}$$

where $\psi_F(\mathbf{y}_F) \propto \mathcal{C}(\mathbf{y}_F; \mathbf{h}_F, \mathbf{J}_F)$ with the $\mathbf{h}_F$ and $\mathbf{J}_F$ parameters of the individual factors chosen so as to add up to $\mathbf{h}$ and $\mathbf{J}$, respectively.

Conversely, any factorization of the above kind fully specifies the parameters of a joint density $\mathcal{C}(\mathbf{y}; \mathbf{h}, \mathbf{J})$. Though they contain some redundancy, we will find it convenient to model $p(\mathbf{y})$ directly in terms of the parameters $\mathbf{h}_F$ and $\mathbf{J}_F$ associated with the factors. It can easily be seen that due to the restricted quadratic form of the model, any global $\mathbf{h}$ and $\mathbf{J}$ can be realized using at most pairwise factors.

Operations on canonical forms

Let us now consider a few important operations that can be carried out using factors in canonical form.[232] Towards this end, we need a notation for referring to sub-blocks of a pairwise factor $F = (s, t)$. Note that the canonical parameters of such a factor are given by $\kappa \times 1$ blocks $[\mathbf{h}]_s$ and $\kappa \times \kappa$ blocks $[\mathbf{J}]_{st}$ as follows:

$$\mathbf{h}_F = \begin{pmatrix} [\mathbf{h}_F]_s \\ [\mathbf{h}_F]_t \end{pmatrix} \quad \text{and} \quad \mathbf{J}_F = \begin{pmatrix} [\mathbf{J}_F]_{ss} & [\mathbf{J}_F]_{st} \\ [\mathbf{J}_F]_{ts} & [\mathbf{J}_F]_{tt} \end{pmatrix}. \tag{213}$$

We will use the above bracket notation to refer to the appropriate block in the following. Similarly, we use $[\mathbf{h}]_s$ to refer to a $\kappa \times 1$ block of the global canonical parameter $\mathbf{h}$ corresponding to variable $\mathbf{y}_s$, and $[\mathbf{J}]_{st}$ to denote the $\kappa \times \kappa$ block of the global parameter matrix $\mathbf{J}$ composed of rows belonging to variable $\mathbf{y}_s$ and columns belonging to variable $\mathbf{y}_t$.

[232] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009

Multiplication. Perhaps the most common operation is to multiply factors. This simply involves adding up the canonical parameters of the factors. For instance, from the factorization

$$p(\mathbf{y}) = \frac{1}{Z} \prod_F \psi_F(\mathbf{y}_F), \quad \psi_F(\mathbf{y}_F) \propto \mathcal{C}(\mathbf{y}_F; \mathbf{h}_F, \mathbf{J}_F), \tag{214}$$

it follows that $\mathbf{y} \sim \mathcal{C}(\mathbf{h}, \mathbf{J})$ with

$$[\mathbf{h}]_s = \sum_{F: s \in F} [\mathbf{h}_F]_s, \qquad \forall s \in V, \tag{215}$$

$$[\mathbf{J}]_{ss} = \sum_{F: s \in F} [\mathbf{J}_F]_{ss}, \qquad \forall s \in V, \tag{216}$$

$$[\mathbf{J}]_{st} = \sum_{F: s,t \in F} [\mathbf{J}_F]_{st}, \qquad \forall (s, t \neq s) \in E. \tag{217}$$

Marginalization. For $F = (s, t)$, assume that a factor $\psi_F(\mathbf{y}_F) \propto \mathcal{C}(\mathbf{y}_F; \mathbf{h}_F, \mathbf{J}_F)$ is marginalized for variable s. One can show that $\mathbf{y}_s \sim \mathcal{C}(\mathbf{h}_s, \mathbf{J}_s)$ with

$$\mathbf{h}_s = [\mathbf{h}_F]_s - [\mathbf{J}_F]_{st} [\mathbf{J}_F]_{tt}^{-1} [\mathbf{h}_F]_t, \tag{218}$$

$$\mathbf{J}_s = [\mathbf{J}_F]_{ss} - [\mathbf{J}_F]_{st} [\mathbf{J}_F]_{tt}^{-1} [\mathbf{J}_F]_{ts}. \tag{219}$$

The latter equality can be seen to be the Schur complement of block $[\mathbf{J}_F]_{tt}$ of matrix $\mathbf{J}_F$ and has important applications in MMSE estimation.[233]

[233] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory*. Prentice Hall, 1999

Conditioning. Finally, if $\psi_F(\mathbf{y}_F) \propto \mathcal{C}(\mathbf{y}_F; \mathbf{h}_F, \mathbf{J}_F)$ with $F = (s,t)$ is conditioned on $\mathbf{Y}_t = \mathbf{y}_t$, the resulting density follows $\mathbf{y}_s \mid \mathbf{y}_t \sim \mathcal{C}(\mathbf{h}_{s|t}, \mathbf{J}_{s|t})$ with

$$\mathbf{h}_{s|t} = [\mathbf{h}_F]_s - [\mathbf{J}_F]_{st}\mathbf{y}_t, \tag{220}$$

$$\mathbf{J}_{s|t} = [\mathbf{J}_F]_{ss}. \tag{221}$$

These building blocks allow to construct a wide range of operations in inference and learning algorithms and will be useful in the sequence.

The Variational Viewpoint

Given a Gaussian density in canonical parameterization, perhaps the most important task is to be able to compute the associated *mean*. Unlike Gaussian densities in standard form, this information is not readily available. Moreover, one may be interested in the *covariance*. As we saw, there is in fact a mapping from canonical parameters to standard parameters that allows us to compute these quantities. This process is analogous to *inference* in discrete graphical models, except that the operation is tractable. Moreover, in a Gaussian model, the mean equals the one and only mode, so the MAP and marginalization problems coincide.

In order to develop an optimization perspective of the above problem, it is again useful to consider the variational viewpoint. In the beginning of the thesis, we already mentioned that Gaussians in canonical parameterization map into the exponential family framework via

$$\boldsymbol{\phi}(\mathbf{y}) = \begin{pmatrix} \mathbf{y} \\ \operatorname{vec}(\mathbf{y}\mathbf{y}^\mathsf{T}) \end{pmatrix} \quad \text{and} \quad \theta = \begin{pmatrix} \mathbf{h} \\ -\frac{1}{2}\operatorname{vec}(\mathbf{J}) \end{pmatrix}. \tag{222}$$

Consequently, the corresponding mean parameters are

$$\boldsymbol{\mu} = \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y};\theta)}[\boldsymbol{\phi}(\mathbf{y})] = \begin{pmatrix} \mathbf{u} \stackrel{\text{def}}{=} \mathbb{E}[\mathbf{y}] \\ \boldsymbol{\Sigma} \stackrel{\text{def}}{=} \mathbb{E}[\mathbf{y}\mathbf{y}^\mathsf{T}] \end{pmatrix}. \tag{223}$$

By definition of the second-order moment matrix $\boldsymbol{\Sigma}$, we have $\boldsymbol{\Sigma} - \mathbf{u}\mathbf{u}^\mathsf{T} = \mathbf{C}$, so the mean parameters must belong to the set

$$\mathcal{M}(G) = \{\mathbf{u}, \boldsymbol{\Sigma} \mid \boldsymbol{\Sigma} - \mathbf{u}\mathbf{u}^\mathsf{T} \succ 0\} \tag{224}$$

such that the covariance matrix $\mathbf{C}$ is positive-definite.

We also saw that in exponential families, the variational inference problem in general exposes the form[234]

$$A(\theta) = \max_{\theta \in \mathcal{M}^\circ(G)} \left\{ \theta^\mathsf{T}\boldsymbol{\mu} + H(p_{\theta(\boldsymbol{\mu})}) \right\}. \tag{225}$$

From the closed-form expression for the Gaussian entropy in (206), as well as the above characterization of the exponential and mean parameters, we thus conclude that—for the special case of Gaussians—this variational problem attains the form

$$\begin{aligned} A(\mathbf{h}, \mathbf{J}) = \max_{\mathbf{u}, \boldsymbol{\Sigma}} \Big\{ -\tfrac{1}{2}\operatorname{tr}(\mathbf{J}^\mathsf{T}\boldsymbol{\Sigma}) + \mathbf{h}^\mathsf{T}\mathbf{u} + \tfrac{1}{2}\log\det(\boldsymbol{\Sigma} - \mathbf{u}\mathbf{u}^\mathsf{T}) + \tfrac{\nu\kappa}{2}\log 2\pi e \Big\} \\ \text{s.t. } \boldsymbol{\Sigma} - \mathbf{u}\mathbf{u}^\mathsf{T} \succ 0. \end{aligned} \tag{226}$$

[234] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008

By our previous discussion of the relation between parameters in standard and canonical form, the maximum of this convex optimization problem must be attained uniquely at

$$\hat{\boldsymbol{\mu}} = \mathbf{J}^{-1}\mathbf{h}, \tag{227}$$

$$\hat{\boldsymbol{\Sigma}} = \mathbf{C} + \mathbf{u}\mathbf{u}^{\mathsf{T}} = \mathbf{J}^{-1} + \mathbf{u}\mathbf{u}^{\mathsf{T}}. \tag{228}$$

This result provides us with an explicit characterization of the mean parameters, which, as we already saw, are required for conditional random field training.

Algorithms for Inference

We already alluded to the fact that the mode of a Gaussian density is equal to its mean. Consequently, computation of the mean can be understood as *energy minimization* in a Gaussian model.

Note that the energy exposes a particularly simple quadratic form,

$$E(\mathbf{y}; \mathbf{h}, \mathbf{J}) = \frac{1}{2}\mathbf{y}^{\mathsf{T}}\mathbf{J}\mathbf{y} - \mathbf{y}^{\mathsf{T}}\mathbf{h}, \tag{229}$$

and the solution is hence given in closed-form by

$$\hat{\mathbf{y}} = \mathbf{u} = \arg\min_{\mathbf{y}} E(\mathbf{y}; \mathbf{h}, \mathbf{J}) = \mathbf{J}^{-1}\mathbf{h}. \tag{230}$$

Another interpretation of inference in a Gaussian model is as determining the solution to a system of linear equations

$$\mathbf{J}\mathbf{u} = \mathbf{h}. \tag{231}$$

In principle, direct methods are applicable to this problem and can solve it in polynomial time.[235] However, for our purposes, the excessive size of $\mathbf{J}$—depending on the application it is not uncommon for it to be a $10^6 \times 10^6$ matrix—render such approaches unattractive. Moreover, since $\mathbf{J}$ is typically extremely sparse in our applications, it is desirable to exploit this sparsity as efficiently as possible. A variety of iterative methods exist that are useful towards this end; we will introduce three different approaches in the following.

[235] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996

The Conjugate Gradient Method

Perhaps the most commonly employed algorithm for iteratively solving systems of linear equations is the *conjugate gradient* (CG) method.[236] This algorithm is outlined (specialized for our setting) in Figure 38.

The key step is computation of the sparse matrix-vector product $\mathbf{J}\mathbf{p}$. This operation can be carried out directly in terms of the per-factor contributions $\mathbf{J}_F$, such that $\mathbf{J}$ need not actually be instantiated. Moreover, computation of the product can be parallelized efficiently over the variables $s \in V$ via

$$[\mathbf{J}\mathbf{p}]_s = \sum_{F : s \in F} [\mathbf{J}_F]_{ss}[\mathbf{p}]_s \sum_{t \in F \setminus s} [\mathbf{J}_F]_{st}[\mathbf{p}]_t. \tag{232}$$

Note that each $\kappa \times 1$ block $[\mathbf{J}\mathbf{p}]_s, s \in V$ can be computed independently of the other blocks and is determined by a number of products of dense $\kappa \times \kappa$ and $\kappa \times 1$ blocks. This allows for additional instruction-level parallelism.

[236] Magnus R. Hestenes and Eduard Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, 49(6), 1952

Figure 38: ConjugateGradient algorithm

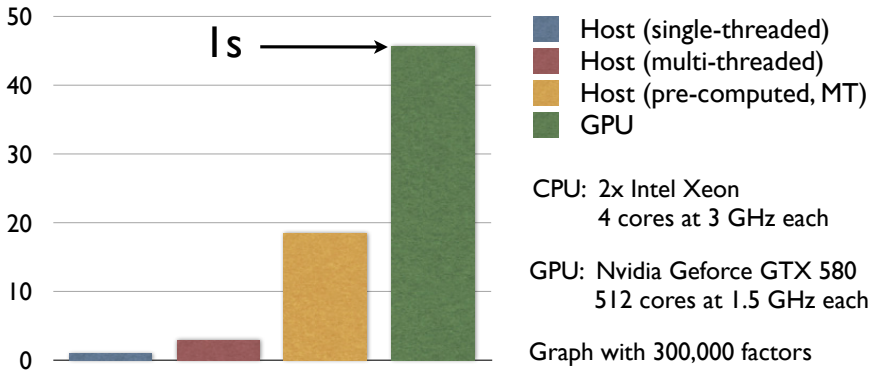
Initialize mean $\mathbf{u} \leftarrow \mathbf{0}$;
Determine residual $\mathbf{r} \leftarrow \mathbf{h} - \mathbf{J}\mathbf{u}$;
Set initial direction $\mathbf{p} \leftarrow \mathbf{r}$;
Compute squared residual norm $r' \leftarrow \mathbf{r}^\mathsf{T}\mathbf{r}$;
repeat
  Determine step size $\alpha \leftarrow \frac{r'}{\mathbf{p}^\mathsf{T}\mathbf{J}\mathbf{p}}$;
  Update mean $\mathbf{u} \leftarrow \mathbf{u} + \alpha\mathbf{p}$;
  Recompute residual $\mathbf{r} \leftarrow \mathbf{r} - \alpha\mathbf{J}\mathbf{p}$;
  Recompute squared residual norm $r \leftarrow \mathbf{r}^\mathsf{T}\mathbf{r}$;
  Compute new direction $\mathbf{p} \leftarrow \mathbf{r} + \frac{r}{r'}\mathbf{p}$;
  Record previous residual norm $r' \leftarrow r$;
until $r < \varepsilon$;
Return mean $\mathbf{u}$;

Moreover, the conjugate gradient method is well-suited for execution on Graphics processing units (GPUs). A GPU implementation is available for instance in the thrust[237] library.

[237] http://developer.nvidia.com/thrust

It is worth pointing out that a highly tuned implementation can be orders of magnitude faster than a naïve approach. Figure 39 shows the relative speed-up over several stages of refinement of our actual implementation, on a Gaussian graphical model resulting from an image inpainting task.

Figure 39: Relative speed-up of inference using CG over several stages of refinement. In absolute numbers, the fastest implementation requires about one second.



A significant advantage of the CG method is that it is guaranteed to converge to the optimal solution. However, the convergence rate is rather sensitive to the condition number of $\mathbf{J}$. If this should turn out to be a problem, various ways of *preconditioning*[238] can be considered.

[238] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996

In practice, as we shall point out subsequently, one can directly learn the model parameters such as to enforce a benign condition number of $\mathbf{J}$.

Gibbs Sampling

Let us now turn to a sampling-based approach. The *Gibbs* sampler[239] is particularly popular for inferring marginal probabilities in discrete Markov random fields, where this problem is NP-hard.[240] Nonetheless, the approach is sufficiently general so that it can be adapted for inference in a Gaussian graphical model.

[239] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distribution and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984

[240] Venkat Chandrasekaran, Nathan Srebro, and Prahladh Harsha. Complexity of Inference in Graphical Models. Technical report, 2010

```
Initialize state y ← 0, mean u ← 0 ;
k ← 1 ;
repeat
    foreach s ∈ V do
        Update y_s ← y_{s|V},   y_{s|V} ~ p(y_s | y_{V\s}; h, J) ;
    if k > N_BurnIn then
        Update u ← u + y/N_Samples ;
    k ← k + 1 ;
until k > (N_BurnIn + N_Samples);
Return mean u ;
```

Figure 40: BlockedGibbsSampler algorithm

The basic algorithm is outlined in Figure 40. Again, we make use of the fact that each $\mathbf{y}_s \in \mathbb{R}^\kappa$, by performing *blocked* sampling.

The key idea behind Gibbs sampling is as follows: It is computationally expensive to draw a sample $\mathbf{y} \sim p(\mathbf{y}; \mathbf{h}, \mathbf{J})$ from the *joint* distribution, but assuming that the realizations $\mathbf{y}_{V\setminus s}$ of all variables *but one* are fixed and given, it is easy to obtain a sample $\mathbf{y}_{s|V} \sim p(\mathbf{y}_s \mid \mathbf{y}_{V\setminus s}; \mathbf{h}, \mathbf{J})$ from the *conditioned* distribution of that variable.

This requires us to compute the canonical parameters of the conditioned distribution, which we obtain by conditioning the factors adjacent to s:

$$\mathbf{h}_{s|V} = \sum_{F: s\in F} \left([\mathbf{h}_F]_s - \textstyle\sum_{t\in F\setminus s} [\mathbf{J}_F]_{st} \mathbf{y}_t \right), \tag{233}$$

$$\mathbf{J}_{s|V} = \sum_{F: s\in F} [\mathbf{J}_F]_{ss}. \tag{234}$$

Samples from the conditioned distribution can then be obtained viz.:[241]

[241] Brian D. Ripley. *Stochastic Simulation*. Wiley-Interscience, 2006

1. Compute $\mathbf{C}_{s|V} = \mathbf{J}_{s|V}^{-1} \in \mathbb{S}_{++}^\kappa$ and $\mathbf{u}_{s|V} = \mathbf{C}_{s|V}\mathbf{h}_{s|V} \in \mathbb{R}^\kappa$.
2. Compute a Cholesky decomposition $\mathbf{L}\mathbf{L}^\mathsf{T} = \mathbf{C}_{s|V}$.
3. Obtain κ independent standard normal variates $\mathbf{z} = (z_1, z_2, \ldots, z_\kappa)^\mathsf{T}$.
4. Obtain $\mathbf{y}_{s|V} = (\mathbf{u} + \mathbf{L}\mathbf{z}) \sim p(\mathbf{y}_s \mid \mathbf{y}_{V\setminus s}; \mathbf{h}, \mathbf{J})$.

Since κ is typically very small, say 3, the associated linear algebra operations can be carried out extremely efficiently. In fact, the main computational burden stems from generation of the standard normal variates—it is important to sample these numbers efficiently.

At each iteration, the Gibbs sampler then performs one sweep over the joint state vector it maintains and re-samples each variable. Variables that are conditionally independent given their Markov blanket can actually be sampled in parallel,[242] allowing to make use of multiple CPU cores.

[242] Joseph Gonzalez, Yucheng Low, Arthur Gretton, and Carlos Guestrin. Parallel Gibbs Sampling: From Colored Fields to Thin Junction Trees. In *Artificial Intelligence and Statistics (AISTATS)*, 2011

Typically, in the first few iterations, called the *burn-in* phase, the samples are discarded. An unbiased estimate of the joint mean $\mathbf{u}$ is then obtained from N joint realizations obtained in the above manner.

While the Conjugate Gradient method is typically more efficient at obtaining highly accurate solutions, the Gibbs sampling approach is attractive for two reasons: First, it allows to draw samples from the joint distribution, which is desirable for instance for visualization purposes. Second, it can be

Figure 41: GAUSSIANBELIEFPROPAGATION algorithm

foreach $F \in \mathcal{F}$ and $s \in F$ **do**
  Initialize to uninformative messages $\mathbf{m}_{F \to s} \propto \mathcal{C}(\mathbf{0}, \mathbf{0})$;
repeat
  foreach $F \in \mathcal{F}$ and $s \in F$ **do**
    Update $m_{F \to s}(\mathbf{y}_s) \propto \int_{\mathcal{Y}_{F \setminus s}} \psi_F(\mathbf{y}_F) \prod_{t \in F \setminus s} \prod_{F' \neq F : t \in F'} m_{F' \to t}(\mathbf{y}_t) \, \mathrm{d}\mathbf{y}_{F \setminus s}$;
until *converged* or *iterations exceeded*;
foreach $s \in V$ **do**
  Compute marginal distribution $p(\mathbf{y}_s) \propto \prod_{F : s \in F} m_{F \to s}(\mathbf{y}_s)$;
  Determine mean $\mathbf{u}_s \leftarrow \arg\max_{\mathbf{y}_s} p(\mathbf{y}_s)$;
Return variable mean $\mathbf{u}$;

used to obtain an estimate of (parts of) the covariance matrix, via

$$\mathbf{C} \stackrel{\text{def}}{=} \mathbb{E}[(\mathbf{y} - \mathbf{u})(\mathbf{y} - \mathbf{u})^\mathsf{T}] \approx \frac{1}{N} \sum_{k=1}^{N} (\mathbf{y}^{(k)} - \mathbf{u})(\mathbf{y}^{(k)} - \mathbf{u})^\mathsf{T}, \tag{235}$$

where $\mathbf{y}^{(k)}$ is the joint sample generated at the k-th iteration of the Gibbs sampler. In contrast, explicit computation of $\mathbf{C}$ is typically intractable due to its size—unlike $\mathbf{J}$, the covariance matrix need not be sparse.

Gaussian Belief Propagation

Finally, akin to discrete Markov random fields, once can run belief propagation in a Gaussian graphical model.[243] The resulting algorithm is closely related to *Kalman filtering*,[244] and is outlined in Figure 41.

In principle, the algorithm remains unchanged over loopy belief propagation in discrete models, except for the fact that the messages that are passed between factors and variables are now *densities*, and that the variables are *continuous*, so we need to integrate, rather than sum, over the realizations of the other variables when marginalizing.

In particular, the messages are κ-dimensional *Gaussians*. Hence, we can represent each message $\mathbf{m}_{F \to s} \propto \mathcal{C}(\mathbf{h}_{F \to s}, \mathbf{J}_{F \to s})$ as a canonical form, simply storing its parameters $\mathbf{h}_{F \to s}$ and $\mathbf{J}_{F \to s}$. An important insight is that the message updates only consist of two kinds of operations, applied to either messages or factors, both of which are canonical forms: *multiplication* and *marginalization*. As we saw previously, multiplication of canonical forms simply involves adding up their parameters, and marginalization can again be conducted in closed form using equalities (218)–(219).

Similarly, after the messages have been updated, the marginal distribution of a variable s can be computed as a product of the incoming messages. This results in a canonical form $\mathcal{C}(\mathbf{h}_s, \mathbf{J}_s)$, and consequently the estimated mean can be obtained as $\mathbf{u}_s = \mathbf{J}_s^{-1} \mathbf{h}_s$.

The correctness results[245] mostly mirror those of discrete belief propagation: In general, Gaussian BP is guaranteed to yield correct marginal probabilities only if the graph is tree-structured or if there is only a single loop. However, there is one important difference: *If* Gaussian BP *converges*, it is guaranteed to recover the exact mean. The same does not apply to estimates of the covariance that can be obtained from the algorithm, however. Recently, convergent variants of Gaussian BP have also been devised.[246]

[243] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009

[244] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001

[245] Yair Weiss and William T. Freeman. Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology. *Neural Computation*, 13(10):2173–2200, 2001

[246] Jason K. Johnson, Danny Bickson, and Danny Dolev. Fixing convergence of Gaussian belief propagation. In *IEEE International Symposium on Information Theory (ISIT)*, 2009

Maximizing the Likelihood of an Encoding

In the past chapter, we developed the tools required to work with a Gaussian mode. We will now return to our original goal, discriminative training of structured prediction models.

The particular approach we are going to consider in this chapter is based on maximizing the conditional likelihood of the observed data. In principle, a Gaussian model is most naturally applicable to continuous data points, that is, to regression problems. However, since the regression problem is strictly more general than the classification problem, the approach we present in this chapter is equally applicable to classification, simply by encoding discrete labels as particular points in a multi-dimension continuous space. Here, we are going to discuss one particularly natural way of encoding the discrete labels and discuss a few of its properties—however, in principle, numerous different encodings are feasible, and construction of such encoding schemes is an interesting direction for future work.

In any case, the use of encodings is just one way of handling discrete variables in a Gaussian model. Another possibility, which we are going to investigate in the chapter to follow, is to use a specific loss function that penalizes mispredictions in a discrete sense.

Gaussian Conditional Random Fields

Recall our general definition of a discriminative graphical model: We let the exponential parameters depend on the observed input $\mathbf{x}$, via a linear function of derived features and model weights:

$$\theta(\mathbf{x}; \mathbf{w}) = \mathbf{B}(\mathbf{x})\mathbf{w}, \quad \mathbf{w} \in \Omega. \tag{236}$$

As we saw, in a Gaussian model, the exponential parameters are determined by the precision matrix $\mathbf{J}$ and the offset vector $\mathbf{h}$. Unlike the discrete case, the exponential parameters are constrained: In particular, $\mathbf{J}$ must be positive-definite.

The question is then how to train or estimate the model parameters $\mathbf{w}$. Discriminative training of Gaussian random fields was first considered by Tappen et al.[247] The approach we are going to discuss here differs in two key ways: First of all, we use a *convex* likelihood-based learning objective such that the resulting model represents probabilities. Second, we do not use a restricted quadratic form but allow arbitrary positive-definite precision matrices to be learned. The latter point will be made clear shortly.

[247] Marshall Tappen, Ce Liu, Edward Adelson, and William Freeman. Learning Gaussian Conditional Random Fields for Low-Level Vision. In *Computer Vision and Pattern Recognition (CVPR)*, 2007

Factor Energy

We will find it convenient to model the energy of a joint realization $\mathbf{y}$ in terms of the contributions of the factors of the Gaussian random field. In a *conditional* random field, the energy of a factor can depend on the observed input $\mathbf{x}$ in terms of its linear and quadratic coefficients. Moreover, we will group factors into types T, where the type determines how these coefficients are constructed from a subset $\mathbf{w}_\mathrm{T} \subset \mathbf{w}$ of model parameters. The energy of a factor can then be written as

$$E_\mathrm{T}(\mathbf{y}_F \mid \mathbf{x}_F; \mathbf{w}_\mathrm{T}) = \frac{1}{2}\mathbf{y}_F^\mathsf{T}\mathbf{J}_\mathrm{T}(\mathbf{x}_F; \mathbf{w}_\mathrm{T})\mathbf{y}_F - \mathbf{y}_F^\mathsf{T}\mathbf{h}_\mathrm{T}(\mathbf{x}_F; \mathbf{w}_\mathrm{T}), \quad \mathbf{J}_\mathrm{T}(\mathbf{x}_F; \mathbf{w}_\mathrm{T}) \succ 0. \tag{237}$$

The above representation leaves several degrees of freedom, most notably in the definition of the local precision matrix $\mathbf{J}_\mathrm{T}(\mathbf{x}_F; \mathbf{w}_\mathrm{T})$, which must be positive-definite, and the offset term $\mathbf{h}_\mathrm{T}(\mathbf{x}_F; \mathbf{w}_\mathrm{T})$, which adjusts the location of the mean.

Simple linear model. In many cases the mapping to the output is locally well approximated as a linear function of some features derived from the input. For such cases, we propose to use an arbitrary *linear model* in each factor using a set of application-dependent *basis* functions.

Such factor-type dependent basis functions $\{b_\mathrm{T}^{(i)}\}_{i=1}^B$ can be readily employed in our model, and can depend on the observed input $\mathbf{x}$ in an arbitrary manner. For notational convenience, we arrange the basis functions into a single vector-valued function $\mathbf{b}_\mathrm{T} \colon \mathcal{X} \to \mathbb{R}^B$. The factor energy of a label $\mathbf{y}_F \in \mathbb{R}^{\kappa|F|}$ then turns into

$$E_\mathrm{T}(\mathbf{y}_F \mid \mathbf{x}_F; \underbrace{\mathbf{w}_\mathrm{T}}_{\mathbf{J}_\mathrm{T}, \mathbf{H}_\mathrm{T}}) = \frac{1}{2}\mathbf{y}_F^\mathsf{T} \underbrace{\mathbf{J}_\mathrm{T}}_{\mathbf{J}_\mathrm{T}(\mathbf{x}_F; \mathbf{w}_\mathrm{T})}\mathbf{y}_F - \mathbf{y}_F^\mathsf{T} \underbrace{\mathbf{H}_\mathrm{T}\mathbf{b}_\mathrm{T}(\mathbf{x}_F)}_{\mathbf{h}_\mathrm{T}(\mathbf{x}_F; \mathbf{w}_\mathrm{T})}, \quad \mathbf{J}_\mathrm{T} \succ 0, \tag{238}$$

and is fully determined by the matrix $\mathbf{H}_\mathrm{T} \in \mathbb{R}^{\kappa|F| \times B}$ that weights the responses of the basis functions, and the local precision matrix $\mathbf{J}_\mathrm{T} \in \mathbb{S}_{++}^{\kappa|F|}$.

Note that it is also possible to let $\mathbf{J}_\mathrm{T}$ depend on the observed input, via

$$\mathbf{J}_\mathrm{T}(\mathbf{x}_F) = \sum_i^B b_\mathrm{T}^{(i)}(\mathbf{x}_F)\mathbf{J}_\mathrm{T}^{(i)}, \quad b_\mathrm{T}^{(i)}(\mathbf{x}_F) > 0, \quad \mathbf{J}_\mathrm{T}^{(i)} \succ 0, \tag{239}$$

where the $\{\mathbf{J}_\mathrm{T}^{(i)}\}$ together with $\mathbf{H}_\mathrm{T}$ form the model parameters $\mathbf{w}_\mathrm{T}$, and each derived feature $b_\mathrm{T}^{(i)}$ must be positive to ensure positive-definiteness of the resulting precision matrix $\mathbf{J}_\mathrm{T}$. For simplicity of our presentation, we will assume the form in (238) in the following.

Global Energy

From the quadratic form of the factors, it follows that the global energy is again a quadratic function of the joint labeling,

$$E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) = \sum_\mathrm{T} \sum_{F \in \mathcal{F}_\mathrm{T}} E_\mathrm{T}(\mathbf{y}_F \mid \mathbf{x}_F; \mathbf{w}_\mathrm{T}) \tag{240}$$

$$= \frac{1}{2}\mathbf{y}^\mathsf{T}\mathbf{J}(\mathbf{x}; \mathbf{w})\mathbf{y} - \mathbf{y}^\mathsf{T}\mathbf{h}(\mathbf{x}; \mathbf{w}), \tag{241}$$

where the non-zero entries of the global coefficients are given by

$$[\mathbf{h}(\mathbf{x};\mathbf{w})]_s = \sum_{\mathrm{T}} \sum_{F \in \mathcal{F}_{\mathrm{T}}: s \in F} [\mathbf{h}_{\mathrm{T}}(\mathbf{x}_F;\mathbf{w}_{\mathrm{T}})]_s, \qquad \forall s \in V, \tag{242}$$

and

$$[\mathbf{J}(\mathbf{x};\mathbf{w})]_{ss} = \sum_{\mathrm{T}} \sum_{F \in \mathcal{F}_{\mathrm{T}}: s \in F} [\mathbf{J}_{\mathrm{T}}(\mathbf{x}_F;\mathbf{w}_{\mathrm{T}})]_{ss}, \qquad \forall s \in V, \tag{243}$$

$$[\mathbf{J}(\mathbf{x};\mathbf{w})]_{st} = \sum_{\mathrm{T}} \sum_{F \in \mathcal{F}_{\mathrm{T}}: s,t \in F} [\mathbf{J}_{\mathrm{T}}(\mathbf{x}_F;\mathbf{w}_{\mathrm{T}})]_{st}, \qquad \forall (s, t \neq s) \in E. \tag{244}$$

Importantly, assuming the linear factor model discussed above, this shows that the entries of $\mathbf{h}$ and $\mathbf{J}$ are simply linear functions of the model parameters $\mathbf{w}$, so we can write

$$\begin{pmatrix} \mathbf{h}(\mathbf{x};\mathbf{w}) \\ \mathbf{J}(\mathbf{x};\mathbf{w}) \end{pmatrix} = \mathbf{B}(\mathbf{x})\mathbf{w}. \tag{245}$$

for a suitably chosen matrix $\mathbf{B}(\mathbf{x})$ consisting of the responses of the basis functions. Consequently, our parameterization follows the general form of a discriminative graphical model set forth in the first part of the thesis, and all parameter estimation approaches discussed therein can be applied in principle (though with some caveats, as we shall see).

Relation to Previous Work

An important difference to the approach of Tappen et al.[248] is that our approach is motivated in terms of local factor models and allows to learn precision matrices $\mathbf{J}_{\mathrm{T}}$, subject only to positive-definite constraints. In contrast, Tappen et al. represent the global precision matrix as $\mathbf{J} = \mathbf{F}^{\mathsf{T}} \operatorname{diag}(\mathbf{j})\mathbf{F}$, learning only the diagonal weights $\mathbf{j}$ on convolution filters represented compactly by matrix $\mathbf{F}$.

[248] Marshall Tappen, Ce Liu, Edward Adelson, and William Freeman. Learning Gaussian Conditional Random Fields for Low-Level Vision. In *Computer Vision and Pattern Recognition (CVPR)*, 2007

Our approach requires us to ensure positive-definiteness of the $\{\mathbf{J}_{\mathrm{T}}\}$ model parameters during parameter estimation, but in turn augments expressiveness of the model significantly. Our strategy for handling the positive-definiteness constraints will be made clear shortly.

Maximum Conditional Likelihood Training

Assume now that we are given i.i.d. training data $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\}$ and want to estimate the parameters of our model. Ideally, we would be able to use the maximum likelihood estimate (MLE) of the parameters, because it is asymptotically consistent and has low asymptotic variance:[249]

[249] Robert V. Hogg, Allen Craig, and Joseph W. McKean. *Introduction to Mathematical Statistics*. Pearson Education, 2005

$$\hat{\mathbf{w}}_{\mathrm{MLE}} = \operatorname*{arg\,min}_{\mathbf{w} \in \Omega} \{ -\textstyle\sum_{(\mathbf{x},\mathbf{y})} \log p(\mathbf{y} \mid \mathbf{x};\mathbf{w}) \}, \tag{246}$$

where constraint set Ω enforces positive-definiteness of the parameters $\{\mathbf{J}_{\mathrm{T}}\}$, the precision matrices of the factor models.

In the first part of the thesis, we already discussed maximum conditional likelihood estimation for exponential families. All results we previously developed apply, but have to be specialized for the Gaussian case we are considering here.

$$-\log p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) = E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) + \log \int_{\mathbb{R}^{\kappa v}} \exp(-E(\dot{\mathbf{y}} \mid \mathbf{x}; \mathbf{w}))\, \mathrm{d}\dot{\mathbf{y}}, \tag{247}$$

$$\nabla_{\mathbf{w}}[-\log p(\mathbf{y} \mid \mathbf{x}; \mathbf{w})] = \nabla_{\mathbf{w}} E(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) - \mathbb{E}_{\dot{\mathbf{y}} \sim p(\dot{\mathbf{y}}|\mathbf{x};\mathbf{w})}\big[\nabla_{\mathbf{w}} E(\dot{\mathbf{y}} \mid \mathbf{x}; \mathbf{w})\big]. \tag{248}$$

Figure 42: General form of the negative log-likelihood and the gradient with respect to the model parameters $\mathbf{w}$.

In particular, the negative log-likelihood can again be expressed as a function of the energy of the observed label and the log-partition function. In a Gaussian model, the log-partition function generates the first and second-order expectations, so the gradient of the negative log-likelihood can be expressed as the gradient of the energy of the observed label, minus the expected gradient of the energy, as shown in Figure 42.

Computation of the gradient. Let us now make this more concrete by developing the gradient of the energy of a single factor F of type T. Assuming the simple factor model in (238), the energy is parameterized in terms of $\mathbf{w}_{\mathrm{T}} = \{\mathbf{J}_{\mathrm{T}}, \mathbf{H}_{\mathrm{T}}\}$, and we have

$$\nabla_{\mathbf{J}_{\mathrm{T}}} E_{\mathrm{T}}(\mathbf{y}_F \mid \mathbf{x}_F; \mathbf{w}_{\mathrm{T}}) = \frac{1}{2}\mathbf{y}_F \mathbf{y}_F^{\mathsf{T}}, \tag{249}$$

and

$$\nabla_{\mathbf{H}_{\mathrm{T}}} E_{\mathrm{T}}(\mathbf{y}_F \mid \mathbf{x}_F; \mathbf{w}_{\mathrm{T}}) = \mathbf{y}_F [\mathbf{b}_{\mathrm{T}}(\mathbf{x}_F)]^{\mathsf{T}}. \tag{250}$$

Computing the gradient of the expected energy with respect to the model parameters is considerably more involved. In particular, we have

$$\mathbb{E}_{\dot{\mathbf{y}} \sim p(\dot{\mathbf{y}}|\mathbf{x};\mathbf{w})}\big[\nabla_{\mathbf{J}_{\mathrm{T}}} E_{\mathrm{T}}(\dot{\mathbf{y}}_F \mid \mathbf{x}_F; \mathbf{w}_{\mathrm{T}})\big] = \frac{1}{2}\mathbb{E}_{\dot{\mathbf{y}}}\big[\dot{\mathbf{y}}_F \dot{\mathbf{y}}_F^{\mathsf{T}}\big], \tag{251}$$

and

$$\mathbb{E}_{\dot{\mathbf{y}} \sim p(\dot{\mathbf{y}}|\mathbf{x};\mathbf{w})}\big[\nabla_{\mathbf{H}_{\mathrm{T}}} E_{\mathrm{T}}(\dot{\mathbf{y}}_F \mid \mathbf{x}_F; \mathbf{w}_{\mathrm{T}})\big] = \mathbb{E}_{\dot{\mathbf{y}}}\big[\dot{\mathbf{y}}_F\big]\big[\mathbf{b}_{\mathrm{T}}(\mathbf{x}_F)\big]^{\mathsf{T}}. \tag{252}$$

The main complication is that to compute these expectations, one needs the relevant blocks of the mean

$$\mathbf{u} \stackrel{\text{def}}{=} \mathbb{E}_{\dot{\mathbf{y}} \sim p(\dot{\mathbf{y}}|\mathbf{x};\mathbf{w})}\big[\dot{\mathbf{y}}\big] = [\mathbf{J}(\mathbf{x}; \mathbf{w})]^{-1}\mathbf{h}(\mathbf{x}; \mathbf{w}) \tag{253}$$

and the second-order expectation matrix

$$\mathbf{\Sigma} \stackrel{\text{def}}{=} \mathbb{E}_{\dot{\mathbf{y}} \sim p(\dot{\mathbf{y}}|\mathbf{x};\mathbf{w})}\big[\dot{\mathbf{y}}\dot{\mathbf{y}}^{\mathsf{T}}\big] = [\mathbf{J}(\mathbf{x}; \mathbf{w})]^{-1} + \mathbf{u}\mathbf{u}^{\mathsf{T}} \tag{254}$$

pertaining to factor F. While polynomial-time, the complexity of this computation is cubic in the number variables v, specifically $\mathcal{O}(\kappa^3 v^3)$, and hence prohibitive even for instances of relatively modest size. Moreover, unlike the global precision matrix $\mathbf{J}(\mathbf{x}; \mathbf{w})$, the second-order expectation matrix $\mathbf{\Sigma}$ is typically not sparse, so infeasible amounts of memory are needed even just to store it.

For this reason, exact maximum likelihood estimation is not in general viable option. Tractable alternatives are needed that avoid computation of the matrix $\mathbf{\Sigma}$, or at least reduce its dimensionality.

$$-\log p(\mathbf{y}_s \mid \mathbf{y}_{V\setminus s}, \mathbf{x}; \mathbf{w}) = E(\mathbf{y}_s, \mathbf{y}_{V\setminus s} \mid \mathbf{x}; \mathbf{w}) + \log \int_{\mathbb{R}^\kappa} \exp(-E(\dot{\mathbf{y}}_s, \mathbf{y}_{V\setminus s} \mid \mathbf{x}; \mathbf{w}))\, \mathrm{d}\dot{\mathbf{y}}_s, \tag{255}$$

$$\nabla_{\mathbf{w}}[-\log p(\mathbf{y}_s \mid \mathbf{y}_{V\setminus s}, \mathbf{x}; \mathbf{w})] = \nabla_{\mathbf{w}} E(\mathbf{y}_s, \mathbf{y}_{V\setminus s} \mid \mathbf{x}; \mathbf{w}) - \mathbb{E}_{\dot{\mathbf{y}}_s \sim p(\dot{\mathbf{y}}_s \mid \mathbf{y}_{V\setminus i}, \mathbf{x}; \mathbf{w})}\big[\nabla_{\mathbf{w}} E(\dot{\mathbf{y}}_s, \mathbf{y}_{V\setminus s} \mid \mathbf{x}; \mathbf{w})\big]. \tag{256}$$

Figure 43: General form of the negative log-pseudolikelihood and the gradient with respect to $\mathbf{w}$ around a single conditioned variable $\mathbf{y}_s$.

Maximum Conditional Pseudo-likelihood Training

An alternative to exact maximum conditional likelihood estimation is to maximize the conditional *pseudolikelihood*[250] of the training data. To our knowledge, this approach has not been followed for training of Gaussian *conditional* random fields before.

[250] Julian Besag. Efficiency of pseudo-likelihood estimation for simple Gaussian fields. *Biometrica*, 64(3):616–618, 1977

Previously, when we discussed discrete models, we saw that pseudo-likelihood estimation does not always work well, in particular if maximum a-posteriori predictions must be obtained approximately. In a Gaussian model, we can determine the MAP prediction efficiently and exactly, so there is less reason for concern about such incompatibilities.

As in the discrete case, the true likelihood of an example is approximated by the likelihood of the individual variables, conditioned on all other variables of the graph:

$$p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) \approx \prod_{s \in V} p(\mathbf{y}_s \mid \mathbf{y}_{V\setminus s}, \mathbf{x}; \mathbf{w}), \tag{257}$$

and the estimation problem turns into

$$\hat{\mathbf{w}}_{\text{MPLE}} = \operatorname*{arg\,min}_{\mathbf{w} \in \Omega} \big\{ -\textstyle\sum_{(\mathbf{x},\mathbf{y})} \sum_{s \in V(\mathbf{x})} \log p(\mathbf{y}_s \mid \mathbf{y}_{V\setminus s}, \mathbf{x}; \mathbf{w}) \big\}. \tag{258}$$

The construction of the pseudolikelihood approximation is in fact very similar to the Gibbs sampler we considered in the previous chapter. In particular, it requires the same basic operation, namely computation of the conditioned density of a single variable $\mathbf{y}_s$.

Mean parameters. Remember that the canonical parameters of the conditioned density of variable $\mathbf{y}_s$ can be obtained by adding up the parameters of the conditioned factors connected to the variable,[251] via

[251] By the Markov property, the factors that are not connected to the variable do not affect the conditioned density.

$$\mathbf{h}_{s|V}(\mathbf{x}; \mathbf{w}) = \sum_{\mathrm{T}} \sum_{F \in \mathcal{F}_{\mathrm{T}}: s \in F} \Big([\mathbf{h}_{\mathrm{T}}(\mathbf{x}_F; \mathbf{w}_{\mathrm{T}})]_s - \textstyle\sum_{t \in F\setminus s} [\mathbf{J}_{\mathrm{T}}(\mathbf{x}_F; \mathbf{w}_{\mathrm{T}})]_{st} \mathbf{y}_t \Big), \tag{259}$$

and

$$\mathbf{J}_{s|V}(\mathbf{x}; \mathbf{w}) = \sum_{\mathrm{T}} \sum_{F \in \mathcal{F}_{\mathrm{T}}: s \in F} [\mathbf{J}_{\mathrm{T}}(\mathbf{x}_F; \mathbf{w}_{\mathrm{T}})]_{ss}. \tag{260}$$

Using these canonical parameters, one can efficiently compute the low-dimensional mean parameters $\mathbf{u}_{s|V} \in \mathbb{R}^\kappa$ and $\mathbf{\Sigma}_{s|V} \in \mathbb{S}^\kappa_{++}$ of the density,

$$\mathbf{u}_{s|V} = [\mathbf{J}_{s|V}(\mathbf{x}; \mathbf{w})]^{-1} \mathbf{h}_{s|V}(\mathbf{x}; \mathbf{w}), \tag{261}$$

and

$$\mathbf{\Sigma}_{s|V} = [\mathbf{J}_{s|V}(\mathbf{x}; \mathbf{w})]^{-1} + \mathbf{u}_{s|V} \mathbf{u}_{s|V}^{\mathsf{T}}, \tag{262}$$

as well as the normalization constant

$$A(\mathbf{h}_{s|V}, \mathbf{J}_{s|V}) = \frac{1}{2} \mathbf{h}_{s|V}^{\mathsf{T}} \mathbf{J}_{s|V}^{-1} \mathbf{h}_{s|V} + \frac{\kappa}{2} \log(2\pi) + \frac{1}{2} \log \det(\mathbf{J}_{s|V}^{-1}). \tag{263}$$

$$\mathbb{E}_{\dot{\mathbf{y}}_s \sim p(\dot{\mathbf{y}}_s | \mathbf{y}_{F\setminus s}, \mathbf{x}; \mathbf{w})} \left[\nabla_{\mathbf{J}_\mathrm{T}} E_\mathrm{T}(\dot{\mathbf{y}}_s, \mathbf{y}_{F\setminus s} \mid \mathbf{x}_F; \mathbf{w}_\mathrm{T}) \right] = \frac{1}{2} \mathbb{E}_{\dot{\mathbf{y}}_s} \left[\begin{pmatrix} \dot{\mathbf{y}}_s \\ \mathbf{y}_{F\setminus s} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{y}}_s \\ \mathbf{y}_{F\setminus s} \end{pmatrix}^\mathsf{T} \right] = \frac{1}{2} \begin{pmatrix} \boldsymbol{\Sigma}_{s|V} & \mathbf{u}_{s|V} \mathbf{y}_{F\setminus s}^\mathsf{T} \\ \mathbf{y}_{F\setminus s} \mathbf{u}_{s|V}^\mathsf{T} & \mathbf{y}_{F\setminus s} \mathbf{y}_{F\setminus s}^\mathsf{T} \end{pmatrix}, \quad (266)$$

$$\mathbb{E}_{\dot{\mathbf{y}}_s \sim p(\dot{\mathbf{y}}_s | \mathbf{y}_{F\setminus s}, \mathbf{x}; \mathbf{w})} \left[\nabla_{\mathbf{H}_\mathrm{T}} E_\mathrm{T}(\dot{\mathbf{y}}_s, \mathbf{y}_{F\setminus s} \mid \mathbf{x}_F; \mathbf{w}_\mathrm{T}) \right] = \mathbb{E}_{\dot{\mathbf{y}}_s} \left[\begin{pmatrix} \dot{\mathbf{y}}_s \\ \mathbf{y}_{F\setminus s} \end{pmatrix} \left[\mathbf{b}_\mathrm{T}(\mathbf{x}_F) \right]^\mathsf{T} \right] = \begin{pmatrix} \mathbf{u}_{s|V} \\ \mathbf{y}_{F\setminus s} \end{pmatrix} \left[\mathbf{b}_\mathrm{T}(\mathbf{x}_F) \right]^\mathsf{T}. \quad (267)$$

Figure 44: Gradient of the expected energy of a factor with respect to the model parameters.

Computation of the gradient. As shown in Figure 43, the pseudolikelihood objective and its gradient with respect to the model parameters expose a general form that is very similar to the true maximum likelihood problem we considered previously. The main difference is that the operations involving an expectation can be carried out efficiently, since the mean parameters are low-dimensional.

Again, we will consider the gradient on a per-factor basis, for factors F of type T adjacent to vertex s. Remember that we condition on all $t \in V \setminus s$, and that each factor is parameterized in terms of $\mathbf{w}_\mathrm{T} = \{\mathbf{H}_\mathrm{T}, \mathbf{J}_\mathrm{T}\}$. The gradient of the energy of the observed output $\mathbf{y}_F$ remains unchanged over the maximum likelihood formulation,

$$\nabla_{\mathbf{J}_\mathrm{T}} E_\mathrm{T}(\mathbf{y}_s, \mathbf{y}_{F\setminus s} \mid \mathbf{x}_F; \mathbf{w}_\mathrm{T}) = \frac{1}{2} \mathbf{y}_F \mathbf{y}_F^\mathsf{T}, \quad (264)$$

and

$$\nabla_{\mathbf{H}_\mathrm{T}} E_\mathrm{T}(\mathbf{y}_s, \mathbf{y}_{F\setminus s} \mid \mathbf{x}_F; \mathbf{w}_\mathrm{T}) = \mathbf{y}_F [\mathbf{b}_\mathrm{T}(\mathbf{x}_F)]^\mathsf{T}. \quad (265)$$

However, the gradient of the *expected* energy changes, since the expectation is only taken over $\mathbf{y}_s$ conditioned on the other variables. This gradient is developed in Figure 44.

Handling the Constraints: Efficient Regularization

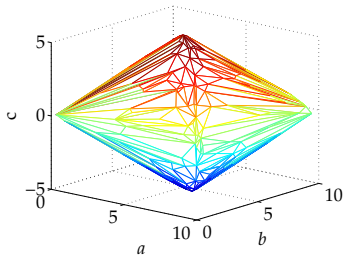


Figure 45: Convex set of 2×2 matrices $[a\ c;\ c\ b]$ whose eigenvalues are restricted to lie within $(\underline{\varepsilon} = 0.1, \overline{\varepsilon} = 10)$.

So far, we have neglected the problem of handling the constraint set Ω that enforces positive-definiteness of the local precision matrices $\{\mathbf{J}_\mathrm{T}\}$. Related to this problem is prevention of *overfitting*: Our model is expressive but can easily overfit the data if the factor models become exceedingly peaked, as determined by the associated precision matrices.

Commonly, the model parameters $\mathbf{w}$ of a conditional random field are regularized using the squared norm, $\|\mathbf{w}\|_2^2$. However, for our Gaussian conditional random field, this has neither the desired effect of keeping the precision matrices positive-definite, nor is it particularly well-motivated as a means of regularization.

Instead, we suggest to prevent both overconfident predictions and violation of positive-definiteness in a common framework, using a novel form of *regularization* for the matrix parameters. In particular, the above goal can be achieved by lower- and upper-bounding all eigenvalues of the $\{\mathbf{J}_\mathrm{T}\}$ parameters to be no smaller than a small positive number $\underline{\varepsilon}$, and no larger than a large positive number $\overline{\varepsilon}$. The set of matrices that fulfill these constraints is again convex (see Figure 45).

Through this restriction, we can enforce a favourable *condition number* of $\mathbf{J}(\mathbf{x}, \mathbf{w})$, leading to fast convergence of the conjugate gradient method

at test-time. Moreover, by adjusting $\bar{\varepsilon}$, we can push local models to be less certain of their mean, effectively regularizing the model. This can be understood as a flat prior over bounded-eigenvalue matrices, and because this set is bounded the prior is *proper*.[252]

To ensure the matrices remain in this constrained set, one can use a projection operator that builds on earlier results by Higham[253] and finds for any given matrix the closest matrix in Frobenius sense that satisfies our eigenvalue constraints. This is computationally efficient and requires one eigenvalue decomposition per $\mathbf{J}^{\mathrm{T}}$ matrix. In particular, let $\mathbf{V}\mathbf{D}\mathbf{V}^{\mathsf{T}} = \mathbf{J}_{\mathrm{T}}$ be an eigenvalue decomposition of $\mathbf{J}_{\mathrm{T}}$. The projection is then obtained as

$$\mathcal{P}_{\Omega}(\mathbf{J}_{\mathrm{T}}) = \mathbf{V} \min(\bar{\varepsilon}, \max(\underline{\varepsilon}, \mathbf{D}))\mathbf{V}^{\mathsf{T}}, \tag{268}$$

where the minimum and the maximum are applied to the diagonal matrix $\mathbf{D}$ component-wise.

Using the above projection operator, and the closed-form expressions for the gradient, the maximum pseudolikelihood estimation problem can be solved efficiently using projected gradient methods, since it is convex. We already discussed a few such methods in the previous part of the thesis on discrete models. In particular, both the spectral projected gradient method[254] and the projected quasi Newton method[255] are applicable and offer rapid convergence to the global optimum.

[252] Robert V. Hogg, Allen Craig, and Joseph W. McKean. *Introduction to Mathematical Statistics*. Pearson Education, 2005

[253] Nicholas J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103–118, 1988

[254] Ernesto G. Birgin, José M. Martínez, and Marcos Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000

[255] Mark Schmidt, Ewout Van den Berg, Michael P. Friedlander, and Kevin Murphy. Optimizing Costly Functions with Simple Constraints: A Limited-Memory Projected Quasi-Newton Algorithm. In *Artificial Intelligence and Statistics (AISTATS)*, 2009

Convexity of the Pseudolikelihood Estimation Problem

To see convexity of (258), recall that the energy $E(\mathbf{y}_s, \mathbf{y}_{V\setminus s} \mid \mathbf{x}; \mathbf{w})$ of labeling $\mathbf{y}_s$ of a conditioned subgraph around s can be written as

$$\frac{1}{2}\mathbf{y}_s^{\mathsf{T}}\mathbf{J}_{s|V}(\mathbf{x}; \mathbf{w})\mathbf{y}_s - \mathbf{y}_s^{\mathsf{T}}\mathbf{h}_{s|V}(\mathbf{x}; \mathbf{w}).$$

Observe from (259)–(260) that this function is linear in $\mathbf{w}$, and hence convex. Consider next the logarithm of the partition function normalizing

$$p(\mathbf{y}_s \mid \mathbf{y}_{V\setminus s}, \mathbf{x}; \mathbf{w}) \propto \exp(-E(\mathbf{y}_s, \mathbf{y}_{V\setminus s} \mid \mathbf{x}; \mathbf{w})),$$

defined as

$$A(\mathbf{y}_{V\setminus s}, \mathbf{x}; \mathbf{w}) = \log \int_{\mathbb{R}^{\kappa}} \exp(-E(\dot{\mathbf{y}}_s, \mathbf{y}_{V\setminus s} \mid \mathbf{x}; \mathbf{w}))\, \mathrm{d}\dot{\mathbf{y}}_s.$$

Convexity of this function in the model parameters can most easily be seen from its variational representation:

$$\max_{\mathbf{u}_{s|V}, \boldsymbol{\Sigma}_{s|V}} \left\{ -\tfrac{1}{2} \operatorname{tr}\left[\boldsymbol{\Sigma}_{s|V}^{\mathsf{T}}\mathbf{J}_{s|V}(\mathbf{x}; \mathbf{w})\right] + \mathbf{u}_{s|V}^{\mathsf{T}}\mathbf{h}_{s|V}(\mathbf{x}; \mathbf{w}) + H(\mathbf{u}_{s|V}, \boldsymbol{\Sigma}_{s|V}) \right\}$$
$$\text{s.t. } \boldsymbol{\Sigma}_{s|V} - \mathbf{u}_{s|V}\mathbf{u}_{s|V}^{\mathsf{T}} \succ 0.$$

Again, the objective is linear in $\mathbf{w}$ by the definition of the conditioned canonical parameters, and by standard results in convex optimization,[256] maximization over a family of convex functions preserves convexity. Together with linearity of the energy and convexity of the constraint set Ω, this establishes convexity of the overall problem.

[256] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004

Computational Efficiency

Besides convexity, a principal advantage of our regularized pseudolikelihood estimation approach is its computational efficiency. A particularly convenient property is that computation of the pseudolikelihood can be easily parallelized over the variables—each conditioned distribution of a variable can be computed completely independently from the others. This enables very fine-grained parallelism, which can be exploited, for instance, on modern graphics processing units (GPUs).

Furthermore, as first proposed by Nowozin et al.,[257] pseudolikelihood estimation allows us to use a subsample of the training set. To do so, one can resample a fraction of all the variables in the training set, uniformly with replacement, to obtain an unbiased estimate of the pseudolikelihood objective (258). This approach is statistically more efficient than simply reducing the number of training examples, since the variables within an example tend to be strongly correlated. Moreover, it allows to trade off accuracy against computational cost at a very fine-grained level.

[257] Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, and Pushmeet Kohli. Decision tree fields. In *13th International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011

Large-scale Training. To demonstrate the scalability of our training procedure, we want to show here the training curve resulting from a denoising experiment on the MirFlickr-25000 dataset,[258] consisting of 25,000 natural images. We use subsets of up to 5,000 images for training. The results are shown in Figure 46. They demonstrate that our approach scales to a large number of images, and that the performance of the model keeps improving as more data is used to obtain the pseudolikelihood estimate of the model parameters. Moreover, training time increases only linearly in the number of training images—a very desirable property.

We will return to the denoising task later in the thesis and provide a thorough evaluation of the denoising performance of our model. Our goal here is simply to demonstrate the convenient computational properties of the pseudolikelhood approximation, which make it very useful for large-scale regression and classification tasks. While it is clear that—compared to exact maximum likelihood estimation—some statistical efficiency is lost, the ability to handle substantial amounts of training data can compensate for this weakness, in particular if training data is plentiful or can even be generated synthetically, as in many image processing tasks.

[258] Mark J. Huiskes and Michael S. Lew. The MIR Flickr retrieval evaluation. In *International Conference on Multimedia Information Retrieval (MIR)*, 2008

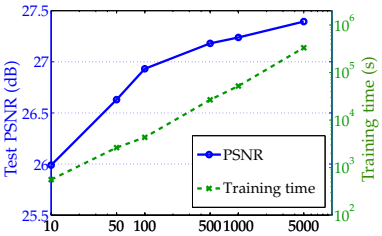


Figure 46: Training efficiency on a single computer (8 cores). We learn a denoising model for a noise level of $\sigma = 25$ from the MIRFLICKR dataset and test on 5,000 hold-out images. The peak signal-to-noise ratio (PSNR) on the test data continues to increase as more training data is used.

Handling Discrete Labels

So far, we have only touched on the topic of how discrete variables can be handled in our framework. We are now going to make this discussion concrete and discuss the properties of our approach.

In particular, we propose to *encode* discrete variables $\mathbf{y}_s$, each of which can take on one of κ discrete labels, by means of κ orthonormal vectors in $\mathbb{R}^\kappa$. In other words, the k-th component is a binary indicator that is one if and only if the k-th class label is assigned to the variable. This corresponds precisely to the sufficient statistics $\boldsymbol{\phi}(\mathbf{y}_s)$ in a discrete MRF, and is illustrated in Figure 47. The model parameters are then estimated to maximize the likelihood of this *encoding* of the observed variable labels

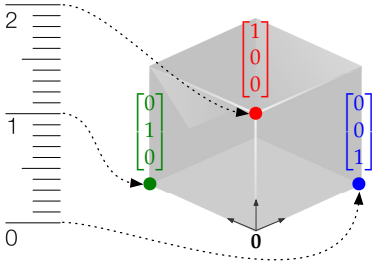


Figure 47: Discrete labels can be represented via an orthonormal basis encoding. This allows for rich interaction terms.

in the training data. At test time, the regressed mean of the Gaussian conditional random field is turned into a discrete prediction by rounding to the closest vector of sufficient statistics, in the Euclidean sense. This rounding approach is well-motivated, since the energy the model assigns to the possible realizations penalizes deviations from the mean quadratically.

Expressiveness for Discrete Tasks

The ability to learn all coefficients of the underlying quadratic energies—together with the above high-dimensional encoding of the labels—lifts the common restriction of Gaussian conditional random fields to associative[259] interactions.[260] Interestingly, such restrictions are also commonly found in discrete models.[261]

In the chapter on applications and results, we will demonstrate empirically that the expressive power of the interaction terms in our model is indeed comparable to unrestricted discrete random fields. First, we want to provide an intuitive perspective on the matter.

An experiment. Consider a learning task involving κ discrete labels, which we encode using κ orthonormal basis vectors (e.g. $[1,0,0]^\mathsf{T}, [0,1,0]^\mathsf{T}, [0,0,1]^\mathsf{T}$ for $\kappa = 3$). Remember that the energy of a pairwise factor $F = (s,t)$ of type T assumes the form

$$E_\mathrm{T}(\mathbf{y}_F \mid \mathbf{x}_F; \mathbf{w}_\mathrm{T}) = \frac{1}{2}\mathbf{y}_F^\mathsf{T}\mathbf{J}_\mathrm{T}(\mathbf{x}_F; \mathbf{w})\mathbf{y}_F - \mathbf{y}_F^\mathsf{T}\mathbf{h}_\mathrm{T}(\mathbf{x}_F; \mathbf{w}),$$

where $\mathbf{y}_F = [\mathbf{y}_s, \mathbf{y}_t]^\mathsf{T} \in \mathbb{R}^{2\kappa}$ is the vector of stacked variable labels. In contrast, in a discrete model, the energy of a particular pairwise labeling $(\mathbf{y}_s, \mathbf{y}_t)$ is determined by a $\kappa \times \kappa$ table that assigns a particular energy to each label configuration.

Interestingly, using the above κ-dimensional orthonormal basis encoding, it is always possible to choose the coefficients $\mathbf{J}_\mathrm{T} \in \mathbb{S}^{2\kappa}_{++}$ and $\mathbf{h}_\mathrm{T} \in \mathbb{R}^{2\kappa}$ (non-uniquely) such that each continuously encoded discrete label indeed receives precisely the energy assigned by *any* discrete $\kappa \times \kappa$ energy table. Using an additional constant bias term in the quadratic form, the same property can be achieved by encoding the κ discrete labels via $\kappa - 1$ orthonormal basis vectors and a single $\mathbf{0}$ vector.[262] Even repulsive energy tables can be "fitted" this way, as illustrated in Figure 48 for the special case of $\kappa = 2$, i.e. binary labels.

Such fitting of energy tables relies crucially on a sufficient number of degrees of freedom and is not in general possible using lower-dimensional encodings. To verify this, the coefficients of a quadratic form (including a bias term) were fitted to random 3×3 energy tables, where each entry was drawn uniformly at random from $(0, 5)$ at each run, and the coefficients were chosen to minimize the sum of squared errors of the assigned energies (subject to the positive-definite constraint on the quadratic coefficients).

The experiment was repeated 10,000 times, using a one-dimensional encoding of the three discrete labels as $\{0, \frac{1}{2}, 1\}$, and a two-dimensional encoding $\{[1,0]^\mathsf{T}, [0,1]^\mathsf{T}, [0,0]^\mathsf{T}\}$. Using the latter, it was *always* possible to achieve a residual of zero, whereas using the former, the residual was *never* zero, with a mean residual of 8.88 and a variance of 27.67.

[259] By *associative* interaction, we mean that the model encourages adjacent variables to take on the same labels. The opposite are *repulsive* interactions.

[260] Marshall Tappen, Kegan Samuel, Craig Dean, and David Lyle. The Logistic Random Field—A convenient graphical model for learning parameters for MRF-based labeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2008

[261] Ben Taskar, Vassil Chatalbashev, and Daphne Koller. Learning associative Markov networks. In *International Conference on Machine Learning (ICML)*, 2004

[262] In our actual model, we always use κ-dimensional basis vectors (rather than $\kappa - 1$) because the quadratic forms do not include a bias term.

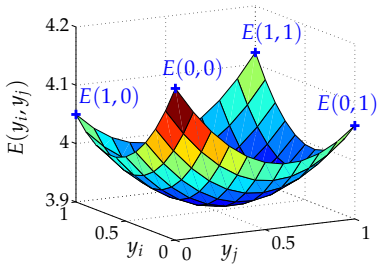


Figure 48: Quadratic fit of a repulsive pairwise discrete energy table assigning $E(0,0) = 4.16$, $E(0,1) = 4.06$, $E(1,0) = 4.05$ and $E(1,1) = 4.12$.

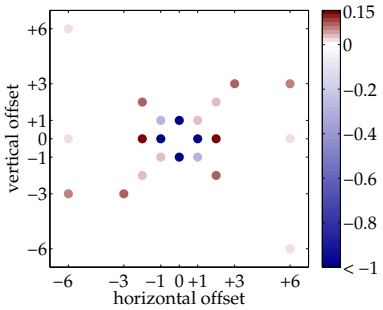


Figure 49: Associativity of the learned pairwise potentials. We plot the value of $E(0,0)+E(1,1)-E(0,1)-E(1,0)$. A negative value indicates that a pairwise interaction encourages its pixels to take the same value.

However, while the ability to match energy values at any given set of points is a necessary condition for accurate modeling of the "true" underlying distribution, this property alone is not sufficient. For instance, there may be a large probability mass far away from any discrete labeling (cf. Figure 48). The main restriction over an unrestricted discrete graphical model stems from the positive-definiteness constraint on the precision matrix. We leave the question of formalizing this trade-off for further study.

Repulsive energies. Finally, we want to show that our model indeed *learns* pairwise terms that assign repulsive energies to the continuously encoded discrete labels. Towards this end, consider the associativity strengths of the pairwise terms learned for a binary black & white inpainting task (concretely, the "Chinese characters" task we are going to consider in full detail in the chapters to follow), as displayed in Figure 49.

The figure displays the associativity of the pairwise potentials, at offsets relative to a given pixel. The offset determines the position of the second variable in the factor, relative to the variable in the center of the plot. Clearly, the learned energies encourage the pixels at some relative positions to take on similar values, whereas at other positions, disparate pixel values are encouraged. This is precisely the effect that is impossible to achieve within many restricted model classes.

It is also worthwhile to point out that the interaction terms reflect the "slant" that is naturally present in Chinese characters, and that these regularities were discovered purely by means of estimating the model parameters from training data. This demonstrates both the expressiveness of the model itself, as well as the effectiveness of the pseudolikelihood estimation framework.

Empirical Risk Minimization within the Gaussian Family

In the previous chapter, we motivated why exact maximum conditional likelihood estimation of the model parameters of a discriminative Gaussian random field is typically intractable, and discussed how pseudolikelihood estimation can be applied efficiently in our setting.

An alternative we already introduced earlier in this thesis is to minimize the *empirical risk* of the model. In the following, we are going to recapitulate a few important concepts and specialize them to our Gaussian model. An important property of the resulting approach is that it can accommodate an arbitrary differentiable loss function, and hence choose the model parameters such that they are optimal in a user-specified sense. As an added benefit, an alternative means of handling discrete labels in a Gaussian model emerges from this viewpoint.

The Empirical Risk of a Model

Recall our original goal, which is to learn a map $\hat{\mathbf{y}}\colon \mathcal{X} \times \Omega \to \mathbf{Y}$ from input to output (parameterized by some $\mathbf{w} \in \Omega$) that exposes low expected loss under the true distribution $p(\mathbf{x}, \mathbf{y})$,

$$R_\ell[\hat{\mathbf{y}}(\cdot)] = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}[\ell(\hat{\mathbf{y}}(\mathbf{x};\mathbf{w}), \mathbf{y})]. \tag{269}$$

This expectation can be approximated empirically from the n i.i.d. training examples $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\}$ at our disposal, via

$$\tilde{R}_\ell[\hat{\mathbf{y}}(\cdot)] = \frac{1}{n} \sum_{(\mathbf{x},\mathbf{y})} \ell(\hat{\mathbf{y}}(\mathbf{x};\mathbf{w}), \mathbf{y}). \tag{270}$$

The definition of the map $\hat{\mathbf{y}}(\mathbf{x};\mathbf{w})$ can in principle be constructed from the model posterior density $p(\mathbf{y} \mid \mathbf{x};\mathbf{w})$ in an arbitrary manner. The two most common approaches, maximum a-posteriori prediction and maximum posterior marginal prediction, coincide in a Gaussian model, and consequently we are going to use

$$\hat{\mathbf{y}}(\mathbf{x};\mathbf{w}) = \mathbf{u}(\mathbf{x};\mathbf{w}) = \arg\max_{\mathbf{y}} p(\mathbf{y} \mid \mathbf{x};\mathbf{w}) \tag{271}$$

$$= \arg\min_{\mathbf{y}} E(\mathbf{y} \mid \mathbf{x};\mathbf{w}) = [\mathbf{J}(\mathbf{x};\mathbf{w})]^{-1}\mathbf{h}(\mathbf{x};\mathbf{w}). \tag{272}$$

in the further development of our approach.

Choosing $\mathbf{w} \in \Omega$ to minimize the empirical risk then draws the unique mode of the model posterior density towards the observed ground truth (for each training example), in the sense of loss function $\ell\colon \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$.

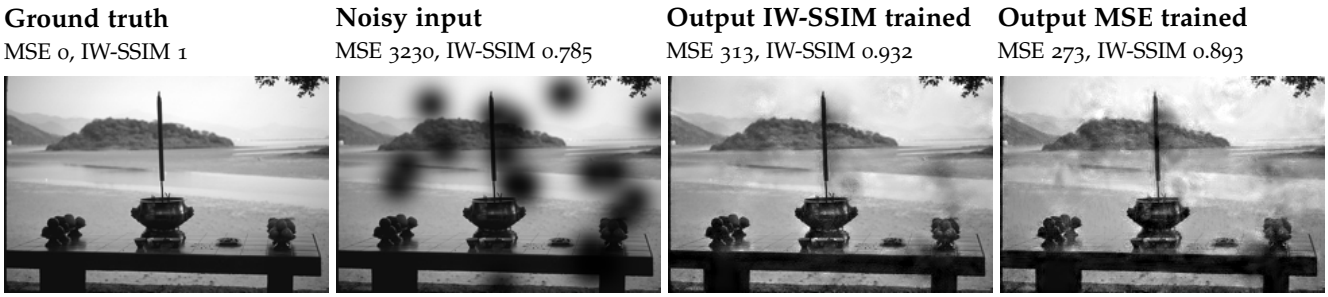


Figure 50: The loss function matters. We show reconstructions of an image corrupted by structured noise, by two models with identical specifications, except for one being optimized for information content-weighted structural similarity (IW-SSIM), and the other optimized for mean squared error (MSE). Each model has a different bias.

Importance of the loss function

The question which loss function should be optimized has received a lot of attention lately. In the past, regression models have been tuned almost ubiquitously so as to optimize the *mean squared error* (MSE), mostly due to its convenient computational properties. Recently, more complex measures of error or quality have found wide-spread acceptance.

Unfortunately, the question which loss function is most appropriate can only be answered on a per-application basis. For instance, for many image processing problems, the *structural similarity* (SSIM) index[263] has meanwhile been accepted as a better performance measure than MSE or measures based thereon, such as the *peak signal-to-noise ratio* (PSNR). When designing a structured prediction method, one should be aware of the implications of optimizing the model for one measure over the other.

[263] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004

An example. To illustrate this point, consider Figure 50. We show the predictions by competing models on an image restoration task. The goal is to remove the structured noise present in the input. The models are both of the kind discussed in this chapter, i.e. Gaussian conditional random fields, the model parameters of which have been trained to optimize a specific loss function. The difference between the two model instances is that the parameters of the first model were tuned for information-content weighted structural similarity (IW-SSIM),[264] whereas the parameters of the second model were trained to minimize the mean squared error (MSE) of its predictions. As one can see, each model clearly exposes a different bias. In particular, the model trained for MSE "hallucinates" turbulent structures in the sky that are not present in the original, uncorrupted image. In contrast, the model trained for IW-SSIM is much more successful at recovering the ground truth. This is due to the fact that the IW-SSIM measure takes into account larger image patches, rather than individual pixels, and is moreover optimized to reflect image quality as *perceived* by humans. Clearly, this makes it a more useful target for optimization.

[264] Zhou Wang and Qiang Li. IW-SSIM: Information Content Weighted Structural Similarity Index for Image Quality Assessment. *IEEE Transactions on Image Processing*, 20(5):1185–1198, 2011

Handling Discrete Labels

Evidently, other applications can require different performance measures. In principle, our approach can accommodate a wide variety of different loss functions. In fact, the only assumption we are going to make is for the function to be differentiable. This flexibility can even be exploited to

construct loss functions that penalize predictions in a *discrete* sense.

For instance, one may wish to penalize predictions by the model using the *Hamming* loss, defined as

$$\ell_{\text{Hamming}}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{s \in V} \mathbb{I}_{\hat{y}_s \neq y_s}, \tag{273}$$

where $\hat{y}_s$ denotes the predicted value of variable s. Unfortunately, this loss function is non-differentiable. However, one can easily obtain a smooth approximation. For instance, for tasks involving binary $\{-1, +1\}$ labels, Tappen et al.[265] propose to use the logistic loss

$$\ell_{\text{Logistic}}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{s \in V} \log(1 + e^{-y_s \hat{y}_s}). \tag{274}$$

[265] Marshall Tappen, Kegan Samuel, Craig Dean, and David Lyle. The Logistic Random Field—A convenient graphical model for learning parameters for MRF-based labeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2008

This approach can easily be extended to multi-class problems. For a problem involving κ classes, we predict for each variable s a vector $\mathbf{y}_s \in \mathbb{R}^\kappa$, and define the multinomial logistic loss

$$\ell_{\text{Multi}}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{s \in V} [A(\hat{\mathbf{y}}_s) - \langle \boldsymbol{\phi}(\mathbf{y}_s), \hat{\mathbf{y}}_s \rangle]. \tag{275}$$

The term contributed by each variable exposes the familiar form of the log-likelihood of a discrete exponential family, where the binary indicator vector $\boldsymbol{\phi}(\mathbf{y}_s)$ plays the role of the sufficient statistics, the prediction $\hat{\mathbf{y}}_s$ corresponds to the exponential parameters, and $A(\cdot)$ is the log-partition function, i.e. the logarithm of the sum of all exponentiated components of $\hat{\mathbf{y}}_s$. Consequently, the loss is close to zero if the component of $\hat{\mathbf{y}}_s$ corresponding to the observed ground truth is much larger than all other components.

This approach is an alternative to maximizing the pseudo-likelihood of an encoding of the observed ground truth, which we suggested previously. In line with the properties of the loss, a discrete prediction should be constructed by picking for each $\hat{\mathbf{y}}_s \in \mathbb{R}^\kappa$ the label corresponding to the component with the largest value.

Note that it is also possible to construct smooth versions of other popular discrete performance measures, such as the F1 score, using a similar approach. We leave the exploration of such loss functions for future work.

Direct Risk Minimization

Let us now consider how the model parameters $\mathbf{w}$ can be optimized efficiently so as to optimize the chosen loss function. In the introduction of the thesis, we already pointed out that direct optimization of the empirical risk is often intractable, in particular for discrete graphical models. A key feature of the Gaussian approach we pursue in this part of the thesis is that direct risk minimization is in fact feasible, and indeed can be achieved at relatively low computational cost.

Optimization of the model parameters

The key to tractability is that, as first noted by Tappen et al.,[266] the prediction of a Gaussian CRF under the current model, $\hat{\mathbf{y}}(\mathbf{x}; \mathbf{w})$, can be differentiated with respect to the model parameters $\mathbf{w}$. Even though our parameterization is more powerful than the one assumed by Tappen et al., as we will point out, this approach is still applicable to our setting.

[266] Marshall Tappen, Ce Liu, Edward Adelson, and William Freeman. Learning Gaussian Conditional Random Fields for Low-Level Vision. In *Computer Vision and Pattern Recognition (CVPR)*, 2007

$$
\begin{aligned}
\frac{\partial \ell(\hat{\mathbf{y}}(\mathbf{x},\mathbf{w}),\mathbf{y})}{\partial w_i} &= \frac{\partial \ell(\hat{\mathbf{y}},\mathbf{y})}{\partial \hat{\mathbf{y}}} \frac{\partial([\mathbf{J}(\mathbf{x};\mathbf{w})]^{-1}\mathbf{h}(\mathbf{x};\mathbf{w}))}{\partial w_i} && \text{(by chain rule)} \\
&= \tfrac{\partial \ell(\hat{\mathbf{y}},\mathbf{y})}{\partial \hat{\mathbf{y}}} [\mathbf{J}(\mathbf{x};\mathbf{w})]^{-1} \times \\
&\quad \left[\tfrac{\partial \mathbf{J}(\mathbf{x};\mathbf{w})}{\partial w_i} [\mathbf{J}(\mathbf{x};\mathbf{w})]^{-1}\mathbf{h}(\mathbf{x};\mathbf{w}) + \tfrac{\partial \mathbf{h}(\mathbf{x};\mathbf{w})}{\partial w_i} \right] && \text{(by matrix inverse rule and product rule)} \\
&= \hat{\mathbf{c}}^{\mathsf{T}} \tfrac{\partial \mathbf{J}(\mathbf{x};\mathbf{w})}{\partial w_i} \hat{\mathbf{y}} + \hat{\mathbf{c}}^{\mathsf{T}} \tfrac{\partial \mathbf{h}(\mathbf{x},\mathbf{w})}{\partial w_i}. && \text{(by substituting } \hat{\mathbf{c}} \stackrel{\text{def}}{=} [\mathbf{J}(\mathbf{x};\mathbf{w})]^{-1}[\tfrac{\partial \ell(\hat{\mathbf{y}},\mathbf{y})}{\partial \hat{\mathbf{y}}}]^{\mathsf{T}})
\end{aligned}
$$

Figure 51: Derivative of the loss function with respect to a single model parameter. It is straightforward to further develop $\frac{\partial \mathbf{J}(\mathbf{x};\mathbf{w})}{\partial w_i}$ and $\frac{\partial \mathbf{h}(\mathbf{x};\mathbf{w})}{\partial w_i}$, since the entries of $\mathbf{J}(\mathbf{x};\mathbf{w})$ and $\mathbf{h}(\mathbf{x};\mathbf{w})$ are affine functions of $\mathbf{w}$, as per the factor energy.

To see this, consider Figure 51, in which we develop the derivative of the loss function with respect to a single model parameter w_i. To evaluate the loss, the prediction $\hat{\mathbf{y}}$ for the given input $\mathbf{x}$ is required. The derivative furthermore requires the solution $\hat{\mathbf{c}}$ to a second sparse linear system of equal dimensionality. To evaluate the full gradient of the empirical risk (270), these two solutions need to be obtained once per training example. A technical complication is that the $\mathbf{J}_{\mathrm{T}}$ parameters must remain positive-definite. However, we already pointed out how these constraints can be handled efficiently: Namely by projecting the precision matrix parameters onto the convex cone of positive-definite matrices. The parameters can then be optimized using any projected gradient method.[267]

[267] Ernesto G. Birgin, José M. Martínez, and Marcos Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000; and Mark Schmidt, Ewout Van den Berg, Michael P. Friedlander, and Kevin Murphy. Optimizing Costly Functions with Simple Constraints: A Limited-Memory Projected Quasi-Newton Algorithm. In *Artificial Intelligence and Statistics (AISTATS)*, 2009

Robustness in the presence of misspecification

Note that the above procedure measures the quality of the actual predictions of our model on the training data and adjusts the model parameters so as to optimize these predictions in the specific sense of loss function ℓ. This can be thought of as a "self correcting" mechanism. The practical benefits of this approach over pseudo-likelihood estimation, as explored in the previous chapter, will studied in detail in the "Applications and Results" chapter to follow.

Intuitively, the direct risk minimization approach has the advantage that—even if the model is *misspecified*, that is, the Gaussian model is too restrictive—the parameters are still chosen such as to result in the best predictions that can be achieved by a model within this restricted class.

Notes on non-convexity of the approach

A disadvantage of the direct risk minimization approach is that it leads to a non-convex optimization problem. Even if the chosen loss function *is* convex in the prediction, the map producing the prediction, that is

$$\hat{\mathbf{y}}(\mathbf{x};\mathbf{w}) = [\mathbf{J}(\mathbf{x};\mathbf{w})]^{-1}\mathbf{h}(\mathbf{x};\mathbf{w}), \tag{276}$$

is in general *not* convex in the parameters $\mathbf{w}$ due to the matrix inversion. In practice, we found that initializing the parameters as $\mathbf{J}_{\mathrm{T}} = \mathbf{I}$ (identity) and $\mathbf{h}_{\mathrm{T}} = \mathbf{0}$ works very well, such that the optimization process does not get stuck in bad local minima. Note that it is also possible to initialize the parameters to the pseudolikelihood estimate, which can be obtained exactly. In practice, we did not observe noticeable gains over the aforementioned initialization, but of course this depends on the application. Even more potential countermeasures are described by Stoyanov et al.[268]

[268] Veselin Stoyanov, Alexander Ropson, and Jason Eisner. Empirical Risk Minimization of Graphical Model Parameters Given Approximate Inference, Decoding, and Model Structure. In *Artificial Intelligence and Statistics (AISTATS)*, pages 725–733, 2011

Increased Expressiveness via Non-Parametric Conditioning

In the previous chapters, we used a simple parameterization of the Gaussian conditional random field model that exposes linear dependence on the basis functions computed from the observed input. Indeed, in many cases, the mapping to the output is locally well approximated as a linear function of some derived input features.

On the other hand, in some cases, the relationship between the input and the output is more complex. In fact, since the introduction of the Perceptron[269]—perhaps the most prominent linear classifier—the issue of linear decision boundaries has been studied extensively, leading to the development of deep architectures such as artificial neural networks,[270] as well as kernel machines.[271]

Letting aside the issue of parameterization in terms of the observed input, the Gaussian family itself is encumbered by several restrictions, notably enforcing symmetry and uni-modality of the predictive density $p(\mathbf{y} \mid \mathbf{x})$. For this reason, it is important to fully exploit the *conditional* nature of our model.

Consider the hypothetical empirical distribution of a variable. In general, it is reasonable to assume that the data cannot be fitted well using the Gaussian bell curve, for instance due to multi-modality. On the other hand, individual subsets of the same data might be roughly normally distributed. If it is possible to distinguish between these subsets by means of inspecting the observed input, multiple Gaussian models can be used to describe each subset separately. This idea is illustrated in Fig. 52, and in fact this *divide and conquer* approach to modelling is precisely the one underlying classification and regression trees,[272] another popular machine learning paradigm that has been developed with the goal of overcoming linearity in mind.

[269] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 66(6):386–408, 1958

[270] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996

[271] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001

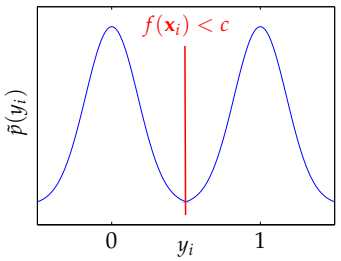


Figure 52: Via conditioning, multi-modal empirical distributions can be split into distributions that are closer to being Gaussian.

[272] Leo Breiman, Jerome Friedman, Charles J. Stone, and R. A. Olshen. *Classification and regression trees*. Chapman and Hall/CRC, 1984

Regression Tree Fields

In this chapter, we will introduce a novel conditional random field model, the *regression tree field* (RTF), which draws on regression trees in order to determine the effective interactions between variables. The basic idea underlying the approach is illustrated in Figure 53.

What makes the approach effective is that it allows to fully leverage the *conditional* aspect of a conditional random field. In particular, the dependence on the input $\mathbf{x}$, say a corrupted image, is *non-parametric*, enabling us to learn arbitrarily complex maps from the input space to the output space. At the same time, unlike an ordinary standalone regression tree approach, dependencies among the output variables $\mathbf{y}$ can be modeled effectively.

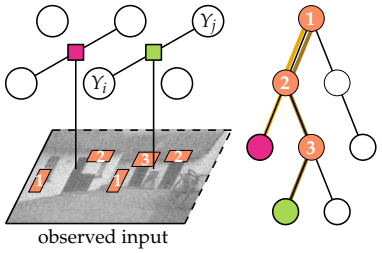


Figure 53: Illustration of how regression trees and random fields are combined in an RTF: a pairwise factor type is instantiated on a grid of random variables. At each instantiation, a tree is evaluated on the surrounding observed input, performing a sequence of tests (1,2, and 3) until a leaf is reached. For each factor, the selected node determines the effective interaction. The conditional model becomes a Gaussian random field, enabling efficient inference.

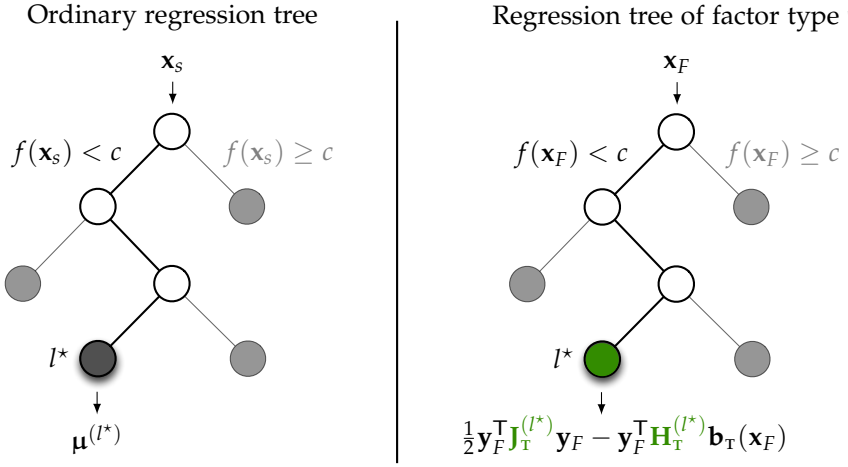


Figure 54: Regression trees: (left) the prediction is determined by the path to leaf $l^\star$ storing sample mean $\boldsymbol{\mu}^{(l^\star)}$; (right) instead of a mean, a quadratic energy is stored, determining a local model.

Parameterization in terms of regression trees

We now discuss how a *non-parametric map* from $\mathbf{x}$ to valid local factor models can be realized using regression trees. Regression trees are commonly employed as follows (see Figure 54): when inferring a prediction about label $\mathbf{y}_s \in \mathbb{R}^\kappa$ of variable s from observations $\mathbf{x}$, one follows a path from the root of the tree to a leaf l^*. This path is determined by the branching decisions made at each node, typically by computing a feature score from the observed input relative to the position of s and comparing it to a threshold. The label $\mathbf{y}_s$ is then chosen as the mean vector $\boldsymbol{\mu}^{(l^*)} \in \mathbb{R}^\kappa$ of those training points that previously ended up at the selected leaf l^*.[273]

[273] Leo Breiman, Jerome Friedman, Charles J. Stone, and R. A. Olshen. *Classification and regression trees*. Chapman and Hall/CRC, 1984

In our model, we use a similar approach to determine the parameterization of the energy term contributed by a factor of type τ in an input context-dependent manner. Our starting point is the original parameterization of factor energies in (238), in terms of model parameters $\mathbf{w}_\tau = \{\mathbf{J}_\tau \succ 0, \mathbf{H}_\tau\}$. However, rather than a single set of parameters $\mathbf{w}_\tau$, we now associate with each factor type τ a regression tree, each leaf $l \in \mathcal{L}_\tau$ of which stores a separate set of parameters $\mathbf{w}_\tau^{(l)} = \{\mathbf{J}_\tau^{(l)} \succ 0, \mathbf{H}_\tau^{(l)}\}$. The active set of parameters for a factor of type τ is then determined by the path through the associated regression tree, in terms of the selected leaf l^*, viz.:

$$E_\tau(\mathbf{y}_F \mid \mathbf{x}_F; \underbrace{\mathbf{w}_\tau}_{\{\mathbf{J}_\tau^{(l)}, \mathbf{H}_\tau^{(l)}\}_{l\in\mathcal{L}_\tau}}) = \frac{1}{2}\mathbf{y}_F^\mathsf{T}\underbrace{\mathbf{J}_\tau^{(l^*)}}_{\mathbf{J}_\tau(\mathbf{x}_F;\mathbf{w}_\tau)}\mathbf{y}_F - \mathbf{y}_F^\mathsf{T}\underbrace{\mathbf{H}_\tau^{(l^*)}\mathbf{b}_\tau(\mathbf{x}_F)}_{\mathbf{h}_\tau(\mathbf{x}_F;\mathbf{w}_\tau)}, \quad l^* = \mathrm{Leaf}(\tau, \mathbf{x}_F). \tag{277}$$

This parameterization is strictly more general than the one we previously used, which emerges as the special case of associating with each factor type a regression tree *stump* consisting only of a single leaf node.

A perhaps counter-intuitive but extremely important property is the fact that as previously, $\mathbf{J}_\tau(\mathbf{x}_F; \mathbf{w}_\tau)$ and $\mathbf{h}_\tau(\mathbf{x}_F; \mathbf{w}_\tau)$ are still *linear* functions of $\mathbf{w}_\tau$: the non-linearity enters in terms of the dependence on $\mathbf{x}_F$, which determines which subset of the parameters is in use.

A similar way of parameterizing factor energies was recently introduced for discrete conditional random fields.[274] The advantage of the model we present here is that it allows for efficient training and inference using the methods we outlined in the previous chapters.

[274] Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, and Pushmeet Kohli. Decision tree fields. In *13th International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011

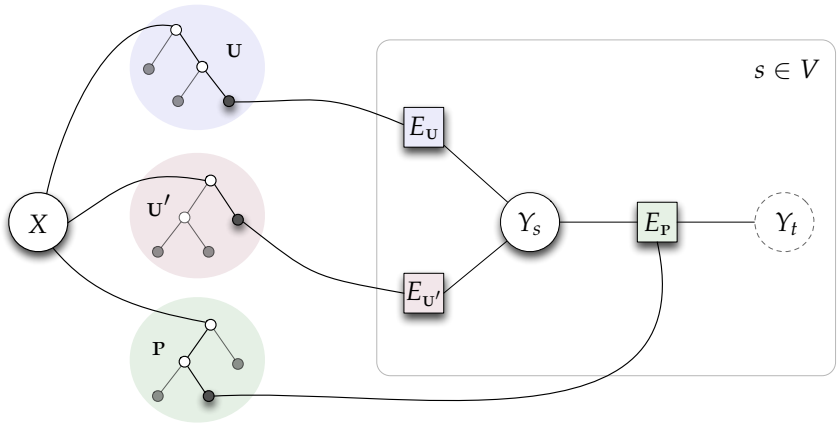


Figure 55: Example of a regression tree field: regression trees (left) determine the effective factors of type U, U′ and P, based on the observed input X, by selecting learned weights stored at their leaves. The model structure (right) is replicated once for each variable $s \in V$.

Summary of the model

Let us briefly summarize the main ingredients of our model. As illustrated in Figure 55, our model consists of several factor types, each of which is associated with a regression tree that stores at its leaves the parameters of a local quadratic energy. A factor type also specifies how factors are instantiated relative to a given variable. Importantly, factors of a common type share a local energy function that is parametrized via the quadratic models at the leaves of the associated tree, that is, the parameters of factors of a common type are *tied*. The input contents relative to the position of a factor determines the path from the root of the regression tree to the selected leaf, and hence selects the local Gaussian model that is in effect. The sum of local energy functions over the entire input determines the overall energy function.

Benefits of non-parametric conditioning

So far, we have introduced a new way of parameterizing the factor energies, which we argued increases the expressiveness of the model. Before going ahead, we wish to present some preliminary evidence that the more powerful parameterization in terms of regression trees is indeed useful.

Towards this end, consider Figure 56. We plot the denoising performance of our model on a natural image denoising task, with input images corrupted by additive white Gaussian noise, as we vary the depth of the regression trees of the pairwise factor types. Keep in mind that a depth of one corresponds to our previous Gaussian CRF parameterization.

The precise details of the denoising task are insubstantial to our discussion at this point and will be provided later in the thesis, as we present many more applications and results obtained using our model. For now, the point we wish to make is that as deeper trees are learned, more complex dependencies between the corrupted input image $\mathbf{x}$ and the original ground truth image $\mathbf{y}$ can be represented. As shown in Figure 56, this results in a substantial increase in denoising performance. To give some perspective, improvements by 0.1dB are typically visible. However, the figure also shows that overfitting can happen as the trees are trained to excessive depth, at which point the performance starts to drop again.

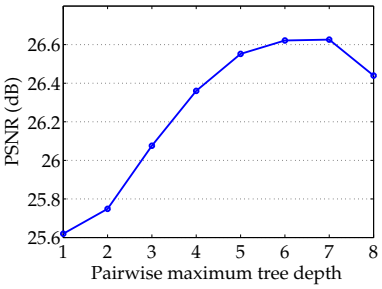


Figure 56: **Non-parametric conditioning at work:** Conditional pairwise interactions lead to improved natural image denoising ($\sigma = 25$). We vary the maximum depth of pairwise regression trees from one to eight. This increases the peak-signal-to-noise ratio (PSNR) on the test set from 25.62dB (depth one) to 26.63dB (depth seven).

Loss-Specific Learning of Regression Tree Fields

So far, we have ignored the question of how the regression trees should be chosen or learned. Ideally both the structure of the regression trees as well as the parameters at their leaves are jointly chosen to minimize a common objective function. But because the RTF model is a random field, all parts of the model interact with each other and this makes joint minimization challenging. Previously, we saw how—for the simpler parameterization—the model parameters of a Gaussian CRF can be estimated so as to minimize the empirical risk of the model. We are now going to demonstrate that it is indeed possible to extend this approach so as to jointly optimize both the choice of trees as well as the model parameters residing at their leaves in the sense of an empirical risk function. Towards this end, we will rely on two alternating optimization steps.

Optimization of the model parameters

Consider first how the model parameters $\mathbf{w}$ can be optimized for a fixed, given set of regression trees associated with the factor types of our model.

This problem is very similar to the parameter estimation problem using the simpler parameterization we previously considered. The key point is that the entries of $\mathbf{J}(\mathbf{x};\mathbf{w})$ and $\mathbf{h}(\mathbf{x};\mathbf{w})$ are still linear functions of $\mathbf{w}$. While each $\mathbf{J}_{\mathrm{T}}(\mathbf{x}_F;\mathbf{w}_{\mathrm{T}})$ or $\mathbf{h}_{\mathrm{T}}(\mathbf{x}_F;\mathbf{w}_{\mathrm{T}})$ can be a highly non-linear function of $\mathbf{x}$, it depends on the model parameters solely as a linear function of the active $\mathbf{w}_{\mathrm{T}}^{(l^*)} = \{\mathbf{J}_{\mathrm{T}}^{(l^*)}, \mathbf{H}_{\mathrm{T}}^{(l^*)}\}$ of the selected leaf l^*.

In short, the main difference over our previous model is that now we have one local model per *leaf* of a factor type, rather than one local model per factor type. To determine the partial derivative with respect to a single model parameter w_i, one can thus proceed exactly as in Figure 51.

Growing of the Trees

Conversely, assume that the model parameters have been optimized for the current tree structure. To allow for further descent in the objective, it is desirable to further grow the trees, effectively introducing new model parameters at the newly added leaf nodes. A common approach in growing stand-alone regression trees is to select splits that minimize the sum of *squared residuals*, i.e. the sum of squared distances of individual data points from their mean.[275] This approach is not well-motivated when learning an RTF. First, it is often desirable to use a loss function other than squared error, and second, the regression trees of the factor types interact with each other in the random field, so it is misguided to grow each tree as if their predictions were made separately. Nonetheless, such „standalone“ training is suggested by Nowozin et al.[276] for decision tree fields.

Instead, we propose to efficiently split all current tree nodes so as to directly decrease the empirical risk incurred by the model. Unfortunately, it is intractable to find the optimal splits under this viewpoint; however, we can base the split decisions on the largest increase in the norm of the gradient of the empirical risk function with respect to the model parameters, which should be well-correlated with the possible decrease in the objective.

[275] Leo Breiman, Jerome Friedman, Charles J. Stone, and R. A. Olshen. *Classification and regression trees*. Chapman and Hall/CRC, 1984

[276] Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, and Pushmeet Kohli. Decision tree fields. In *13th International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011

We now demonstrate that this approach is feasible for *any* differentiable loss function ℓ, and indeed, as shown in Fig. 57 (again for denoising), it results in considerable gains. The main idea is to consider the gradient contributions by individual factors as separate data points, in terms of which the split criterion can be evaluated efficiently. Let us make this more precise.

Lemma 3 *For any differentiable loss function $\ell(\cdot,\cdot)$, the derivative of the empirical risk $\tilde{R}_\ell(\mathbf{w})$ with respect to the parameters of leaf l of factor type T, $\frac{\partial \tilde{R}_\ell(\mathbf{w})}{\partial \mathbf{w}_\mathrm{T}^{(l)}}$, decomposes into contributions of the factors $F \in \mathcal{F}_\mathrm{T}^{(l)}$ for which leaf l is active.*

PROOF Consider the derivative of the loss function with respect to a single model parameter w_i, given in Fig. 51. By further noting that the entries of $\mathbf{J}(\mathbf{x};\mathbf{w})$ and $\mathbf{h}(\mathbf{x};\mathbf{w})$ are affine functions of the $\mathbf{w}_\mathrm{T}^{(l)} = \{\mathbf{J}_\mathrm{T}^{(l)}, \mathbf{H}_\mathrm{T}^{(l)}\}$ parameters, as per the factor energy (277), we develop the partial derivatives as

$$\frac{\partial \ell(\hat{\mathbf{y}}(\mathbf{x};\mathbf{w}),\mathbf{y})}{\partial \mathbf{J}_\mathrm{T}^{(l)}} = \sum_{F \in \mathcal{F}_\mathrm{T}^{(l)}} \hat{\mathbf{c}}_F \hat{\mathbf{y}}_F^\mathsf{T} \quad \text{and} \quad \frac{\partial \ell(\hat{\mathbf{y}}(\mathbf{x};\mathbf{w}),\mathbf{y})}{\partial \mathbf{H}_\mathrm{T}^{(l)}} = \sum_{F \in \mathcal{F}_\mathrm{T}^{(l)}} \hat{\mathbf{c}}_F [\mathbf{b}_\mathrm{T}(\mathbf{x}_F)]^\mathsf{T}, \tag{278}$$

where we again use $\hat{\mathbf{y}}$ and $\hat{\mathbf{c}}$ to denote the solutions to the sparse linear systems that must be solved (cf. Fig. 51), and $\hat{\mathbf{y}}_F$ and $\hat{\mathbf{c}}_F$ denote column vectors containing only the components of the variables covered by F.

Notably, $\hat{\mathbf{c}}$ is the only term in (278) that depends on the loss function, so the decomposition over factor contributions holds irrespective of the definition of $\ell(\cdot,\cdot)$, as long as it is differentiable and $\hat{\mathbf{c}}$ is thus well-defined.

To state our main result, let $\mathbf{w}$ denote the model parameters before a leaf l is split into two new leaves l_left and l_right. We denote by $\mathbf{w}'$ the parameters after a particular split. Since l is no longer a leaf in the new tree, and two new leaves are added, we have $\mathbf{w}' = \{\mathbf{w} \setminus \mathbf{w}_\mathrm{T}^{(l)}\} \cup \{\mathbf{w}_\mathrm{T}^{(l_\text{left})}, \mathbf{w}_\mathrm{T}^{(l_\text{right})}\}$.

Proposition 6 *The increase in the gradient norm $\Delta = \left\|\frac{\partial \tilde{R}_\ell(\mathbf{w}')}{\partial \mathbf{w}'}\right\| - \left\|\frac{\partial \tilde{R}_\ell(\mathbf{w})}{\partial \mathbf{w}}\right\|$ achieved by a split of leaf l of factor type T can be computed purely locally in terms of the contributions by the factors $F \in \mathcal{F}_\mathrm{T}^{(l)}$ for which leaf l is active.*

PROOF Consider the gradient norm before a split, $C \overset{\text{def}}{=} \left\|\frac{\partial \tilde{R}_\ell(\mathbf{w})}{\partial \mathbf{w}}\right\|$. The squared norm decomposes over the components of the individual leaves, so we obtain

$$\Delta = \sqrt{C^2 - \left\|\frac{\partial \tilde{R}_\ell(\mathbf{w})}{\partial \mathbf{w}_\mathrm{T}^{(l)}}\right\|_2^2 + \left\|\frac{\partial \tilde{R}_\ell(\mathbf{w}')}{\partial \mathbf{w}_\mathrm{T}^{(l_\text{left})}}\right\|_2^2 + \left\|\frac{\partial \tilde{R}_\ell(\mathbf{w}')}{\partial \mathbf{w}_\mathrm{T}^{(l_\text{right})}}\right\|_2^2} - C. \tag{279}$$

Note that C remains constant among splits and can be pre-computed. By our result of Lemma 1, the other terms depend only on the individual contributions of factors $F \in \mathcal{F}_\mathrm{T}^{(l)} = \mathcal{F}_\mathrm{T}^{(l_\text{left})} \cup \mathcal{F}_\mathrm{T}^{(l_\text{right})}$ and can thus be computed efficiently.

In practice, when evaluating split candidates, we initialize the parameters of the candidate leaves to $\mathbf{w}_\mathrm{T}^{(l_\text{left})} = \mathbf{w}_\mathrm{T}^{(l)}$ and $\mathbf{w}_\mathrm{T}^{(l_\text{right})} = \mathbf{w}_\mathrm{T}^{(l)}$. This way, the increase in gradient norm achieved by a split can be interpreted as a measure of how much gain is possible over the current parameter setting. Moreover, this approach ensures monotonic decrease in the objective

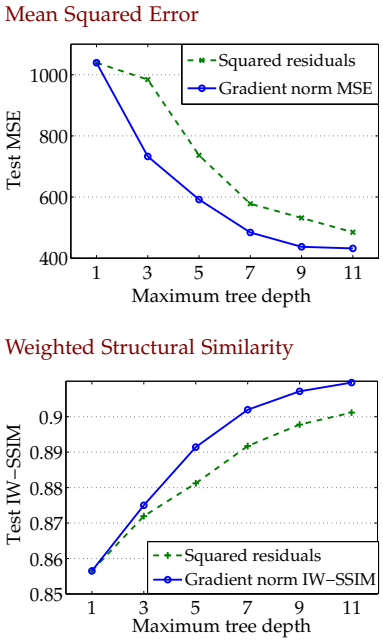


Figure 57: Benefits of splitting tree nodes according to the largest increase in gradient norm. Nodes are split either by maximizing the norm of the gradient with respect to the model parameters, or using the classic squared residuals criterion. Comparison in terms of MSE (top): the squared residuals criterion aims at the right loss, but cannot take into account that the trees are combined in a random field. Comparison in terms of IW-SSIM (bottom): the squared residuals criterion additionally optimizes the wrong loss, so the gradient norm criterion is even more important.

Figure 58: OptimizeLossJointly algorithm

Start with trees consisting solely of root nodes;
repeat
  (Re-)optimize all parameters $\mathbf{w}$ at the current leaf nodes ;
  foreach *training example* i **do**
    Solve two sparse linear systems to obtain $\hat{\mathbf{y}}^{(i)}$ and $\hat{\mathbf{c}}^{(i)}$, as in Fig. 51;
  foreach *factor type* T *and its regression tree* **do**
    foreach *training example* i **do**
      foreach *factor* $F \in \mathcal{F}_{\mathrm{T}}$ *of matching type* **do**
        Compute gradient contribution via $\hat{\mathbf{y}}_F^{(i)}$ and $\hat{\mathbf{c}}_F^{(i)}$, as in (278) ;
        Sort F and its contribution into the target leaf ;
    foreach *leaf* l **do**
      From the contributions, find the split that maximizes $\| \frac{\partial \tilde{R}_\ell(\mathbf{w}')}{\partial \mathbf{w}'} \|$;
      Split node l into new child leaves $(l_{\text{left}}, l_{\text{right}})$;
      Set $\mathbf{w}_{\mathrm{T}}^{(l_{\text{left}})} \leftarrow \mathbf{w}_{\mathrm{T}}^{(l)}$ and $\mathbf{w}_{\mathrm{T}}^{(l_{\text{right}})} \leftarrow \mathbf{w}_{\mathrm{T}}^{(l)}$;
until *maximum depth reached*;
Optimize all parameters $\mathbf{w}$ to final accuracy ;

function, since immediately after a split, the same local factor models are in effect as before. However, the degrees of freedom have increased, so further progress in the objective may be possible.

We note in passing that the above arguments also extend to computation of the increase in the norm of the *projected* gradient. Indeed, this is the criterion we use in practice, as it can be expected to more reliably reflect the possible decrease in the objective function. However, for simplicity of our exposition, we use the norm of the gradient in our development.

Putting Things Together

So far, we developed procedures for optimizing the model parameters given a fixed set of regression trees, and for splitting the trees given the model parameters that are optimal for the current tree structure. Using these building blocks, one can start from regression tree stumps consisting solely of root nodes and optimize over the model parameters and the tree structure in a greedy manner. At each iteration, the model parameters are first optimized, and the leaves of the trees are then split according to the largest increase in gradient norm to enable further progress in the objective. This iterative scheme is outlined in Figure 58. The main hyper parameter is the maximum depth of regression trees, which we suggest should be determined from validation data.

Joint Optimization of the Pseudolikelihood

Prior to introducing the empirical risk as our objective of choice, we suggested pseudolikelihood estimation as a tractable alternative to exact maximum likelihood estimation. It is natural to ask whether an analogous joint training routine, which optimizes both the structure of the trees and the model parameters so as to maximize the pseudolikelihood, can be realized. Indeed, this is possible—again, the idea is to choose splits leading

Start with trees consisting solely of root nodes;
repeat
  (Re-)optimize parameters of current leaf nodes ;
  foreach *conditioned subgraph* s **do**
    Pre-compute mean parameters $\mathbf{u}_{s|V}, \mathbf{\Sigma}_{s|V}$;
  foreach *factor type* T *and its tree* **do**
    foreach *conditioned subgraph* s **do**
      foreach *factor* $F \in \mathcal{F}_\mathrm{T}$ *of matching type* **do**
        Compute gradient contribution via $\mathbf{u}_{s|V}, \mathbf{\Sigma}_{s|V}$ – as in Fig. 44 ;
        Sort contribution into target leaf ;
      foreach *leaf* p **do**
        From the contributions, find the split that maximizes $\| \frac{\partial \mathcal{O}_{\mathrm{NLPL}}(\mathbf{w}')}{\partial \mathbf{w}'} \|$;
        Split node l into new child leaves $(l_\mathrm{left}, l_\mathrm{right})$;
        Set $\mathbf{w}_\mathrm{T}^{(l_\mathrm{left})} \leftarrow \mathbf{w}_\mathrm{T}^{(l)}$ and $\mathbf{w}_\mathrm{T}^{(l_\mathrm{right})} \leftarrow \mathbf{w}_\mathrm{T}^{(l)}$;
until *maximum depth reached*;
Optimize parameters of leaf nodes to final accuracy ;

Figure 59: OptimizeLikelihood-Jointly algorithm

to the largest increase in gradient norm. In this setting, the gradient norm with respect to model parameters $\mathbf{w}_\mathrm{T}^{(l)} = \{\mathbf{J}_\mathrm{T}^{(l)}, \mathbf{H}_{\mathrm{T}^{(l)}}\}$ of a given leaf l has a particularly intuitive interpretation: In particular, it can be thought of as a measure of disagreement between the mean parameters $\{\mathbf{u}_{s|V}, \mathbf{\Sigma}_{s|V}\}$ and the empirical distribution of the labels $\{\mathbf{y}_s, \mathbf{y}_s\mathbf{y}_s^\mathsf{T}\}$ in the conditioned subgraphs affected by the leaf. Consequently, this split criterion prefers splits introducing new parameters relevant to those subgraphs where the disagreement is largest, as these are most likely to achieve significant gains in terms of the pseudolikelihood.

The algorithm in Figure 59 gives an outline of how this works. As previously, the key to tractability is that the increase in gradient norm is computed for the parameters of the candidate child nodes set to those of their parent node. This way, the increase in overall gradient norm again can be computed efficiently and purely locally in terms of the norms resulting from the gradient contributions of the factors that are relevant to the respective candidate child. As previously, by initializing the parameters of the new leaf nodes to those of their parent, the algorithm achieves monotonic descent in the negative log-pseudolikelihood (denoted by $\mathcal{O}_{\mathrm{NLPL}}$). This holds even if re-optimization of the parameters at each round is approximate, which is often preferable from an efficiency perspective.

Practical benefits. Figure 60 shows the performance of the same RTF denoising model given additive white Gaussian noise with $\sigma = 10$ (see the chapter to follow for details), both for joint minimization of the negative log-pseudolikelihood and separate training. Joint training optimizes the learning objective more effectively as a function of the tree depth, producing in this case more accurate predictions in terms of the error measure (PSNR). For other noise levels, joint training always optimized the learning objective better, which, however, did not always improve PSNR. Again, this is a strong argument in support of empirical risk minimization.

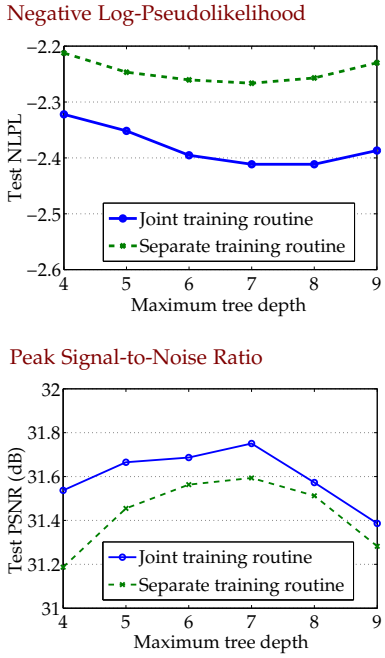


Figure 60: Joint training reduces the negative log-pseudolikelihood objective faster than separate training (top), which translates into improved peak signal-to-noise ratio (bottom).

Applications and Results

In the previous chapters, we first introduced a Gaussian conditional random field model based on local quadratic energies, and subsequently discussed how the coefficients of these energy terms can be learned freely (modulo a positive-definiteness constraint on the local precision matrices), either by maximizing the pseudolikelihood of the training data, or by directly minimizing the loss of the predictions on the training data obtained from the model, i.e. the empirical risk. Subsequently, we introduced *Regression Tree Fields* (RTFs), which extend the basic aforementioned model by allowing for non-parametric dependence on the input, effectively determining the active interactions in the Gaussian random field via the paths taken through regression trees.

We are now going to investigate the performance of these models by means of several structured prediction tasks. First of all, we are interested in the expressive power of the Gaussian CRF models we previously introduced. What kind of problem can be tackled successfully using these models, and how does a Gaussian model compare to alternative approaches, in particular when discrete labels shall be predicted? Moreover, does non-parametric conditioning substantially improve the predictive accuracy of a Gaussian CRF? We are going to answer these question by means of several specifically constructed benchmark tasks.

Moreover, we will consider a problem of great practical relevance, *image restoration*, and discuss the application of RTFs to common instances of this problem, such as *natural image denoising* and *removal of JPEG artifacts*. The former in particular has received an enormous amount of attention in the image processing literature, and highly engineered algorithms exist that exploit the specific properties of the task. Nonetheless, as we will show, our image restoration framework based on RTFs allows for substantial improvements over the state of the art.

Discrete Problems

We will start by considering two problems involving discrete labels. In the previous chapters, we introduced two different ways of handling such labels in Gaussian conditional random fields: a) maximizing the pseudolikelihood of an orthonormal basis encoding, and b) minimizing the empirical risk of the model using a multi-nomial logistic loss function that penalizes incorrect predictions in a discrete sense.

The results we are going to present were obtained using the former approach; while it is possible that direct risk minimization might result in

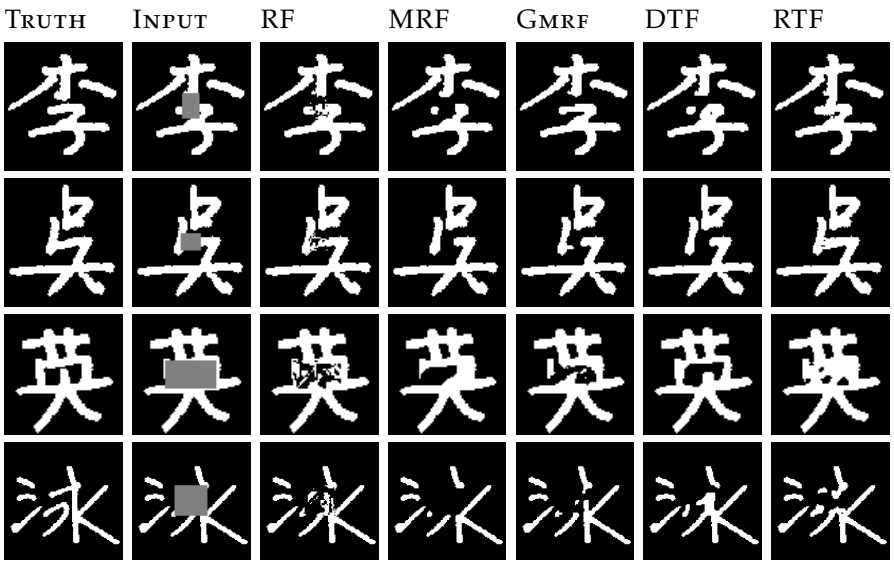


Figure 61: Chinese characters with large occlusions—test set predictions.

some minor improvements, it does not increase the overall expressiveness of the model. Indeed, restrictions concerning the expressive power mostly stem from the positive-definiteness constraint on the global system matrix, which is present irrespective of the particular parameter estimation routine.

Chinese Characters

The first task we are going to consider is of interest to us since it was used by Nowozin et al.[277] to assess the performance of the discrete *Decision Tree Field* (DTF) model, which can be seen as the discrete counterpart of our RTF model—allowing for a direct comparison.

[277] Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, and Pushmeet Kohli. Decision tree fields. In *13th International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011

The goal in this task is to in-paint the occluded parts of handwritten Chinese characters from the KAIST Hanja2 database (Figure 61). Each character is occluded by a centred grey box of varying size. Following Nowozin et al., we measure the prediction accuracy on a dataset with small occlusions, and visualize the predictions on images with larger occlusions. We replicate the DTF model as closely as possible (same features and neighborhood). For RTF training, we consider 2D orthonormal basis encoding $\{[1\ 0]^T, [0\ 1]^T\}$, as well as plain 1D encoding of the binary labels. We consider a Gaussian MRF where the pairwise trees are restricted to a single leaf (GMRF), as well as systems with deep pairwise trees (RTF), analogous to the MRF and DTF systems of Nowozin et al. We also include the random forest (RF) baseline result of Nowozin et al. for comparison.

All systems were trained on a training set of 300 training images (pairs of occluded images and the corresponding ground truths), and evaluated on a disjoint set of 100 occluded test images. Since the in-painting task is highly ambiguous, this evaluation is performed using somewhat smaller occlusions than depicted in Figure 61 (again, following Nowozin et al.), whereas the visualization is performed on the larger occlusions to highlight the different biases of the models.

The results are shown in Table 3. Our 2D-encoded systems are very competitive, with a particular RTF system achieving the best result on this task

	DEPTH_U	DEPTH_P	TEST	TRAIN
Random Forest	15	∼	67.74%	∼
Regression Tree 1D	15	∼	70.50%	77.91%
Regression Tree 2D	15	∼	69.70%	77.12%
Discrete MRF	15	1	75.18%/≈20s	∼
Gaussian MRF 1D	15	1	70.14%/0.19s	73.11%
Gaussian MRF 2D	15	1	74.19%/0.32s	80.97%
DTF	15	6	76.01%/≈20s	∼
RTF 1D	15	6	75.37%/0.27s	79.38%
RTF 2D	15	6	75.02%/0.49s	81.73%
RTF 1D	20	20	76.39%/0.23s	94.56%
RTF 2D	20	20	**77.55%**/0.24s	**94.91%**

Table 3: Chinese characters—accuracy on small occlusions: The prediction accuracy of a model is computed as the fraction of pixels in the occlusion box that are correctly inpainted by the model.

so far. Moreover, the best RTF system requires typically 0.2s per prediction, which is two orders of magnitude faster than the current DTF implementation (Nowozin et al.; private communication with the authors). The DTF predictions were obtained using simulated annealing and therefore may not be optimal, whereas inference in the RTF model is always exact. Note that 2D encoding is particularly important for GMRF, where the restricted pairwise terms in 1D encoding cannot be compensated for by conditioning. If deeper pairwise trees are allowed, as in RTF, this difference mostly vanishes. This clearly demonstrates the utility of non-parametric conditioning on the image contents.

Snakes

Next, we are going to consider a multi-label discrete learning task with weak local evidence for any particular label; the ability of the pairwise terms to capture the relevant interactions is crucial. Each "snake" (Figure 62) consists of a sequence of adjacent pixels whose color in the input encodes the direction of the next pixel: *go north* (red), *go south* (green), as well as *go east* (yellow) and *go west* (blue). Each snake is 10 pixels long, and in output space exposes a grey-scale gradient that starts at its head in black and ends at its tail in white.

Severe limitations in the expressiveness of our model, if any, would prevent us from learning the map from input space to output space.

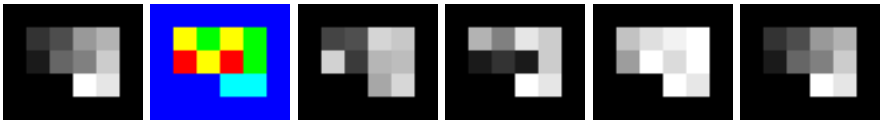


Figure 62: "Snakes" task—1D encoding seeks to minimize RMSE; 11D encoding injects a loss that is closer to multi-label error.

Again, we use the systems from the DTF paper[278] as our baseline. For RTF-training, we compare 1D encoding, which directly models the grey-scale pixel intensity, to 11D encoding that assigns an orthonormal basis label to each of the 11 different grey-scale values. The latter "injects" a particular loss function during training: Since all labels are equally close in

[278] Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, and Pushmeet Kohli. Decision tree fields. In *13th International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011

Table 4: Results on the "Snakes" test data, 4-connected model: The predictive performance of each model is measured both via the root mean squared error (RMSE) over all predicted continuous pixel values, as well as the percentage of correctly predicted discretized pixel values (Accuracy).

	DEPTH_U	DEPTH_P	ACCURACY	RMSE
Random Forest	25	$\sim$	90.30%	$\sim$
Decision Tree	25	$\sim$	90.90%	$\sim$
Regression Tree 1D	36	$\sim$	82.69%	0.1020
Regression Tree 11D	36	$\sim$	82.43%	0.1125
MRF	25	1	91.90%	$\sim$
Gaussian MRF 1D	36	1	82.52%	0.0999
Gaussian MRF 11D	36	1	84.22%	0.1352
DTF	25	15	**99.40%**	$\sim$
RTF 1D	0	10	91.14%	0.0512
RTF 11D	0	7	98.77%	**0.0268**

Euclidean space, we attain invariance with respect to label permutations, and MPLE minimizes a quadratic approximation of the discrete multi-label error. In contrast, in 1D grey-scale encoding, MPLE minimizes a quadratic loss that is closely correlated with RMSE. The regressed label of a pixel is decoded as follows: For 1D encoding, RMSE can be computed directly from the prediction, while multi-label error is computed by rounding to the nearest discrete label. With 11D, we find the basis vector closest to the prediction and use the corresponding grey-scale value (RMSE) or discrete label (multi-label error).

The numeric results are given in Table 4, and example predictions are shown in Figure 62. Tree depths were optimized for each system. RTF using 11D encoding and DTF essentially solve the task, while all other systems fail. Consider the error rates achieved by GMRF: Clearly, 11D encoding leads to smaller multi-label error, while 1D encoding favours RMSE. On the other hand, using the fully conditional pairwise terms of the RTF, 11D encoding yields better results in terms of both error metrics. This result suggests that high-dimensional encodings yield additional benefits even beyond the above loss function perspective. In particular, it may be useful to construct high-dimensional codebooks that preserve the original distances in 1D space.

A Mixed Problem

We now consider a problem that requires both discrete labels and continuous labels. As we shall see, our method is capable of solving such problems very effectively.

Detection and Registration

In this task we jointly detect and register deformable objects within an image. The input, Figure 63(b), are two flags with variable position and deformation. We use the 60 deformations provided by Garg et al.[279] The background is an arbitrary crop from a large mosaic of flags. The output labeling, see Figure 63(a) is a 3D (RGB) labeling where the first channel defines fore- and background and the last two represent the mapping of each pixel to a reference frame of the flat flag. We use an RTF model with dense

[279] Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Robust trajectory-space TV-L 1 optical flow for non-rigid sequences. In *8th international conference on Energy minimization methods in computer vision and pattern recognition (EMMCVPR)*, pages 300–314, 2011

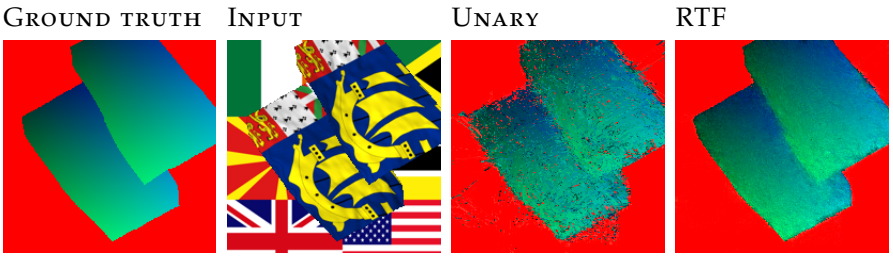


Figure 63: Detection and registration—from left to right: ground truth (RGB), input image, unary prediction, RTF 5x5 prediction.

pairwise connectivity in a 5×5 window around each pixel and a maximum tree depth of 50 for all trees, trained for maximum pseudolikelihood. We compare to a similar model involving only unary factors (UNARY) for each pixel, roughly corresponding to a standalone regression tree approach.

We use 400 generated training images and 100 test images to evaluate both systems. Figure 63(c,d) shows that a regression tree field performs much better than the simpler model. This is clearly reflected in the mean squared error (MSE); $6.1 \cdot 10^{-2}$ for UNARY, versus $1.0 \cdot 10^{-2}$ for the RTF.

While the joint detection and registration problem may seem to be somewhat obscure, it is in fact a proxy task for important real-world problems such as human pose estimation, e.g. Girshick et al.[280] Since this problem has been approached very successfully using regression trees, it is safe to assume that the RTF framework could be employed gainfully in this setting.

[280] Ross Girshick, Jamie Shotton, Pushmeet Kohli, Antonio Criminisi, and Andrew Fitzgibbon. Efficient Regression of General-Activity Human Poses from Depth Images. In *International Conference on Computer Vision (ICCV)*, 2011

A Continuous Problem

Before turning to more practical applications, we consider one more example application that is mainly meant to illustrate the expressiveness and utility of our method.

Face Colorization

Colorization is the task of adding color to a gray-scale image, e.g. an old photograph. In most works, e.g. by Levin et al.,[281] this under-constrained task is solved with some user guidance. Here we demonstrate a fully automatic system that exploits domain knowledge. We are given a training set of 200 frontal faces and a test set of 200 different people,[282] where the face images are roughly registered. Given the gray-scale input, the goal is to predict the three-dimensional (RGB) output, as illustrated in Figure 64.

As the features of our RTF model, we use Haar-wavelets of size 1–32 pixels at various relative offsets (Gaussian-distributed with $\sigma = 10$ pixels). We train the model for maximum pseudolikelihood and compare its predictions, achieving an MSE of $4.7 \cdot 10^{-4}$, to several competitors (Figure 64):

[281] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *SIGGRAPH*, 2004

[282] Images available from `http://fei.edu.br/~cet/facedatabase.html`

GLOBAL AVG I: A simple, "global-average" competitor. First, 10 nearest-neighbor images are retrieved from the training set, in terms of pixel-wise gray-scale difference. Then these images are superimposed and the median color (hue, saturation) is computed at every pixel location. Since the NN-faces are not perfectly registered, color bleeding (e.g. around the left ear) can be observed, translating into an MSE of $7.3 \cdot 10^{-4}$.

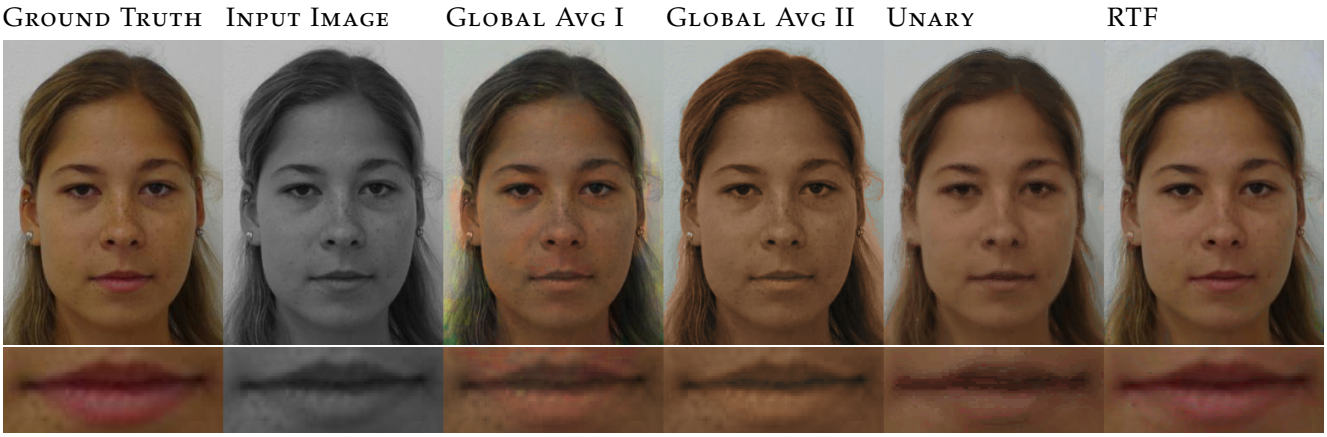


Figure 64: **Face colorization** (top row: full images, bottom row: zoom-in). Given a gray-scale test image, the goal is to recover its color (best viewed in color).

Global Avg II: A second competitor, which uses the same 10 nearest-neighbors as the above. For each luminance value, the median color (hue, saturation) is derived. The result does not show color-bleeding, but suffers from the fact that the whole face and hair has virtually the same color. This is also reflected in the worse MSE of $7.8 \cdot 10^{-4}$.

Unary: Our result with unary factors only (one tree, depth ten). While the overall result is encouraging the details are unfortunately blurry (see the zoom-in). This is likely caused by the fact that neighboring pixels make independent decisions, and results in an MSE of $8.2 \cdot 10^{-4}$.

Compared to these competitors, the results achieved by our system (RTF) are both visually superior *and* achieve a lower error rate, by a wide margin.

Image Restoration

We now turn to our final application, which is at the same time the most relevant from a practical perspective. Image restoration has a rich history in image processing, with special cases such as image denoising having received significant attention over the years. In general terms, the problem can be defined as follows: a natural image $\mathbf{y}$ is corrupted by a distortion process $\mathbf{x} = \mathbf{f}(\mathbf{y})$. We are only given the corrupted image $\mathbf{x}$ and our goal is to recover the original image through an estimate $\hat{\mathbf{y}}$. Ideally, the estimate could be obtained through the inverse process $\mathbf{f}^{-1}$. However, in practice $\mathbf{f}$ is either stochastic in nature, or deterministic but non-invertible. As a consequence, perfect reconstruction of $\mathbf{y}$ is impossible most of the time.

While one can still aim at finding a reconstruction $\hat{\mathbf{y}}$ that is reasonably close to the original image, this immediately raises the question how the quality of such a reconstruction should be measured. In the past, the squared error $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ has often been chosen because it is convenient computationally. More recently, measures of *perceived quality*, such as the structural similarity index[283] have been accepted as a better performance measure. When designing a restoration method, one should be aware of the implications of optimizing the algorithm for one measure over the other. As we saw in Figure 50, this choice affects the reconstructions considerably.

[283] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004

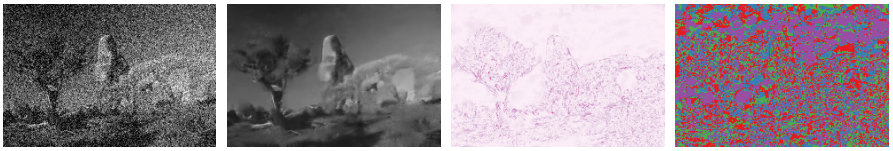


Figure 65: Existing denoising methods contain complementary information: we process a noisy image (left, $\sigma = 40$) using four denoising methods (BM3D, EPLL, FoE, LSSC). For each pixel and using the ground truth, we select the best possible prediction among the methods (second column). Compared to the ground truth this prediction has some remaining error (third column). In different parts of the image different methods are selected over larger regions (fourth column), indicating that the methods are consistently different and have varying strengths that depend on the image content.

Our contributions

In this section, we introduce a novel image restoration framework based on the non-parametric Regression Tree Field (RTF) model we devised in the previous chapters. We draw on the direct risk minimization approach we previously introduced in order to directly optimize both the regression trees associated with factor types, as well as the parameters at their leaves, for a user-specified performance measure.

Owing to the RTF framework, our model is a highly-connected conditional random field that produces globally consistent image reconstructions tailored to specific loss functions. Both image features and reconstructions made by existing restoration methods are seamlessly integrated into the random field, and their dependency on the local image context is represented non-parametrically.

Moreover, we present the first learning-based approach that directly optimizes all aspects of a model for measures of *perceived* quality. In terms of the structural similarity index (SSIM), but also peak signal-to-noise-ratio (PSNR) and mean absolute error (MAE), we obtain the best published image denoising results by a statistically significant margin. Our method is visibly better than the best published methods, LSSC by Mairal et al.[284] and EPLL by Zoran and Weiss.[285] We further present results in removal of JPEG blocking artefacts that surpass the state-of-the-art SA-DCT method.[286]

Previous approaches and related work

Image denoising has a rich history in image processing and a wide variety of image denoising methods exist. Patch-averaging methods such as BM3D[287] build weighted averages of noisy image patches and combine these into a single prediction. Sparse coding methods like LSSC optimize a dictionary of image patches within each image. Fields-of-Experts (FoE)[288] use a higher-order Markov random field as generative probabilistic image model and combine it with an analytic noise model to obtain a posterior distribution over noise-free images. Finally, as the last approach we will consider here, the recent *expected patch log likelihood* (EPLL) method uses an image patch model but combines all individual patch predictions to jointly maximize the expected patch likelihood of the predicted image.

As far as measures of perceived quality are concerned, Estrada et al.[289] optimize a stochastic denoising method explicitly for SSIM, but only using a few manually set hyper parameters. In contrast, our method is capable of optimizing tens of thousands of parameters automatically for SSIM.

Our observation that existing methods can be complementary has previously been made in a different context, namely optical flow estimation.[290] This suggests that our approach is widely applicable and may lead to gains even in other areas.

[284] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local Sparse Models for Image Restoration. In *International Conference on Computer Vision (ICCV)*, 2009

[285] Daniel Zoran and Yair Weiss. From Learning Models of Natural Image Patches to Whole Image Restoration. In *International Conference on Computer Vision (ICCV)*, 2011

[286] Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images. *IEEE Transactions on Image Processing*, 16(5):1395–1411, 2007

[287] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007

[288] Stefan Roth and Michael J. Black. Fields of Experts. *International Journal of Computer Vision (IJCV)*, 82(2):205–229, 2009

[289] Francisco Estrada, David Fleet, and Allan Jepson. Stochastic Image Denoising. In *British Machine Vision Conference (BMVC)*, 2009

[290] Oisin Mac Aodha, Gabriel J. Brostow, and Marc Pollefeys. Segmenting Video into Classes of Algorithm-Suitability. In *Computer Vision and Pattern Recognition (CVPR)*, 2010

Applying Regression Tree Fields to Image Restoration

The image restoration problem maps into our framework as follows. The observed input image $\mathbf{x}$ denotes the corrupted image, which is generated from ground truth $\mathbf{y}$ via some perturbation process. In the classical image denoising setting, an additive white Gaussian noise assumption is made, that is, $\mathbf{x} = \mathbf{y} + \mathbf{z}$ for $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. However, we will also consider images corrupted by JPEG blocking artefacts and a structured noise model (see Figure 69) in our experiments. In fact, the ability to handle arbitrary noise models is a major strength of our approach.

The restored image $\hat{\mathbf{y}}$ is then obtained as the prediction of our model given the corrupted input, i.e. $\hat{\mathbf{y}} \stackrel{\text{def}}{=} \hat{\mathbf{y}}(\mathbf{x}; \mathbf{w}) = [\mathbf{J}(\mathbf{x}; \mathbf{w})]^{-1}\mathbf{h}(\mathbf{x}; \mathbf{w})$.

Feature engineering. Remember that the entries of $\mathbf{J}(\mathbf{x}; \mathbf{w})$ and $\mathbf{h}(\mathbf{x}; \mathbf{w})$ arise as sums of per-factor contributions $\mathbf{J}_{\mathrm{T}}(\mathbf{x}_F; \mathbf{w}_{\mathrm{T}}) \stackrel{\text{def}}{=} \mathbf{J}_{\mathrm{T}}^{(l^*)}$ and $\mathbf{h}_{\mathrm{T}}(\mathbf{x}_F; \mathbf{w}_{\mathrm{T}}) \stackrel{\text{def}}{=} \mathbf{H}_{\mathrm{T}}^{(l^*)}\mathbf{b}_{\mathrm{T}}(\mathbf{x}_F)$, which depend on the evaluation of a regression tree.

Depending on our system configuration, the basis vector $\mathbf{b}_{\mathrm{T}}(\mathbf{x}_F)$ in the leaf model of a unary or pairwise factor is initialized from one or more of the following sources: a) the corrupted image itself, b) responses of a fixed filterbank, and c) predictions by base methods; at the position of the pixels covered by the factor.

In the regression trees, we use feature tests inspecting the input image, the filter responses and the output of base methods at offsets relative to the position of a factor. For JPEG deblocking, we use two more feature tests indicating whether the position of the factor lies at the boundary of a 4×4 or 8×8 block. For the filter responses, we use the RFS filterbank[291] to derive 38 responses per pixel of the input image.

[291] http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html

The use of base methods varies depending on the restoration task and will be described per experiment. Our motivation is as follows: for many established image restoration tasks, there exist highly engineered task-specific methods. These competing approaches often contain complementary information, as illustrated for denoising in Figure 65. In our RTF model, the relative contribution of the base methods can be learned per image context, such that their complementary strenghts can be exploited.

Model selection and training. We choose among RTF models with dense pairwise connectivity in either a 3×3 or a 5×5 window centered around the current pixel, and tree depths of 1, 3, 5, 7, 8 or 9, based on validation data (in most cases, a 5×5 field at depth 8 or 9 was selected).

We train and evaluate using peak signal-to-noise ratio (PSNR); mean absolute error (MAE); and unweighted structural similarity (SSIM), defined over fixed 8×8 windows as in Wang and Simoncelli.[292] To *train* for MAE, we use the smoothed, differentiable version suggested by Tappen et al.[293], but evaluate in terms of the original definition. All measures are computed per image and then averaged over the number of images.

[292] Zhou Wang and Eero P. Simoncelli. Maximum differentiation (MAD) competition: Methodology for comparing computational models of perceptual quantities. *Journal of Vision*, 8(12):1–13, 2008

[293] Marshall Tappen, Ce Liu, Edward Adelson, and William Freeman. Learning Gaussian Conditional Random Fields for Low-Level Vision. In *Computer Vision and Pattern Recognition (CVPR)*, 2007

For regularization, we follow the procedure we previously suggested and not only restrict the $\mathbf{J}_{\mathrm{T}}^{(l)}$ parameters to be positive-definite, but furthermore bound their eigenvalues by $(10^{-2}, 10^2)$. In practice, we found our training procedure to be insensitive to the choice of these hyper parameters.

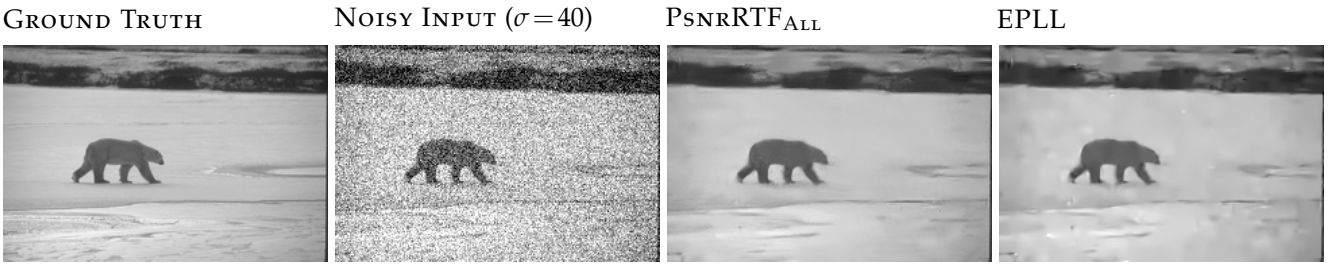


Figure 66: Visual improvement in denoising quality. Our PSNRRTF$_{\text{ALL}}$-system clearly produces more natural restorations than EPLL.

Experiments

We adhere to a strict experimental protocol, using the disjoint training, validation and test splits from the BSDS500 database[294] (with images scaled by a factor of 0.5). In particular, we pay attention to clearly separate the model selection from the final performance evaluation. We perform model selection using the validation set only and evaluate the performance on the test set only once. Given the final results on the test set, we perform a Wilcoxon signed-ranks test[295] testing for the null-hypothesis of equal performance between competing methods.

We consider 12 configurations of our method, based on the combinations of the loss functions we optimize (PSNRRTF, MAERTF, SSIMRTF, NLPLRTF) and three different feature sets: using only the filterbank (RTF$_{\text{PLAIN}}$), the filterbank and the output of BM3D (RTF$_{\text{BM3D}}$), as well as the filterbank, FoE, BM3D, LSSC and EPLL (RTF$_{\text{ALL}}$). Note that the NLPLRTF-systems are trained to minimize the negative log-pseudolikelihood. As for the loss-specific systems, we use joint training of trees and parameters for the NLPLRTF-systems.

Denoising. We perturb the images of the BSDS500 database with additive white Gaussian noise (AWGN), for noise levels $\sigma \in \{20, 30, 40, 50\}$. The results achieved by our system configurations, as well as the strongest competitors, are shown in Table 6. We compare against EPLL, LSSC, BM3D as previously discussed, as well as the most recent FoE release,[296] which optimizes for MSE.

In all cases, an RTF$_{\text{ALL}}$-system trained for the specific loss achieves the best result. In terms of PSNR, the gains over the best published method range from 0.26dB to 0.29dB across the different noise levels. This is a substantial improvement and is clearly visible, as shown in Figure 66.

In many applications, the right trade-off between speed and quality is required. Table 5 shows the average running time of the considered denoising methods on 241×161 pixel images. The improvement of our RTF$_{\text{BM3D}}$-systems over the best published methods (LSSC and EPLL) ranges from 0.07dB to 0.16dB and is statistically significant, yet they run $20\times$ faster.

Observe that the RTF configurations trained for a specific loss perform

	FoE	BM3D	LSSC	EPLL	RTF$_{\text{PLAIN}}$	RTF$_{\text{BM3D}}$	RTF$_{\text{ALL}}$
Running time (s)	1,063	0.9	172	38	0.7	1.6	1,275
PSNR ($\sigma = 30$)	26.81	27.32	27.39	27.44	26.97	27.58	27.72

Table 5: **Typical running time of denoising methods**, for a single natural image (241×161 pixels) on an 8-core Intel Xeon machine (2.4GHz), along with their performance in PSNR. RTF using BM3D as a feature (RTF$_{\text{BM3D}}$) offers the best trade-off between denoising performance and computational cost, as it is better than the strongest competitor (EPLL), yet about twenty times faster.

[294] Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Countour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011

[295] Janez Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30, 2006

[296] Uwe Schmidt, Qi Gao, and Stefan Roth. A Generative Perspective on MRFs in Low-Level Vision. In *Computer Vision and Pattern Recognition (CVPR)*, 2010

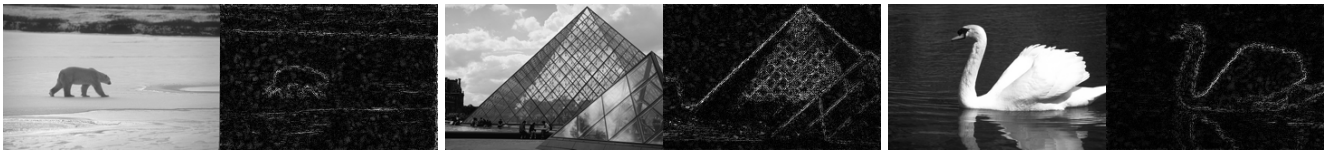


Figure 67: **What has our model learned about images?** On the test set we visualize the original image and the difference image between our best method, PsnrRTF$_{\text{All}}$, and the uniform average of our competitors' predictions (UniformAvg). One can clearly see structure in the difference: our model has learned to refine smooth areas (left), texture patterns (middle), and edges (right).

much better than the NlplRTF-systems. The impressive difference between PsnrRTF$_{\text{Plain}}$ and NlplRTF$_{\text{Plain}}$ ranges from 0.31db to 0.39db across the noise levels. This gap narrows as more powerful features are added to the models, but it remains statistically significant. On the other hand, training of NlplRTF-systems is typically faster than that of the loss-specific models (for example 22h for NlplRTF$_{\text{Bm3d}}$ versus 35h for PsnrRTF$_{\text{Bm3d}}$), and it supports subsampling of pixels both for parameter estimation and node splitting, while subsampling is only possible for the latter in our approach.

The gains of our loss-specific RTF models, both over state-of-the-art denoising methods and NlplRTF, are even more pronounced in terms of MAE and SSIM. Note that it is not at all apparent how these systems could be made to take into account these measures.

A natural question is whether the gains of our approach simply stem from averaging of strong base methods. This is not the case—in Table 6, we show the performance achieved by averaging the predictions of our competitors uniformly (UniformAvg). Our RTF$_{\text{All}}$-systems outperform this naïve strategy by a wide margin. The difference is statistically significant and clearly visible, cf. Figure 67.

Table 6: Denoising test set results for natural images. We compare state-of-the-art competitors to configurations of our method (RTF). For each measure, the result of the strongest competitor is printed in blue, and the best RTF result is printed in green. The gain of our method is **statistically significant** as per Wilcoxon signed-ranks test (with $p < 10^{-5}$ for each blue-green pair in each column). In all cases the RTF trained for the corresponding loss achieves the best result.

Method	**PSNR** (↑ better)				**MAE** (↓ better)				**SSIM** (↑ better)			
σ	20	30	40	50	20	30	40	50	20	30	40	50
Input	22.11	18.59	16.09	14.15	15.96	23.93	31.91	39.89	0.541	0.401	0.307	0.242
FoE	28.87	26.81	25.45	24.47	6.79	8.56	10.03	11.24	0.848	0.776	0.712	0.660
BM3D	29.25	27.32	25.98	25.09	6.40	7.95	9.25	10.22	0.855	0.793	0.741	0.699
LSSC	29.40	27.39	26.08	25.09	6.39	7.96	9.23	10.33	0.861	0.799	0.745	0.700
EPLL	29.38	27.44	26.17	25.22	6.37	7.90	9.12	10.17	0.864	0.800	0.747	0.703
UniformAvg	29.47	27.50	26.21	25.25	6.30	7.84	9.08	10.12	0.863	0.802	0.749	0.705
PsnrRTF$_{\text{Plain}}$	28.95	26.97	25.71	24.76	6.78	8.44	9.72	10.85	0.840	0.771	0.716	0.666
PsnrRTF$_{\text{Bm3d}}$	29.52	27.58	26.24	25.38	6.23	7.73	8.99	9.92	0.863	0.803	0.750	0.711
PsnrRTF$_{\text{All}}$	**29.67**	**27.72**	**26.43**	**25.51**	6.14	7.62	8.80	9.78	0.868	0.809	0.758	0.717
MaeRTF$_{\text{Plain}}$	28.92	26.94	25.69	24.75	6.78	8.43	9.71	10.81	0.840	0.771	0.715	0.669
MaeRTF$_{\text{Bm3d}}$	29.53	27.58	26.22	25.36	6.21	7.71	8.96	9.88	0.863	0.803	0.750	0.711
MaeRTF$_{\text{All}}$	29.67	27.72	26.43	25.50	**6.12**	**7.59**	**8.77**	**9.74**	0.867	0.808	0.758	0.717
SsimRTF$_{\text{Plain}}$	28.49	26.55	25.31	24.41	7.17	8.92	10.23	11.34	0.844	0.778	0.721	0.676
SsimRTF$_{\text{Bm3d}}$	29.17	27.13	25.69	24.85	6.60	8.31	9.80	10.79	0.868	0.809	0.757	0.719
SsimRTF$_{\text{All}}$	29.23	27.14	25.67	24.75	6.60	8.39	9.96	11.06	**0.872**	**0.815**	**0.766**	**0.726**
NlplRTF$_{\text{Plain}}$	28.61	26.66	25.32	24.42	7.09	8.80	10.28	11.37	0.828	0.758	0.694	0.653
NlplRTF$_{\text{Bm3d}}$	29.43	27.44	26.10	25.21	6.32	7.88	9.16	10.13	0.861	0.799	0.747	0.708
NlplRTF$_{\text{All}}$	29.60	27.64	26.34	25.40	6.20	7.71	8.92	9.93	0.866	0.806	0.755	0.714

Removal of blocking and ringing artefacts. To demonstrate once more that our approach is very flexible and can be applied to numerous low-level vision and imaging problems, we distort the images of the BSDS500 database by JPEG blocking artefacts. We use the JPEG quality settings 10, 20, 30 and 40 of the Matlab JPEG encoder. Again, we compare the loss-specific system configurations to maximum pseudolikelihood estimation (NlplRTF),

Figure 68: **Improvement in JPEG deblocking** (quality 10): SA-DCT fails to remove the blocking artefacts in the sky while our MaeRTF$_{\text{SADCT}}$-system succeeds.

as well as the state-of-the-art deblocking method SA-DCT.[297] We consider configurations of our system that use only the filterbank (RTF$_{\text{PLAIN}}$), as well as those that include SA-DCT as a base method (RTF$_{\text{SADCT}}$).

Again, loss-specific training of RTFs achieves the best results, as shown in Table 7. The gains over SA-DCT are statistically significant and clearly visible, as demonstrated in Figure 68. The PSNR and MAE measures are strongly correlated in this task, so there is little difference between PsnrRTF and MaeRTF, but SsimRTF achieves better results in terms of the loss it optimizes.

It is also interesting to note that the "standalone" RTF$_{\text{PLAIN}}$-systems perform extremely competitively. Unlike denoising, there is not much to be gained by incorporating the predictions of SA-DCT as a base method, and it is encouraging to see that our simple feature engineering is already sufficient to beat a state-of-the-art approach.

[297] Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images. *IEEE Transactions on Image Processing*, 16(5):1395–1411, 2007

Method	**PSNR** (↑ better))				**MAE** (↓ better)				**SSIM** (↑ better)			
quality	10	20	30	40	10	20	30	40	10	20	30	40
Input	26.62	28.80	30.08	31.01	8.64	6.64	5.70	5.11	0.790	0.868	0.900	0.918
SA-DCT	27.44	29.48	30.70	31.58	7.67	6.00	5.20	4.69	0.810	0.880	0.909	0.926
PsnrRTF$_{\text{PLAIN}}$	27.66	29.84	31.15	32.10	7.49	5.78	4.95	4.44	0.817	0.886	0.914	0.930
PsnrRTF$_{\text{SADCT}}$	27.70	29.86	**31.17**	32.12	7.43	5.75	4.94	4.42	0.819	0.887	0.915	0.931
MaeRTF$_{\text{PLAIN}}$	27.66	29.83	31.16	32.10	7.46	5.77	4.94	4.43	0.817	0.886	0.914	0.930
MaeRTF$_{\text{SADCT}}$	**27.71**	**29.87**	31.17	**32.13**	**7.40**	**5.73**	**4.93**	**4.41**	0.818	0.887	0.915	0.930
SsimRTF$_{\text{PLAIN}}$	27.18	29.47	30.81	31.80	8.07	6.12	5.23	4.66	0.823	0.889	0.916	**0.932**
SsimRTF$_{\text{SADCT}}$	27.25	29.49	30.82	31.82	7.97	6.10	5.22	4.64	**0.824**	**0.890**	**0.917**	0.932
NlplRTF$_{\text{PLAIN}}$	27.50	29.69	31.01	31.96	7.64	5.90	5.05	4.53	0.813	0.883	0.913	0.928
NlplRTF$_{\text{SADCT}}$	27.61	29.76	31.06	32.00	7.52	5.84	5.01	4.49	0.816	0.885	0.913	0.929

Table 7: JPEG deblocking results for natural images. We compare SA-DCT, a state-of-the-art deblocking method, to configurations of our method (RTF). The best RTF result is printed in green. Again, **statistically significant** gains are printed in **bold font**.

Removal of structured noise. We simulate synthetic dust artifacts as follows. For each image, we sample a random number of dust particles (Poisson-distributed with $\lambda = 20$), and then for each particle we sample a position uniformly at random on the image plane. Each dust particle decreases the image intensity according to a fixed 2D Gaussian-shaped function with scaling of $s = 5$ (small dust) or $s = 20$ (large dust) pixels. Our image restoration framework is highly capable of recovering the images, even for the large-dust case (see Figure 69). We emphasize that none of the other denoising methods described in this paper can handle the structured noise present in the corrupted images. In fact, when applying a conventional denoising method such as BM3D to the noisy input, the algorithm not only fails to remove the dust artifacts, but even worse, it blurs the uncorrupted parts of the image.

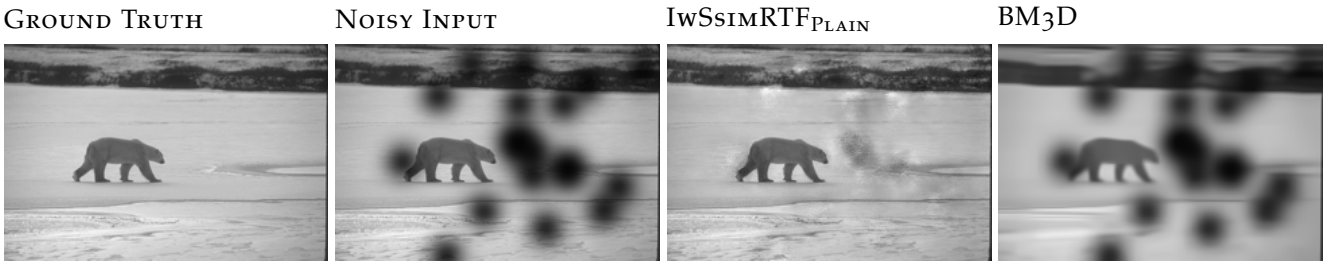


Figure 69: **No real competitor**: Common denoising algorithms fail on the structured noise dataset since the AWGN assumption is violated. While our method is surprisingly capable of restoring the corrupted image regions, BM3D fails to remove the noise, and moreover blurs the uncorrupted parts.

Conclusions and Future Work

In this final part of the thesis, we introduced a Gaussian conditional random field model that is suitable both for discrete and continuous structured prediction tasks. Already the basic parameterization we proposed extends on existing Gaussian CRFs by allowing to freely learn the shape of the local precision matrices of factors from training data. Furthermore, we demonstrated how the expressive power of the model can be further extended by letting the factor energies depend on the observed input in a non-parametric manner, via regression trees.

We devised two efficient training routines for estimating the parameters of such conditional random fields, or learning the regression trees determining the factors, in the case of the non-parametric model class. The first approach is based on maximizing the pseudo-likelihood of the training data and is particularly convenient from a computational perspective: The objective function is convex, and since it decomposes over all variables in the training data, computation of the objective function can be further sped up either through parallelization or by sub-sampling from the variables within each training example. The second training approach we introduced directly seeks to minimize the empirical risk of the model with respect to a user-specified, differentiable loss function. This approach has the advantage of being robust even in the presence of model misspecification, since the quality of the predictions obtained from the model are directly assessed in terms of the quality measure that is eventually applied to test instances. This way, the model parameters are directly chosen so as to obtain the best possible predictions within the restricted model class.

We first evaluated our models on specifically constructed benchmark tasks, demonstrating that in several cases, their expressive power is indeed comparable to discrete models. We also demonstrated the great flexibility afforded by being able to freely mix discrete and continuous variables.

In order to confirm the practical relevance of our method, we then proposed a framework for image restoration, based on three ideas. First, non-parametric regression tree fields as a flexible representation. Second, loss-specific joint training, selecting all model aspects to optimize a task-specific losses. Third, making efficient use of existing restoration methods, combining and improving their predictions. All three ideas *together* produce a new state-of-the-art in image denoising and JPEG deblocking. Importantly, we leveraged the work that has been invested into specialized methods for these tasks by incorporating their predictions into our field model, which

makes it future-proof and applicable to a wide variety of tasks.

Is image denoising solved? We believe it is not solved yet, because common performance measures (PSNR and SSIM) are just a proxy for the perceptual quality. With our model we can efficiently optimize for a given measure, and by analyzing the loss-specific predictions, we hope that in the future this will provide insight into the remaining shortcomings of measures such as SSIM. After all, we would argue that image denoising is as much about the perceived quality by a human observer as it is about the natural image statistics.

Aside from potential improvements in image restoration, we believe that the methods developed in this part of the thesis allow both for exciting new applications, as well as for a number of algorithmic improvements in future work. As far as applications are concerned, low-level computer vision tasks such as image segmentation or optical flow seem to be obvious candidates. Our encouraging results for the joint detection and registration task furthermore suggest that our model could be gainfully employed for human pose estimation, a topic that has received significant attention in recent years.

On the algorithmic side, it will be beneficial to work on improved training routines. While the direct risk minimization approach seems to yield superior results over pseudolikelihood estimation, it is slightly worrisome that the objective function is not convex in the model parameters. It will be an interesting challenge to develop methods that can gainfully incorporate loss functions and are nonetheless convex. Moreover, it would be useful to gain an improved theoretical understanding of the decrease in expressiveness incurred by the restriction to positive-definite factor potentials, thereby characterizing the tightness of our approximation for discrete labeling tasks and setting the ground for future improvements.

In any case, perhaps the main strength of the methods developed in this thesis is their wide applicability and versatility, allowing for novel applications in a variety of different fields. It will certainly be exciting to see them put to use by the scientific community.

Bibliography

[1] Oisin Mac Aodha, Gabriel J. Brostow, and Marc Pollefeys. Segmenting Video into Classes of Algorithm-Suitability. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[2] Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Countour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

[3] Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, Classification, and Risk Bounds. *Journal of the American Statistical Association*, 101(473):138–156, March 2006.

[4] Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

[5] Dhruv Batra, Sebastian Nowozin, and Pushmeet Kohli. A Local Primal-Dual Gap based Separation Algorithm. In *Artificial Intelligence and Statistics (AISTATS)*, 2011.

[6] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.

[7] Dimitri P. Bertsekas, Angelia Nedic, and Asuman E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.

[8] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[9] Julian Besag. Statistical Analysis of Non-Lattice Data. *The Statistician*, 24(3):179—-195, 1975.

[10] Julian Besag. Efficiency of pseudolikelihood estimation for simple Gaussian fields. *Biometrica*, 64(3):616–618, 1977.

[11] Ernesto G. Birgin, José M. Martínez, and Marcos Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000.

[12] Maximilian Bisani and Hermann Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, 2008.

[13] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

[14] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[15] Eran Borenstein, Eitan Sharon, and Shimon Ullmann. Combining top-down and bottom-up segmentation. In *IEEE Workshop on Perceptual Organization in Computer Vision*, June 2004.

[16] Léon Bottou and Olivier Bousquet. The Tradeoffs of Large Scale Learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[17] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[18] Leo Breiman, Jerome Friedman, Charles J. Stone, and R. A. Olshen. *Classification and regression trees*. Chapman and Hall/CRC, 1984.

[19] Richard H. Byrd, Jorge Nocedal, and Robert B. Schnabel. Representations of quasi-Newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1):129–156, 1994.

[20] Venkat Chandrasekaran, Nathan Srebro, and Prahladh Harsha. Complexity of Inference in Graphical Models. Technical report, 2010.

[21] Trevor Cohn. Efficient Inference in Large Conditional Random Fields. In *17th European Conference on Machine Learning (ECML)*, pages 606–613, 2006.

[22] Michael Collins, Amir Globerson, Terry Koo, Xavier Carreras, and Peter L. Bartlett. Exponentiated Gradient Algorithms for Conditional Random Fields and Max-Margin Markov Networks. *Journal of Machine Learning Research*, 9:1775–1822, 2008.

[23] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic networks and expert systems*. Statistics for Engineering and Information Science. Springer Verlag, 1999.

[24] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[25] Hal Daumé III. *Practical Structured Learning Techniques for Natural Language Processing*. PhD thesis, University of Southern California, 2006.

[26] Janez Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[27] Michel M. Deza and Monique Laurent. *Geometry of cuts and metric embeddings*. Springer Verlag, 1997.

[28] Justin Domke. Dual Decomposition for Marginal Inference. In *Conference on Artificial Intelligence (AAAI)*, 2011.

[29] Justin Domke. Parameter learning with truncated message-passing. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2937–2943, 2011.

[30] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.

[31] Francisco Estrada, David Fleet, and Allan Jepson. Stochastic Image Denoising. In *British Machine Vision Conference (BMVC)*, 2009.

[32] Thomas Finley and Thorsten Joachims. Training structural SVMs when exact inference is intractable. In *International Conference on Machine Learning (ICML)*, 2008.

[33] Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images. *IEEE Transactions on Image Processing*, 16(5):1395–1411, 2007.

[34] Varun Ganapathi, David Vickrey, John Duchi, and Daphne Koller. Constrained Approximate Maximum Entropy Learning. In *Uncertainty in Artificial Intelligence (UAI)*, 2008.

[35] Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Robust trajectory-space TV-L 1 optical flow for non-rigid sequences. In *8th international conference on Energy minimization methods in computer vision and pattern recognition (EMMCVPR)*, pages 300–314, 2011.

[36] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distribution and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[37] Arthur M. Geoffrion. Duality in Nonlinear Programming: A Simplified Applications-Oriented Development. *SIAM Review*, 13(1):1–37, 1971.

[38] Kevin Gimpel and Noah A. Smith. Softmax-Margin CRFs: Training Log-Linear Models with Cost Functions. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010.

[39] Ross Girshick, Jamie Shotton, Pushmeet Kohli, Antonio Criminisi, and Andrew Fitzgibbon. Efficient Regression of General-Activity Human Poses from Depth Images. In *International Conference on Computer Vision (ICCV)*, 2011.

[40] Amir Globerson and Tommi Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *Advances in Neural Information Processing Systems*, 2007.

[41] Amir Globerson and Tommi S. Jaakkola. Convergent propagation algorithms via oriented trees. In *Uncertainty in Artificial Intelligence (UAI)*, 2007.

[42] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[43] Joseph Gonzalez, Yucheng Low, Arthur Gretton, and Carlos Guestrin. Parallel Gibbs Sampling: From Colored Fields to Thin Junction Trees. In *Artificial Intelligence and Statistics (AISTATS)*, 2011.

[44] Joseph E. Gonzalez, Yucheng Low, and Carlos Guestrin. Residual splash for optimally parallelizing belief propagation. In *Artificial Intelligence and Statistics (AISTATS)*, 2009.

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

[46] Luigi Grippo, Francesco Lampariello, and Stefano Lucidi. A non monotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, 23:707–716, 1986.

[47] Tamir Hazan and Amnon Shashua. Convergent message-passing algorithms for inference over general graphs with convex free energies. In *Uncertainty in Artificial Intelligence (UAI)*, 2008.

[48] Tamir Hazan and Amnon Shashua. Norm-product belief propagation: Primal-dual message-passing for approximate inference. *IEEE Transactions on Information Theory*, 56(12):6294–6316, 2010.

[49] Tamir Hazan and Rachel Urtasun. A Primal-Dual Message-Passing Algorithm for Approximated Large Scale Structured Prediction. In *Advances in Neural Information Processing Systems*, 2010.

[50] Uri Heinemann and Amir Globerson. What Cannot be Learned with Bethe Approximations. In *Uncertainty in Artificial Intelligence (UAI)*, 2011.

[51] Tom Heskes. Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies. *Journal of Artificial Intelligence Research*, 26:153–190, 2006.

[52] Magnus R. Hestenes and Eduard Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, 49(6), 1952.

[53] Nicholas J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103–118, 1988.

[54] Robert V. Hogg, Allen Craig, and Joseph W. McKean. *Introduction to Mathematical Statistics*. Pearson Education, 2005.

[55] Mark J. Huiskes and Michael S. Lew. The MIR Flickr retrieval evaluation. In *International Conference on Multimedia Information Retrieval (MIR)*, 2008.

[56] Jeremy Jancsary, Johannes Matiasek, and Harald Trost. Revealing the Structure of Medical Dictations with Conditional Random Fields. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

[57] Finn V. Jensen and Frank Jensen. Optimal Junction Trees. In *Uncertainty in Artificial Intelligence (UAI)*, pages 360–366, 1994.

[58] Jason K. Johnson, Danny Bickson, and Danny Dolev. Fixing convergence of Gaussian belief propagation. In *IEEE International Symposium on Information Theory (ISIT)*, 2009.

[59] Jason K. Johnson, Dmitry Malioutov, and Alan S. Willsky. Lagrangian relaxation for MAP estimation in graphical models. In *Allerton Conference on Communication, Control and Computing*, 2007.

[60] Vladimir Jojic, Stephen Gould, and Daphne Koller. Accelerated dual decomposition for MAP inference. In *International Conference on Machine Learning (ICML)*, 2010.

[61] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. Prentice Hall, 1998.

[62] Steven M. Kay. *Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory*. Prentice Hall, 1999.

[63] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[64] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568 – 1583, 2006.

[65] Nikos Komodakis, Nikos Paragios, and Georgios Tziritas. MRF Optimization via Dual Decomposition: Message-Passing Revisited. In *International Conference on Computer Vision (ICCV)*, 2007.

[66] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[67] Alex Kulesza and Fernando Pereira. Structured Learning with Approximate Inference. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[68] Pawan M. Kumar, Vladmir Kolmogorov, and Philip H. S. Torr. An Analysis of Convex Relaxations for MAP Estimation. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[69] Pawan M. Kumar, Philip H. S. Torr, and Andrew Zisserman. Solving Markov Random Fields using Second Order Cone Programming Relaxations. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1045–1052, 2006.

[70] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *International Conference on Machine Learning (ICML)*, 2001.

[71] Steffen. L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50(2):157–224, 1988.

[72] Guy Lebanon and John Lafferty. Boosting and Maximum Likelihood for Exponential Models. In *Advances in Neural Information Processing Systems*, 2001.

[73] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *SIGGRAPH*, 2004.

[74] Anat Levin and Yair Weiss. Learning to Combine Bottom-Up and Top-Down Segmentation. In *European Conference on Computer Vision (ECCV)*, 2006.

[75] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local Sparse Models for Image Restoration. In *International Conference on Computer Vision (ICCV)*, 2009.

[76] André F. T. Martins, Mário A. T. Figueiredo, Pedro M. Q. Aguiar, Noah A. Smith, and Eric P. Xing. An Augmented Lagrangian Approach to Constrained MAP Inference. In *International Conference on Machine Learning (ICML)*, 2011.

[77] André F. T. Martins, Noah A. Smith, and Eric P. Xing. Polyhedral outer approximations with application to natural language parsing. In *International Conference on Machine Learning (ICML)*, 2009.

[78] Talya Meltzer, Amir Globerson, and Yair Weiss. Convergent message passing algorithms - A unifying view. In *Uncertainty in Artificial Intelligence (UAI)*, 2009.

[79] Ofer Meshi and Amir Globerson. An Alternating Direction Method for Dual MAP LP Relaxation. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2011.

[80] Ofer Meshi, Ariel Jaimovich, Amir Globerson, and Nir Friedman. Convexifying the Bethe Free Energy. In *Uncertainty in Artificial Intelligence (UAI)*, 2009.

[81] Ofer Meshi, David Sontag, Tommi Jaakkola, and Amir Globerson. Learning Efficiently with Approximate Inference via Dual Losses. In *International Conference on Machine Learning (ICML)*, 2010.

[82] Thomas P. Minka. Empirical Risk Minimization is an incomplete inductive principle. Technical report, MIT Media Lab, 2000.

[83] Joris M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, 2010.

[84] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In *Uncertainty in Artificial Intelligence (UAI)*, 1999.

[85] Angelia Nedic and Dimitri P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12:109–138, 2001.

[86] Angelia Nedic and Vijay G. Subramanian. Approximately optimal utility maximization. In *IEEE Information Theory Workshop on Networking and Information Theory*, pages 206–210, 2009.

[87] Arkadi S. Nemirovski and Michael J. Todd. Interior-point methods for optimization. *Acta Numerica*, 17:191–234, 2008.

[88] Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980.

[89] Sebastian Nowozin, Carsten Rother, Shai Bagon, Toby Sharp, Bangpeng Yao, and Pushmeet Kohli. Decision tree fields. In *13th International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 2011.

[90] Judea Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *National Conference on Artificial Intelligence (AAAI)*, pages 133–136, 1982.

[91] Patrick Pletscher, Cheng Soon Ong, and Joachim M. Buhmann. Spanning Tree Approximations for Conditional Random Fields. In *Artificial Intelligence and Statistics (AISTATS)*, 2009.

[92] Patrick Pletscher, Cheng Soon Ong, and Joachim M. Buhmann. Entropy and Margin Maximization for Structured Output Learning. In *European Conference on Machine Learning (ECML)*, 2010.

[93] Yuan Qi, Martin Szummer, and Thomas P. Minka. Bayesian Conditional Random Fields. In *Artificial Intelligence and Statistics (AISTATS)*, 2005.

[94] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[95] Pradeep Ravikumar, Alekh Agarwal, and Martin J. Wainwright. Message-passing for Graph-structured Linear Programs: Proximal Methods and Rounding Schemes. *Journal of Machine Learning Research*, 11:1043–1080, 2010.

[96] Pradeep Ravikumar and John Lafferty. Quadratic programming relaxations for metric labelling and Markov random field MAP estimation. In *International Conference on Machine Learning (ICML)*, 2006.

[97] Brian D. Ripley. *Stochastic Simulation*. Wiley-Interscience, 2006.

[98] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 66(6):386–408, 1958.

[99] Dan Roth and Wen-tau Yih. Integer linear programming inference for conditional random fields. In *22nd International conference on Machine learning (ICML)*, pages 736–743, August 2005.

[100] Stefan Roth and Michael J. Black. Fields of Experts. *International Journal of Computer Vision (IJCV)*, 82(2):205–229, 2009.

[101] Michail I. Schlesinger. Syntactic analysis of two-dimensional visual signals in the presence of noise. *Cybernetics and Systems Analysis*, 12(4):612–628, 1976.

[102] Mark Schmidt, Ewout Van den Berg, Michael P. Friedlander, and Kevin Murphy. Optimizing Costly Functions with Simple Constraints: A Limited-Memory Projected Quasi-Newton Algorithm. In *Artificial Intelligence and Statistics (AISTATS)*, 2009.

[103] Uwe Schmidt, Qi Gao, and Stefan Roth. A Generative Perspective on MRFs in Low-Level Vision. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[104] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[105] Nicol N. Schraudolph and Dmitry Kamenetsky. Efficient Exact Inference in Planar Ising Models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1417–1424, 2009.

[106] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1998.

[107] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *International Conference on Machine Learning (ICML)*, 2007.

[108] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation. In *European Conference on Computer Vision (ECCV)*, 2006.

[109] Mathieu Sinn and Pascal Poupart. Asymptotic Theory for Linear-Chain Conditional Random Fields. In *Artificial Intelligence and Statistics (AISTATS)*, pages 679–687, 2011.

[110] Noah A. Smith. *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, 2011.

[111] David Sontag, Amir Globerson, and Tommi Jakkola. Clusters and Coarse Partitions in LP Relaxations. In *Neural Information Processing Systems (NIPS)*, 2008.

[112] David Sontag, Talya Meltzer, Amir Globerson, Yair Weiss, and Tommi Jakkola. Tightening LP Relaxations for MAP using Message Passing. In *Uncertainty in Artificial Intelligence (UAI)*, 2008.

[113] Veselin Stoyanov, Alexander Ropson, and Jason Eisner. Empirical Risk Minimization of Graphical Model Parameters Given Approximate Inference, Decoding, and Model Structure. In *Artificial Intelligence and Statistics (AISTATS)*, pages 725–733, 2011.

[114] Charles Sutton and Andrew McCallum. An Introduction to Conditional Random Fields for Relational Learning. In Lise Getoor, editor, *Introduction to Statistical Relational Learning*, pages 93–128. MIT Press, 2007.

[115] Charles Sutton and Andrew McCallum. Piecewise training for structured prediction. *Machine learning*, 77(2):165–194, 2009.

[116] Marshall Tappen, Ce Liu, Edward Adelson, and William Freeman. Learning Gaussian Conditional Random Fields for Low-Level Vision. In *Computer Vision and Pattern Recognition (CVPR)*, 2007.

[117] Marshall Tappen, Kegan Samuel, Craig Dean, and David Lyle. The Logistic Random Field—A convenient graphical model for learning parameters for MRF-based labeling. In *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[118] Ben Taskar, Vassil Chatalbashev, and Daphne Koller. Learning associative Markov networks. In *International Conference on Machine Learning (ICML)*, 2004.

[119] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-Margin Markov Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[120] Choon Hui Teo, S. V. N. Vishwanathan, Alex Smola, and Quoc V. Le. Bundle Methods for Regularized Risk Minimization. *Journal of Machine Learning Research*, 11:311–365, 2010.

[121] Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 shared task. In *4th Conference on Computational natural language learning (CoNLL)*, September 2000.

[122] Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

[123] S. V. N. Vishwanathan, Nicol N. Schraudolph, Mark W. Schmidt, and Kevin P. Murphy. Accelerated Training of Conditional Random Fields with Stochastic Gradient Methods. In *International Conference on Machine Learning (ICML)*, 2006.

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

[125] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, 51:2313–2335, 2005.

[126] Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky. MAP Estimation via Agreement on Trees: Message-Passing and Linear Programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005.

[127] Martin J. Wainwright and Michael I. Jordan. Variational inference in graphical models: The view from the marginal polytope. In *Allerton Conference on Communication, Control, and Computing*, 2003.

[128] Martin J. Wainwright and Michael I. Jordan. Log-Determinant Relaxation for Approximate Inference in Discrete Graphical Models. *IEEE Transactions on Signal Processing*, 54(6):2099–2109, 2006.

[129] Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

[130] Changyu Wang, Qian Liu, and Xinmin Yang. Convergence properties of non monotone spectral projected gradient methods. *Journal of Computational and Applied Mathematics*, 182(1):51–66, 2005.

[131] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[132] Zhou Wang and Qiang Li. IW-SSIM: Information Content Weighted Structural Similarity Index for Image Quality Assessment. *IEEE Transactions on Image Processing*, 20(5):1185–1198, 2011.

[133] Zhou Wang and Eero P. Simoncelli. Maximum differentiation (MAD) competition: Methodology for comparing computational models of perceptual quantities. *Journal of Vision*, 8(12):1–13, 2008.

[134] Jerod J. Weinman, Lam Tran, and Christopher J. Pal. Efficiently Learning Random Fields for Stereo Vision with Sparse Message Passing. In *European Conference on Computer Vision (ECCV)*, 2008.

[135] Yair Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.

[136] Yair Weiss and William T. Freeman. Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology. *Neural Computation*, 13(10):2173–2200, 2001.

[137] Yair Weiss, Chen Yanover, and Talya Meltzer. MAP Estimation, Linear Programming and Belief Propagation with Convex Free Energies. In *Uncertainty in Artificial Intelligence (UAI)*, 2007.

[138] Halbert White. Maximum Likelihood Estimation of Misspecified Models. *Econometrica*, 50(1):1–25, 1982.

[139] Chen Yanover, Talya Meltzer, and Yair Weiss. Linear Programming Relaxations and Belief Propagation - An Empirical Study. *Journal of Machine Learning Research*, 7:1887–1907, 2006.

[140] Chen Yanover, Ora Schueler-Furman, and Yair Weiss. Minimizing and Learning Energy Functions for Side-Chain Prediction. In *International Conference on Research in Computational Molecular Biology (RECOMB)*, 2007.

[141] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*, chapter 8, pages 239–270. Morgan Kaufmann, 2002.

[142] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms. *IEEE Transactions on Information Theory*, 51:2282–2312, 2004.

[143] Daniel Zoran and Yair Weiss. From Learning Models of Natural Image Patches to Whole Image Restoration. In *International Conference on Computer Vision (ICCV)*, 2011.

Curriculum Vitae

Jeremy Jancsary

Zur Spinnerin 1/3/313
A-1100 Vienna
Austria

☎ 0043 664 7388 1983

✉ jeremy.jancsary@gmail.com

Personal Details

26/09/1981 Born in Au, Vorarlberg, Austria; of Austrian nationality.

Education

2008–present Ph.D. candidate, *Vienna University of Technology*, Vienna, Austria.
Investigating discriminative training of cyclic graphical models.
Advisor: ao.Univ.-Prof. Dr.techn. Gerald Matz

2005–2008 M.Sc., *Vienna University of Technology*, Vienna, Austria; graduated with distinction.
Studies in *Software Engineering & Internet Computing*, with a focus on AI.
Thesis: *Recognizing structure in report transcripts – an approach based on CRFs*
Advisor: ao.Univ.-Prof. Dr.techn. Harald Trost

2001–2005 B.Sc., *Vienna University of Technology*, Vienna, Austria; graduated with distinction.
Studies in *Software & Information Engineering*, Information Engineering branch.
Thesis: *Identification of Authors of Classic Literature*
Advisor: ao.Univ.-Prof. Dr.techn. Harald Trost

1996–2001 Diploma, *Handelsakademie Feldkirch*, Feldkirch, Austria; graduated with distinction.
Attended higher-level secondary commercial college, *Controlling* branch.

Professional Experience

9/2012– Post-Doctoral Researcher, *Microsoft Research*, Cambridge, UK.

3/2006–8/2012 Researcher, *Austrian Research Institute for Artificial Intelligence*, Vienna, Austria.

8/2011–2/2012 Contractor, *Microsoft Research*, Cambridge, UK.

4/2011–7/2011 Research Intern, *Microsoft Research*, Cambridge, UK.

2004–2006 Software Engineer, *RISE/Vienna University of Technology*, Vienna, Austria.

2003–2004 Teaching Assistant, *Vienna University of Technology*, Vienna, Austria.

1999–2001 Area Manager (Controlling & Budget), *Verein Kulturbad*, Feldkirch, Austria.

Professional Service & Memberships

conferences — Program chair of the *11th Conference on Natural Language Processing (KONVENS)*, to be held in September 2012 in Vienna, Austria.

workshops — Co-organizer of the *1st Workshop on Algorithms and Resources for Modeling of Dialects and Language Varieties*, held in conjunction with EMNLP 2011, Edinburgh, Scotland.

reviewing — Referee for *Applied Artificial Intelligence*, *AI Communications*, *Neural Information Processing Systems (NIPS)*, and the *10th Conference on Natural Language Processing (KONVENS)*.

memberships — Member of the *Austrian Society for Artificial Intelligence (ÖGAI)*, the *Association for Computational Linguistics (ACL)*, the *Association for Computing Machinery (ACM)*, and the *Institute of Electrical and Electronics Engineers (IEEE)*.

Publications

edited proceedings — J. Jancsary, F. Neubarth, and H. Trost (eds). Proceedings of the EMNLP 2011 Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties. Association for Computational Linguistics (ACL), New Brunswick, NJ, 2011.

journals — J. Jancsary, F. Neubarth, S. Schreitter, and H. Trost. Multi-Faceted Analysis of News Articles for Intelligent User- and Context-Sensitive Presentation. *Journal article under preparation*, 2012.

S. Petrik, C. Drexel, J. Jancsary, A. Klein, G. Kubin, J. Matiasek, F. Pernkopf, and H. Trost. Semantic and Phonetic Automatic Reconstruction of Medical Dictations. *Computer Speech & Language*, 25(2):363–385, 2011.

conferences — J. Jancsary, S. Nowozin, and C. Rother. Loss-Specific Training of Non-Parametric Image Restoration Models: A New State of the Art. In *12th European Conference on Computer Vision (ECCV)*, Florence, Italy, 2012.

J. Jancsary, S. Nowozin, T. Sharp, and C. Rother. Regression Tree Fields – An Efficient, Non-Parametric Approach to Image Labeling Problems. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA.

J. Jancsary and G. Matz. Convergent Decomposition Solvers for Tree-reweighted Free Energies. In *14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Ft. Lauderdale, FL, USA, 2011.

J. Jancsary, F. Neubarth, and H. Trost. Towards Context-Aware Personalization and a Broad Perspective on the Semantics of News Articles. In *4th ACM Conference on Recommender Systems (RECSYS)*, Barcelona, Spain, 2010.

J. Jancsary, J. Matiasek, and H. Trost. Revealing the Structure of Medical Dictations with Conditional Random Fields. In *2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Honolulu, HI, USA, 2008.

M. Huber, J. Jancsary, A. Klein, and H. Trost. Mismatch interpretation by semantics-driven alignment. In *8th Conference on Natural Language Processing (KONVENS)*, Konstanz, Germany, 2006.

peer-reviewed workshops

J. Jancsary, F. Neubarth, S. Schreitter, and H. Trost. Towards a Context-Sensitive Online Newspaper. In *IUI 2011 Workshop on Context-awareness in Retrieval and Recommendation*, Palo Alto, CA, USA, 2011.

J. Jancsary, G. Matz, and H. Trost. An Incremental Subgradient Algorithm for Approximate MAP Estimation in Graphical Models. In *NIPS 2010 Workshop on Optimization for Machine Learning*, Whistler, BC, Canada, 2010.

J. Matiasek, J. Jancsary, A. Klein, and H. Trost. Identifying Segment Topics in Medical Dictations. In *EACL 2009 Workshop on Semantic Representation of Spoken Language*, Athens, Greece, 2009.

J. Jancsary, A. Klein, J. Matiasek, and H. Trost. Semantics-Based Automatic Literal Reconstruction of Dictations. In *CAEPIA 2007 Workshop on Semantic Representation of Spoken Language*, Salamanca, Spain, 2007.

Invited Talks

J. Jancsary. Regression Tree Fields – A Practical and Effective Framework For Structured Learning and Prediction. *Microsoft Research*, Cambridge, UK, April 2012.

J. Jancsary. Piecewise Training Re-visited: The Tightened Piecewise Estimator. *IST Austria*, Klosterneuburg, Austria, December 2011.

J. Jancsary. Learning with Tree-Reweighted Upper Bounds. *IST Austria*, Klosterneuburg, Austria, January 2011.

Software

J. Jancsary. VieCRF: A Fast Toolkit for Factorial Conditional Random Fields, 2010. Available at `http://www.ofai.at/~jeremy.jancsary/`.

Index