Learning and Evaluating Markov Random Fields for Natural Images

Master's thesis by Uwe Schmidt
February 2010

TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Computer Science
Interactive Graphics Systems Group

Learning and Evaluating Markov Random Fields for Natural Images
Lernen und Evaluieren von Markov Random Fields für Natürliche Bilder

vorgelegte Masterarbeit von Uwe Schmidt

Fachbereich Informatik
Fachgebiet Graphisch-Interaktive Systeme
Prof. Stefan Roth, PhD

Tag der Einreichung: 12. Februar 2010

Erklärung zur Masterarbeit

Hiermit versichere ich die vorliegende Masterarbeit ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 12. Februar 2010

(U. Schmidt)

Essentially, all models are wrong, but some are useful.

George E. P. Box

Abstract

Many problems of computer vision are (mathematically) ill-posed in the sense that there are many solutions; those problems are therefore in need of some form of regularization that guarantees a sensible and unique solution. This is also true for problems in low-level vision, which are addressing visual information at a basic level (e.g. pixels of an image), and are of interest for this work.

Markov Random Fields (MRFs) are widely used probabilistic models of "prior knowledge", which are used for regularization in a variety of computer vision problems, in particular those in low-level vision; we focus on generic MRF models for natural images and apply them to image restoration tasks. Learning MRFs from training data with a popular approach like the generic maximum likelihood (ML) method is often difficult, however, because of its computational complexity and the requirement to draw samples from the MRF. Because of these difficulties, a number of alternative learning methods have been proposed over the years, of which score matching (SM) is a promising one that has not been properly explored in the context of MRF models.

Armed with an efficient sampler, we propose a flexible MRF model for natural images that we train under various circumstances. Instead of evaluating MRFs using a specific application and inference technique, as is common in the literature, we compare them in a fully application-neutral setting by means of their generative properties, i.e. how well they capture the statistics of natural images. We find that estimation with score matching is problematic for MRF image priors, and tentatively attribute this to the use of heavy-tailed potentials, which are required for MRF models to match the statistics of natural images. Hence, we also take a different route and exploit our efficient sampler to improve learning with contrastive divergence (CD), an efficient learning method closely related to ML, which has successfully been applied to MRF parameter learning in the past. We let score matching and contrastive divergence compete to learn the parameters of MRFs, which enables us to better understand the weaknesses and strengths of both methods.

Using contrastive divergence, we learn MRFs that capture the statistics of natural images very well. We additionally find that popular MRF models from the literature exhibit poor generative properties, despite their good application performance in the context of maximum a-posteriori (MAP) estimation; they surprisingly even outperform our good generative models. By computing the posterior mean (MMSE) using sampling, we are able to achieve excellent results in image restoration tasks with our application-neutral generative MRFs, that can even compete with application-specific discriminative approaches.

Zusammenfassung

Viele Probleme des Maschinellen Sehens sind (mathematisch) nicht korrekt gestellt in dem Sinne, dass es meist viele Lösungen gibt; solche Probleme benötigen deshalb eine gewisse Form der Regularisierung, die eine vernünftige und eindeutige Lösung garantiert. Das gilt auch für Probleme im Bereich "Low-Level Vision", die sich mit visuellen Information auf einem niedrigen Level befassen (z.B. Pixel eines Bildes) und von Belang für diese Arbeit sind.

Markov Random Fields (MRFs) sind weithin genutzte probabilistische Modelle von "Vorwissen", die für Regularisierung in vielfältigen Problemen des Maschinellen Sehens verwendet werden, insbesondere jene in "Low-Level Vision"; wir konzentrieren uns auf generische MRF-Modelle für natürliche Bilder und wenden diese auf Probleme der Bildwiederherstellung an. MRFs mit beliebten Ansätzen wie der allgemeinen Maximum-Likelihood (ML) Methode von Trainingsdaten zu lernen ist jedoch oft schwer, angesichts des Rechenaufwands und der Anforderung Stichproben des MRF-Modells zu produzieren ("Sampling"). Diese Schwierigkeiten haben dazu geführt dass im Laufe der Jahre einige alternative

Lernverfahren vorgeschlagen wurden, von denen Score Matching (SM) ein vielversprechendes ist, das jedoch im Kontext von MRFs nicht gründlich erforscht wurde.

Ausgerüstet mit einem effizienten Sampler schlagen wir ein flexibles MRF-Modell für natürliche Bilder vor, welches wir unter verschiedenen Gegebenheiten trainieren. Anstatt MRFs anhand einer Kombination von spezifischer Anwendung und Inferenzverfahren zu bewerten, wie in der Literatur üblich, vergleichen wir sie in einem vollkommen anwendungsneutralem Rahmen durch ihre generativen Eigenschaften, d.h. wie gut sie die statistischen Eigenschaften von natürlichen Bildern modellieren.

Wir stellen fest dass Score Matching problematisch für das Lernen von MRF-Modellen von Bildern ist, und schreiben dies vorläufig der Verwendung von Heavy-tailed-Verteilungen zu, welche benötigt werden um die statistischen Eigenschaften von natürlichen Bildern mit MRFs zu modellieren. Deshalb schlagen wir auch einen anderen Weg ein und verwenden unseren effizienten Sampler um das Lernen mit Contrastive Divergence (CD) zu verbessern, welches ein effizientes Lernverfahren ähnlich der ML-Methode ist und bereits in der Vergangenheit erfolgreich zum Lernen von MRFs verwendet wurde. Wir lassen Score Matching und Contrastive Divergence gegeneinander antreten die Parameter von MRFs zu lernen, was uns ermöglicht die Stärken und Schwächen beider Verfahren besser zu verstehen.

Mittels Contrastive Divergence lernen wir MRFs welche die statistischen Eigenschaften von natürlichen Bildern sehr gut modellieren. Wir stellen zudem fest dass populäre MRF-Modelle aus der Literatur schlechte generative Eigenschaften aufweisen, ungeachtet ihrer guten Anwendungs-Ergebnisse im Zusammenhang mit Maximum-A-Posteriori (MAP) Schätzung; sie sind erstaunlicherweise sogar besser als unsere guten generativen Modelle. Durch Berechnung des Erwartungswertes der A-posteriori-Verteilung (MMSE) mittels Sampling erzielen unsere anwendungsneutralen generativen MRFs exzellente Resultate in Bildwiederherstellungs-Aufgaben und können sogar mit anwendungsspezifischen diskriminativen Ansätzen konkurrieren.

Acknowledgments

This work is the result of research carried out under the supervision of Stefan Roth, whom I'm very grateful for his support and advice; I learned a great deal about doing research from him. Parts of this work have been submitted in similar form together with Qi Gao and Stefan Roth to the *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, California, June 13–18, 2010.

I am thankful to Yair Weiss for sharing his ideas on the efficient Gibbs sampler with my supervisor, which made this work practical. I furthermore appreciate the detailed results that Kegan Samuel and Marshall Tappen shared with Stefan Roth. My thanks also go to Siwei Lyu for discussing some of his work on score matching with me. I am grateful to the Franziskanergymnasium Kreuzburg in Großkrotzenburg for letting me work at their library, where parts of this work have been created.

Last but not least, I'm deeply indebted to my parents for exposing me to computers at an early age; they heavily invested in my education and let me follow my interests.

Contents

List of Figures

List of Tables

1 Introduction

Computer vision addresses problems at various levels of abstraction, which range from extracting information at a pixel-basis, called *low-level vision*, up to semantic understanding of an image, called *high-level vision*. Many problems are (mathematically) ill-posed in the sense that there is no unique solution. An intuitive example is the problem of *image inpainting*, where the goal is to restore missing pixels of an image. Certainly, there are many possible solutions and no objective measure to assess which one is best if the original uncorrupted image is not available – which is the case in a real-world application. Hence, we need to impose additional constraints to guarantee a unique solution, which is commonly referred to as *regularization*. Regularization can be thought of as using prior domain knowledge to solve a particular ill-posed problem. To pick up the above example, general knowledge about "good" images can be used to assess possible image restorations.

Markov Random Fields (MRFs) are widely used probabilistic models for regularization, since they allow to integrate prior knowledge of images and scenes. Due to their generic nature, they have found widespread use across low-level vision, and in particular *image restoration* [Geman and Geman, 1984; Roth and Black, 2009; Zhu and Mumford, 1997], which is the focus of this work. While much of this work should apply to other areas of low-level vision, we focus on generic MRF models for *natural images* and apply them to image restoration tasks.

The number of parameters of MRF models typically grows as they become increasingly more sophisticated. Hence, tweaking those parameters by hand is sub-optimal, although sometimes still practiced due to the fact that probabilistic parameter learning in MRFs is often complicated and computationally demanding. This stems from MRFs usually being *unnormalized statistical models*, i.e. the probability density function (pdf) defined by the MRF is only known up to a normalization constant. *Maximum Likelihood* (ML), probably the most common and popular method of probabilistic parameter estimation, requires the pdf to be normalized, i.e. evaluation of the so-called partition function that is mostly intractable in MRF models. Hence, one usually has to resort to approximative inference, often requiring computationally demanding sampling techniques, such as *Markov chain Monte Carlo* (MCMC).

Because of these difficulties, a number of alternative learning methods have been proposed over the years (see Li [2009] for an overview). Of particular interest is *contrastive divergence* (CD) [Hinton, 2002], a learning method closely related to ML, although much more efficient, which has successfully been applied to MRF parameter learning [Roth and Black, 2009]. Despite the success of contrastive divergence, learning is far from perfect and sampling still remains a bottleneck in practice. For example, Roth and Black [2009] employed a hybrid Monte Carlo sampler that only allowed them to train the MRF on rather small image patches. Hence, better learning methods for MRFs are still desirable and suitable candidates should be explored.

One of these candidates is *score matching* (SM), a novel general-purpose estimation method proposed by Hyvärinen [2005], that does not require the model density to be normalized. It works by "minimizing the expected squared distance between the gradient of the log-density given by the model and the gradient of the log-density of the observed data" [Hyvärinen, 2005]. The method is interesting because the author proves the surprising result that the objective function can be re-written to not require evaluation of the gradient of the (unknown) data log-density. Furthermore, subsequent work [Hyvärinen, 2008] suggests that estimation by score matching is optimal and thus preferable to ML for signal restoration under certain circumstances (image denoising in the MAP-MRF framework [Li, 2009] seems to qualify to some extent). Köster et al. [2009] used SM for MRF parameter estimation, however in a rather restricted way that does not allow capturing the statistics of the data, as we will show in Chapter 5. A more

thorough investigation in MRF parameter learning via SM is lacking, which is one of the contributions of this work.

An MRF is often defined in terms of univariate (so-called) *potential functions* that model the responses to a bank of linear filters. The study of natural images [Srivastava et al., 2003] suggests that these potentials need to be *heavy-tailed* to allow MRFs to capture the statistics of natural images. Preliminary experiments with (univariate) Student-t distributions, used as potentials by Roth and Black [2009], suggested that SM does not work well for heavy-tailed distributions, and can especially run into problems if the model parameters allow for "unbounded peakedness" (Chapter 4). This supported our decision to use *Gaussian Scale Mixture* (GSM) models [Portilla et al., 2003] instead, which are like regular Gaussian mixture models that share a common mean value (0 here). Although GSMs are slightly more complicated to work with, they allow for a very flexible model with more control over the possible shapes of the distribution.

Moreover, GSM-based MRFs admit a very efficient sampling procedure (Section 3.1) which allows us to compare the learned models in terms of their *generative properties*, i.e. how well the learned MRFs capture the statistics of natural images. By doing this, we adopt the strategy of Zhu and Mumford [1997] who evaluated their image priors from model samples already over 10 years ago. Ever since, the statistical properties of MRFs have rarely been evaluated. Instead, model evaluation usually happens in the context of a particular application and inference method, e.g. *image denoising* using gradient-based methods in case of image priors [Roth and Black, 2009]. The difficulty of computing probabilistic properties of MRFs may be the culprit; although generic samplers are often applicable, they are mostly slow and inefficient.

We find that MRFs trained with score matching do not match the statistics of natural images under realistic circumstances. Our observations in univariate experiments extend to MRFs and suggest that SM is rather unsuitable for heavy-tailed potential functions, as required for MRF models of images. Hence, we also take a different route and exploit the efficient sampler to improve parameter learning with contrastive divergence. We let both estimators, score matching and contrastive divergence, compete under various circumstances to learn the parameters of MRFs (Chapter 5); this enables us to better understand the weaknesses and strengths of both methods. For instance, we find that our efficient sampler allows CD to be actually faster than SM, although SM was proposed as a computationally efficient alternative to learning methods that rely on costly MCMC sampling techniques.

Using contrastive divergence, we learn MRFs with good generative properties that exhibit *heavier-tailed potentials* than have previously been used. This is the first time, as far as we are aware, that it has been shown which potential shapes are required to capture the statistics of natural images in pairwise and high-order MRFs with learned filters.

Furthermore, we highlight the issue of boundary pixels in (high-order) MRFs, and their adverse effects on parameter learning and model analysis through sampling. We are able to alleviate this problem to some extent by adopting a conditional sampling strategy [Norouzi et al., 2009].

In the last part of this work, we show that popular MRF models from the literature exhibit poor generative properties, despite their good application performance in the context of *maximum a-posteriori* (MAP) estimation; they surprisingly even outperform our good generative models in an image denoising application. SM-trained MRFs also do not perform better in MAP-based image denoising, despite theoretical properties that would suggest so [Hyvärinen, 2008].

We demonstrate that it is feasible to apply our efficient sampler to compute the posterior mean, or *Bayesian minimum mean squared error estimate* (MMSE), in image denoising and inpainting. The MMSE estimate for our good generative models not only substantially outperforms MAP, but also solves some of its problems that have been pointed out in a number of recent theoretical and empirical results [Levin et al., 2009; Nikolova, 2007; Woodford et al., 2009].

First, we obtain state-of-the-art image restoration results in a purely generative setting without ad-hoc modifications (cf. Roth and Black [2009]), that can compete with recent discriminative methods [Samuel and Tappen, 2009]. Additionally, MAP estimates in image restoration have been shown to exhibit δ-like

marginals [Woodford et al., 2009]. The MMSE estimate gets rid of this problem "for free", without the need to abandon the well-proven MRF framework [Woodford et al., 2009].

The remainder of this work is structured as follows. Chapter 2 introduces the necessary background material and may be skipped by an experienced reader. We formally define our MRF model in Chapter 3 and derive the efficient Gibbs sampler. In Chapter 4, we describe experiments that suggest SM might be rather unsuitable for learning heavy-tailed distributions. Chapter 5 gives a detailed comparison of learning with CD and SM in MRFs, with evaluation in terms of generative properties. In Chapter 6, we point to problems of MAP estimation for posterior inference and show that using the MMSE estimate for good generative models leads to state-of-the-art application results for image restoration tasks. We conclude with a summary in Chapter 7.

2 Background

2.1 Graphical Models in Low-Level Vision

We will review the basics of probabilistic graphical models to lay the foundation for understanding the Markov Random Field models of natural images used in this work. We refer to Roth [2007] and Bishop [2006] for a more thorough treatment.

Taking advantage of graph theory, *probabilistic graphical models* (GMs) are a useful tool to formalize and visualize probability distributions, especially with regard to the conditional independence properties of random variables. Like all graphs, they are comprised of *nodes* (vertices) V and *edges* (links, arcs) E that connect pairs of nodes. Every random variable is represented by a node[1], and the edges encode the relationships between the variables. "The graph then captures the way in which the joint distribution over all of the random variables can be decomposed into a product of factors each depending only on a subset of variables" [Bishop, 2006, p. 360]. Graphical models are usually separated into two major classes:

Directed graphical models. Directed graphical models are represented by directed acyclic graphs, i.e. all edges in the graph are directed and no directed cycles are allowed. These GMs are also known as *Bayesian networks*. The graph structure tells us directly how the model factors into a product of conditional distributions. The example graph from Figure 2.1(a) unambiguously factors in the probability distribution

$$p(a,b,c,d,e) = p(e|b,c,d) \cdot p(d|a,b) \cdot p(c|a) \cdot p(b) \cdot p(a). \tag{2.1}$$

This example applies generally, the probability distribution $p(\mathbf{x})$ of a Bayesian network can be written as the product of conditional distributions of each node x_k that only involve the parent nodes $\pi(x_k)$:

$$p(\mathbf{x}) = \prod_k p(x_k|\pi(x_k)). \tag{2.2}$$

Furthermore, this product is also properly normalized if all conditional distributions are normalized, which is in contrast to undirected graphical models where normalization is often intractable.

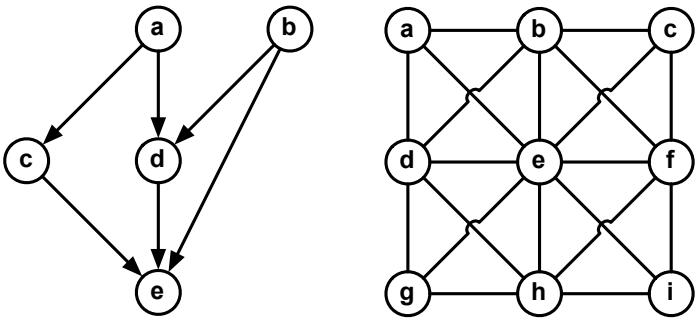
Since no cycles are allowed, the graph structure defines an order of random variables which restricts the types of conditional independence assumptions that can be modeled.

Undirected graphical models. Undirected graphical models are represented by undirected graphs and can, in contrast to directed graphical models, contain arbitrary cycles. The probability distribution factors over the cliques $\mathscr{C}$ of the graph – these are the subsets of fully connected nodes. Each clique $c \in \mathscr{C}$ is associated with a potential function f_c that assigns a positive value to the subset of random variables $\mathbf{x}_{(c)}$ represented by the clique. The potential functions f_c do not necessarily have a probabilistic interpretation, and are not directly related to marginal distributions of subsets of nodes.

The joint distribution can be written as

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in \mathscr{C}} f_c(\mathbf{x}_{(c)}) \tag{2.3}$$

[1] We will not formally distinguish between a node in the graph and the random variable that it represents.



(a) Directed GM (b) Undirected GM

Figure 2.1: Two types of probabilistic graphical models.

where

$$Z = \int \prod_{c \in \mathcal{C}} f_c(\mathbf{x}_{(c)}) d\mathbf{x} \tag{2.4}$$

is a normalization constant (or partition function) that guarantees $p(\mathbf{x})$ to integrate to 1. A big problem in undirected graphical models is that the normalization constant usually cannot be computed in closed form, which complicates learning and inference significantly.

Undirected graphical models are also called Markov Random Fields (MRFs), because they can equivalently be defined in terms of the conditional independence properties of each random variable. Each node v is conditionally independent of all other nodes, given its direct neighbors. In the example graph from Figure 2.1(b), node a is conditionally independent of all other nodes, given its neighbors b, d, and e. This type of conditional independence assumption is often made for low-level vision applications, where a pixel, given a small neighborhood of surrounding pixels, is assumed to be independent of all other pixels of an image.

Since the formulation in terms of cliques is ambiguous (the graph in Figure 2.1(b) contains cliques of size 2, 3, and 4), we will assume to use the maximal cliques in the graph, i.e. no node can be added to the clique such that it ceases to be clique. If the maximal cliques only connect pairs of nodes, we talk about *pairwise Markov Random Fields*; if the cliques contain more than 2 nodes, we call those models *high-order Markov Random Fields*.

Bayesian Networks and Markov Random Fields can express different kinds of conditional independence assumptions; there are also probability distributions whose conditional independence properties cannot be preserved by either of those two types of graphical models.

The order of variables in Bayesian Networks is often interpreted as a causal structure, which may be the reason why most consider directed graphical models to be unsuitable for low-level vision applications [Domke et al., 2008]. A notable recent exception is the work of Domke et al. [2008], where a directed graphical model was trained and applied to common low-level vision tasks. The authors admit that undirected models may theoretically be more suitable for low-level vision problems, but the computational advantages of directed models justify closer investigation.

2.1.1 Inference

The values of some variables (nodes) in the graphical model are usually observed in a concrete application; inference means computing information about the unobserved (hidden) variables $\mathbf{x}$, given the observed variables $\mathbf{y}$. Quantities of interest can be marginal distributions of one or some of the hidden variables; for our purposes it will be an "optimal" configuration of all hidden variables $\mathbf{x}$, given observed

$\mathbf{y}$, which is governed by the posterior distribution $p(\mathbf{x}|\mathbf{y})$. Maximum a-posteriori (MAP) estimation is prevalent in low-level vision and seeks $\mathbf{x}^\star$ that maximizes $p(\mathbf{x}|\mathbf{y})$. We will also be interested in the *posterior mean*, that is the expected value $E[\mathbf{x}|\mathbf{y}]$.

Exact inference in graphical models is generally very hard, which is the reason why approximative inference is usually employed in practice. There are many different classes of approximative inference algorithms (variational, sampling-based, (local) optimization, graph-cuts, etc.; see Roth [2007] for an overview). We will only briefly describe the two approaches required to understand this work.

First, gradient-based techniques for MAP estimation find a (local) optimum of the posterior distribution. The problem with this approach is that for some applications, the result is highly dependent on the initialization of the algorithm. Hence, the technique may only be applicable where a good initial value can be provided. Since this local optimization technique does not make use of the probabilistic properties of the graphical models, it cannot be used to compute the posterior mean.

The other inference approach of interest here is sampling-based. A simple way to approximate a MAP estimate is to draw samples from $p(\mathbf{x}|\mathbf{y})$ and keep the sample with the highest posterior probability. Furthermore, we can approximate the posterior mean by averaging samples drawn from the posterior. Direct sampling is, however, rarely possible so that computationally demanding Markov chain Monte Carlo (MCMC) methods have to be employed (see Section 2.3.2).

2.1.2 Learning

We focus here on learning in undirected GMs, which are relevant for this work. To make an MRF a concrete probability distribution, we have to specify the clique potentials – which are often defined as a parametric family of functions. Such an approach is called parametric, as opposed to non-parametric approaches, where the number of "parameters" of the model is usually dependent on the set of training data. Adopting a Bayesian approach, the parameters $\boldsymbol{\Theta}$ of the potential functions could be treated as additional random variables, which are marginalized out during inference. Unfortunately, such a "fully Bayesian" treatment is computationally infeasible for many problems in practice, including the problems considered in this work. Hence, we can learn the parameters $\boldsymbol{\Theta}^\star$ ahead of time and use them during inference: $p(\mathbf{x}|\mathbf{y}) = \int p(\mathbf{x}, \boldsymbol{\Theta}|\mathbf{y})\, d\boldsymbol{\Theta} \approx p(\mathbf{x}|\mathbf{y}; \boldsymbol{\Theta}^\star)$.

During learning, we assume to have knowledge of all variables of the MRF, including unobserved variables $\mathbf{x}$. When applying models of natural images, $\mathbf{x}$ is usually assumed to be the uncorrupted image behind a corrupted observation $\mathbf{y}$ (cf. Section 2.2.4). Hence, "fully observed" training is possible since sets of (almost) uncorrupted natural images are readily available (e.g., Martin et al. [2001]).

Learning the parameters of MRFs and other unnormalized statistics models is a difficult problem for which many different techniques have been proposed. The most popular learning criteria are arguably maximum likelihood (ML) and maximum a-posteriori (MAP). In ML, the models parameters $\boldsymbol{\Theta}^\star$ are determined by maximizing the likelihood $p(\mathbf{X}; \boldsymbol{\Theta})$ of a training set $\mathbf{X}$; in MAP, an additional prior $p(\boldsymbol{\Theta})$ is imposed on the model parameters. Both approaches unfortunately rely on evaluation of the usually intractable partition function $Z(\boldsymbol{\Theta})$ (cf. Section 2.3.1). Hence, alternative learning methods haven been proposed to alleviate this problem (see Li [2009] for an overview), including score matching [Hyvärinen, 2005] and contrastive divergence [Hinton, 2002], which we will both review in Section 2.3.

2.2 Modeling Natural Images

In the introduction, we motivated the need for prior knowledge to regularize the solution space to ill-posed low-level vision problems. While there are many different approaches to regularization, we will focus on pairwise and high-order MRFs, which allow for generic image priors. First, we will briefly review some key statistical properties of natural images and see how they motivate the MRF models of natural images used in this work.

2.2.1 Natural image statistics

Here we define natural images as photographs that people would typically take with their cameras in everyday life and on special occasions, which includes images from cities and nature, humans and animals, as well as objects encountered in real life. A suitable database for this definition of natural images is the Berkeley segmentation dataset [Martin et al., 2001], which contains 200 training images from a wide variety of scenes that fit the above description. The properties reported here are for "normal" (gamma-compressed) intensity images (e.g. grayscale versions of photographs taken by a digital camera), as opposed to logarithmic or linear intensities which are often used in the literature (e.g. Huang [2000]). The following aspects of natural images are relevant for understanding the motivation behind modeling natural images with MRFs, and MRF evaluation in terms of generative properties. Please note that only a somewhat large collection of natural images will exhibit the properties presented here; they do not necessarily apply to individual images.

Marginal statistics. Marginal distributions of image derivatives are strongly non-Gaussian (Figure 5.19(d)); they exhibit a very strong peak and the tails are very heavy, i.e. they decline very slowly. This phenomenon can be attributed to overlapping and occluding objects in images, which cause large differences in intensity values at object boundaries; a "dead leaves" model of images [Matheron, 1968], that mimics this attribute, has indeed shown similar statistical properties [Lee et al., 2001]. Even marginals of random zero-mean linear filters (Figure 5.19(a)) show characteristic heavy-tailed properties [Huang, 2000].

Scale invariance. Objects in natural images usually occur throughout a large range of sizes, an observation which has been used to explain the approximate scale invariance of natural image statistics (e.g. [Ruderman, 1997]). An example of this property can be seen in Figure 5.19(d), which shows the similarity of derivative statistics at three spatial scales.

Joint statistics. The joint statistics of two neighboring pixels x_1 and x_2 in natural images reveal very strong statistical dependence [Huang, 2000]. The product of the marginal distributions of $x_1 + x_2$ and $x_2 - x_1$ is able to approximate the joint statistics rather well [Huang, 2000], which suggest that the sum and the difference of neighboring pixel values are largely independent. "Dependent random variables in an image can be transformed using difference observations, which makes them more independent." [Roth, 2007]

Most pairwise MRF models of images actually model an image in terms of differences between two neighboring pixels. It is important to note that natural images also exhibit long-range correlations for pixels that are further apart. While pairwise MRFs can only consider differences between pairs of pixels, high-order MRFs usually consider a weighted sum of more than two pixels. Hence, they generalize pairwise MRFs and can potentially capture more of the statistical dependencies in natural images.

Potential functions are often defined in terms of "difference observations" by choosing univariate functions[2] that model the dot product of a linear (zero-mean) filter with the pixels of an MRF clique, especially in high-order MRFs. Zero-mean filters make the MRF invariant to the global gray level of the pixels associated with each clique.

2.2.2 Pairwise Markov Random Fields

In a pairwise MRF model of natural images, each pixel of an image corresponds to a node in the undirected graph. The simplest way to construct a sound pairwise MRF is to connect each pixel with its

[2] Which we will sometimes also call potential functions in a slight abuse of terminology.

horizontal and vertical neighbors (Figure 2.2(a)), hence assuming its conditional independence of all other pixels given the direct neighbors. The potential functions associated with each clique (of two pixels) are usually assumed to be the same for all cliques; such an MRF model is called homogenous. Although this neighborhood structure is very simple, it indirectly connects all pixels in the image by transitivity.

A pairwise MRF, considering the difference of (horizontal and vertical) neighboring pixels, results in the model

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in \mathscr{C}} f\left([1, -1] \cdot \mathbf{x}_{(c)}\right), \tag{2.5}$$

where $\mathbf{x}_{(c)}$ denotes the vector of the neighboring pixel pair associated with clique c, and $[1, -1]^{\mathrm{T}}$ is a linear filter that corresponds to an image derivative. Choosing an appropriate potential function f is crucial to accurately model the statistical properties of natural images. Despite this fact, potentials are often hand-defined. They have early on been modeled as Gaussian (e.g. Woods [1972]), but this does not allow for image discontinuities of large intensity differences (e.g. edges) due to the very low probability at the tails of the potential. To improve this, a number of so-called robust potentials with heavier tails have been proposed over the years.

The potentials are often defined as a parametric family of functions whose parameters can be learned (cf. Section 2.1.2). Although learning and inference is difficult as in most MRFs, the small clique size enables to use some specialized techniques (e.g. graph cuts [Boykov et al., 2001], belief propagation [Yedidia et al., 2003]), which are not (yet) generally applicable to larger clique sizes.

Pairwise MRFs have also been used with more complicated neighborhood structures that connect more distant pixels (e.g. Gimel'farb [1996]). While this can improve performance, pairwise MRFs are conceptually limited because they only consider pairs of pixels. Although they are very general and widely applicable to many problems, they have often performed worse compared to specialized techniques for applications like image denoising.

2.2.3 High-order Markov Random Fields

High-order MRFs are a generalization of pairwise MRFs, because the maximal cliques can generally be defined as all overlapping $m \times m$ pixel neighborhoods in the MRF. Each pixel is connected to its closest $4m^2 - 4m$ neighbors, therefore making a weaker conditional independence assumption as in the pairwise MRF. See Figure 2.2(b) for an example of a high-order MRF with 2×2 cliques. Other, less connected neighborhood structures, can easily be achieved by having the potential functions ignore some of the clique's pixels.

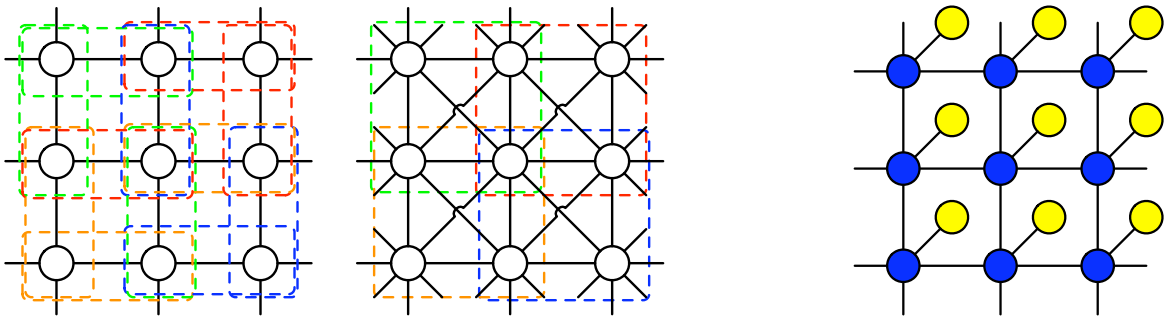
This results in the abstract model definition

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in \mathscr{C}} f(\mathbf{x}_{(c)}), \tag{2.6}$$

where $\mathbf{x}_{(c)}$ is a vector of the pixels that make up the $m \times m$ patch associated with maximal clique c.

High-order MRFs are also typically homogenous, i.e. the potential f is the same for all cliques in the MRF. Note that the maximal cliques overlap, which is also true in the pairwise MRF, and that boundary pixels are overlapped by fewer cliques than interior pixels, which can lead to problems during learning and inference (Chapter 5 and 6).

Although high-order MRFs possess increased modeling power compared to pairwise models, they are computationally more demanding, and choosing suitable potential functions is generally more difficult because they are defined on larger cliques (high-dimensional space).



(a) Pairwise MRF (b) High-order MRF (2×2 cl.) (c) Pairw. MRF for denoising

Figure 2.2: Graphical model representation of MRFs. *(a, b)* MRF neighborhood structure; the dashed colored lines denote overlapping cliques. *(c)* Pairwise MRF for a denoising application; the blue nodes represent the hidden variables of the denoised image, which are constrained by the pairwise MRF. The pixels of the observed noisy image are shown with yellow nodes.

Fields of Experts

The Fields of Experts (FoE) framework was introduced by Roth and Black [2005], which is a high-order MRF with potential functions modeled by Products of Experts (PoE) [Hinton, 1999]. It has been widely adopted by others (e.g. Samuel and Tappen [2009]; Weiss and Freeman [2007]) and will also serve as the foundation for our MRF image prior (Chapter 3).

We mentioned above that potentials in high-order MRFs need to model a high-dimensional space; this is accomplished by using Products of Experts in the FoE, which take the product of several low-dimensional distributions, so-called expert functions, to model a high-dimensional probability distribution. In the FoE, the potential functions

$$f(\mathbf{x}_{(c)}) = \prod_{i=1}^{N} \phi(\mathbf{w}_i^{\mathrm{T}} \mathbf{x}_{(c)}; \boldsymbol{\alpha}_i) \tag{2.7}$$

are PoEs with a family of univariate expert functions ϕ specified by parameters $\boldsymbol{\alpha}_i$ and associated linear zero-mean filters $\mathbf{w}_i$. Roth and Black [2005] chose Student-t experts

$$\phi(\mathbf{w}_i^{\mathrm{T}} \mathbf{x}_{(c)}; \alpha_i) = \left(1 + \frac{1}{2} \left(\mathbf{w}_i^{\mathrm{T}} \mathbf{x}_{(c)}\right)^2\right)^{-\alpha_i} \tag{2.8}$$

in the original FoE model, and learned all model parameters $\boldsymbol{\Theta} = \{\mathbf{w}_i, \alpha_i | i = 1, \ldots, N\}$ ($N = 24$, 5×5 cliques/filters) from training data using the method of contrastive divergence (see Section 2.3.3).

Being a general purpose image prior, the FoE has shown remarkable performance for various applications including image denoising, which however required a regularization weight to emphasize the likelihood over the FoE prior during inference; the denoised images were otherwise too smooth.

2.2.4 Applications

MRF models of natural images can in principal be used to regularize all ill-defined problems that can be expressed in a probabilistic way, although it highly depends on the problem and inference method whether a good solution can be achieved. Applications of interest in general, and for this work, contain problems of image restoration, where parts of an image are missing or corrupted.

Performance is mostly measured in *peak signal-to-noise ratio* (PSNR)

$$\mathrm{PSNR} = 20 \log_{10} \frac{255}{\sigma_e}, \tag{2.9}$$

which is based on the pixel-wise mean squared error (MSE) σ_e^2 of the restored image. PSNR is expressed in decibels (dB) on a logarithmic scale to cover a wide range of error values.

Human perception of restoration quality is usually the goal in image restoration, and although a human observer often agrees with the PSNR, this is not the case on all accounts. A more realistic error measure, based on human perception, is offered by the *structural similarity index* (SSIM) [Wang et al., 2004]. SSIM is expressed between 0 and 1, where 1 is a perfect restoration.

Image denoising. Image denoising in the context of i.i.d. Gaussian noise with known standard deviation σ has become a benchmark for MRF priors of natural images. We assume that the observed noisy image $\mathbf{y}$ was generated by adding noise to the uncorrupted (and unobserved) image $\mathbf{x}$ – which we want to recover. Assuming that $\mathbf{y} = \mathbf{x} + \mathbf{n}$ with $\mathbf{n} \sim \mathcal{N}(0, \sigma^2)$, we formalize the relation between $\mathbf{x}$ and $\mathbf{y}$ by specifying the likelihood

$$p(\mathbf{y}|\mathbf{x}) \propto \mathcal{N}(\mathbf{y}; \mathbf{x}, \sigma^2 \mathbf{I}). \tag{2.10}$$

Using Bayes rule, we obtain the posterior $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{x})$ where we use our MRF model of natural images as the prior $p(\mathbf{x})$. The posterior is depicted as graphical model in Figure 2.2(c), where a pairwise MRF is used as a prior.

Image inpainting. The goal in image inpainting [Bertalmío et al., 2000] is to fill in missing, corrupted, or unwanted pixels of an observed image $\mathbf{y} \in \mathbb{R}^D$. We assume that a mask M of defective pixels is provided to us, but make otherwise no further assumptions. The masked pixels are entirely dependent on the image prior, the other pixels must not be changed; this is formally defined by using the likelihood

$$p(\mathbf{y}|\mathbf{x}) \propto \prod_{d=1}^{D} \left\{ \begin{array}{ll} 1, & d \in M \\ \delta(y_d - x_d), & d \notin M \end{array} \right\}, \tag{2.11}$$

which assigns a uniform probability to all masked pixels of the image. The Dirac delta with $\delta(a) = 0$ for $a \neq 0$ guarantees probability 0 for all image restorations that change the value of unmasked pixels. Since all unmasked pixels need to stay fix, we can alternatively set $\mathbf{x}_{\setminus M} = \mathbf{y}_{\setminus M}$, and express the problem as the conditional distribution $p(\mathbf{x}_M|\mathbf{x}_{\setminus M})$ using the MRF prior alone ($\setminus M = \{1, \ldots, D\} \setminus M$).

There are a variety of other applications for image priors, such as super-resolution [Tappen et al., 2003], where the goal is to produce a natural looking image of increased spatial resolution.

2.3 Learning Unnormalized Statistical Models

As introduced in Section 2.1.2, learning in undirected graphical models is a hard problem. In this section we want to review the learning methods of relevance to this work, since they are crucial to understand the experiments that we carried out. These methods are however not specific to graphical models, they apply generally to unnormalized statistical models.

2.3.1 Maximum likelihood

Maximum likelihood (ML) is probably the most popular learning method in general. It unfortunately requires evaluation of the often intractable partition function in statistical models, which depends on the model parameters.

Let

$$p(\mathbf{x}; \boldsymbol{\Theta}) = \frac{1}{Z(\boldsymbol{\Theta})} q(\mathbf{x}; \boldsymbol{\Theta}) \tag{2.12}$$

be the normalized probability density function (pdf), whose defining parameters $\boldsymbol{\Theta}$ we want to estimate from given i.i.d. training data $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}\}$. The partition function

$$Z(\boldsymbol{\Theta}) = \int q(\mathbf{x}; \boldsymbol{\Theta}) \, d\mathbf{x}. \tag{2.13}$$

is mostly intractable in practice, so that we have to work with the unnormalized pdf $q(\mathbf{x}; \boldsymbol{\Theta})$. Instead of maximizing the likelihood directly, the log-likelihood function

$$\ell(\boldsymbol{\Theta}) = \log \prod_{t=1}^{T} p(\mathbf{x}^{(t)}; \boldsymbol{\Theta}) = \sum_{t=1}^{T} \log p(\mathbf{x}^{(t)}; \boldsymbol{\Theta}) = \sum_{t=1}^{T} -\log Z(\boldsymbol{\Theta}) + \log q(\mathbf{x}^{(t)}; \boldsymbol{\Theta}) \tag{2.14}$$

is usually considered for computational reasons – but still depends on the partition function $Z(\boldsymbol{\Theta})$. This integral is usually impossible to compute if no closed-form expression exists, which is the case for most MRF models that do not use Gaussian potentials. It is sometimes possible to approximate $Z(\boldsymbol{\Theta})$, but there is no generic way of doing it. Hence, it is generally hard to come up with a good approximation, which could otherwise be quite poor.

Although evaluation of $\ell(\boldsymbol{\Theta})$ is intractable, its derivatives w.r.t. the model parameters $\boldsymbol{\Theta}$ can be approximated if it is possible to sample from $p(\mathbf{x}; \boldsymbol{\Theta})$:

$$\begin{aligned}
\frac{\partial \ell(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} &= \frac{\partial}{\partial \boldsymbol{\Theta}} \left(\sum_{t=1}^{T} -\log Z(\boldsymbol{\Theta}) + \log q(\mathbf{x}^{(t)}; \boldsymbol{\Theta}) \right) \\
&= -T \frac{\partial \log Z(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} + \sum_{t=1}^{T} \frac{\partial \log q(\mathbf{x}^{(t)}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \\
&= T \left(-\frac{\frac{\partial}{\partial \boldsymbol{\Theta}} Z(\boldsymbol{\Theta})}{Z(\boldsymbol{\Theta})} + \frac{1}{T} \sum_{t=1}^{T} \frac{\partial \log q(\mathbf{x}^{(t)}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right) \\
&\overset{(2.13)}{=} T \left(-\frac{\frac{\partial}{\partial \boldsymbol{\Theta}} \int q(\mathbf{x}; \boldsymbol{\Theta}) \, d\mathbf{x}}{Z(\boldsymbol{\Theta})} + \left\langle \frac{\partial \log q(\mathbf{x}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{\mathbf{x}} \right) \\
&= T \left(-\int \frac{1}{Z(\boldsymbol{\Theta})} \frac{q(\mathbf{x}; \boldsymbol{\Theta})}{q(\mathbf{x}; \boldsymbol{\Theta})} \frac{\partial q(\mathbf{x}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \, d\mathbf{x} + \left\langle \frac{\partial \log q(\mathbf{x}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{\mathbf{x}} \right) \\
&\overset{(2.12)}{=} T \left(-\int p(\mathbf{x}; \boldsymbol{\Theta}) \frac{\frac{\partial}{\partial \boldsymbol{\Theta}} q(\mathbf{x}; \boldsymbol{\Theta})}{q(\mathbf{x}; \boldsymbol{\Theta})} \, d\mathbf{x} + \left\langle \frac{\partial \log q(\mathbf{x}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{\mathbf{x}} \right) \\
&= T \left(-\int p(\mathbf{x}; \boldsymbol{\Theta}) \frac{\partial \log q(\mathbf{x}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \, d\mathbf{x} + \left\langle \frac{\partial \log q(\mathbf{x}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{\mathbf{x}} \right) \\
&= T \left(-\left\langle \frac{\partial \log q(\mathbf{x}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{p} + \left\langle \frac{\partial \log q(\mathbf{x}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{\mathbf{x}} \right) \\
&\approx T \left(-\frac{1}{S} \sum_{\substack{s=1 \\ \mathbf{y}^{(s)} \sim p(.; \boldsymbol{\Theta})}}^{S} \frac{\partial \log q(\mathbf{y}^{(s)}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} + \frac{1}{T} \sum_{t=1}^{T} \frac{\partial \log q(\mathbf{x}^{(t)}; \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right)
\end{aligned} \tag{2.15}$$

In the above equation, $\langle . \rangle_p$ denotes taking the expected value w.r.t. the model pdf $p(.; \boldsymbol{\Theta})$ and $\langle . \rangle_{\mathbf{X}}$ takes the expected value w.r.t. the empirical data distribution $\mathbf{X}$. The above derivation assumes that $q(\mathbf{x}; \boldsymbol{\Theta})$ is continuous, differentiable and greater than zero for all $\mathbf{x}$; alternative derivations are possible if $q(\mathbf{x}; \boldsymbol{\Theta}) = e^{-E(\mathbf{x}; \boldsymbol{\Theta})}$ is assumed.

2.3.2 Gibbs sampling

As shown above, ML estimation is possible if we can draw samples from the model pdf. For this purpose, a Gibbs sampler [Geman and Geman, 1984] is often used when applicable, which is an algorithm of the general class of Markov chain Monte Carlo (MCMC) methods. MCMC methods allow to draw samples from (unnormalized) probability distributions of high dimensionality; they work by iteratively drawing samples that form a Markov chain (a sequence of random variables with the Markov property). The Markov chain is set up to have the desired probability distribution at its equilibrium, i.e. the samples are distributed according the the target distribution when the Markov chain is run for long enough.

Consider the distribution $p(\mathbf{x}) = p(x_1, \ldots, x_N)$ from which we want to sample but are unable to do so directly. Assume that the conditional distributions $p(x_i|\mathbf{x}_{\setminus i})$ can be obtained where $\mathbf{x}_{\setminus i} = \left[x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_D\right]^{\mathrm{T}}$ denotes the random vector $\mathbf{x}$ without the i^{th} component. If we further suppose that direct sampling is relatively easy for $p(x_i|\mathbf{x}_{\setminus i})$, then Gibbs sampling is often an efficient way to obtain samples from $p(\mathbf{x})$.

The algorithm works by replacing the value of one of the variables x_i with a sample drawn from conditional distribution $p(x_i|\mathbf{x}_{\setminus i})$ at each iteration, thereby advancing the Markov Chain. This is repeated for all variables in some particular order or even randomly, especially when D is large. Although it is not obvious from the above explanation, the Gibbs sampler can indeed be interpreted as a special of a more general MCMC algorithm. This can be used to show that the procedure converges to samples from the target distribution $p(\mathbf{x})$, regardless of initialization, given that no conditional distribution is zero anywhere.

There are strong dependencies between successive samples because the Gibbs sampler, as described above, only considers one variable at a time. Iterating over all variables may also take a long time if D is large. Both of these issues can be improved by sampling groups of variables at each iteration, a strategy sometimes called *blocking Gibbs* sampling.

Consider the joint distribution $p(\mathbf{x}, \mathbf{z})$ with random vectors $\mathbf{x}$ and $\mathbf{z}$. If the conditional distributions $p(\mathbf{x}|\mathbf{z})$ and $p(\mathbf{z}|\mathbf{x})$ are easy to sample from directly, they can be used to sample from $p(\mathbf{x}, \mathbf{z})$ as follows:

1: Initialize $\mathbf{x}^{(1)}$
2: **for** $j = 1$ to J **do**
3: Sample $\mathbf{z}^{(j+1)} \sim p\left(\mathbf{z}|\mathbf{x}^{(j)}\right)$
4: Sample $\mathbf{x}^{(j+1)} \sim p\left(\mathbf{x}|\mathbf{z}^{(j+1)}\right)$
5: **end for**

Note that $\mathbf{x}^{(j)}$ and $\mathbf{z}^{(j)}$ denote the values of the random vectors after the j^{th} iteration, hence $\left[\mathbf{x}^{(J)}, \mathbf{z}^{(J)}\right]$ denotes a single sample from the target distribution $p(\mathbf{x}, \mathbf{z})$ if J has been chosen large enough for the Markov chain to converge; all iterations prior convergence are usually called "burn-in phase". Assessing convergence of the Gibbs sampler, or in other words choosing the number of iterations J, is a difficult and long-standing problem – we explain the approach that we adopted in Section 3.1.2.

2.3.3 Contrastive divergence

As outlined above, (blocking) Gibbs sampling transforms (randomly initialized) random vectors to samples from the model distribution by iteratively sampling from conditional distributions. It can however take many iterations for the Markov chain to converge, which causes ML estimation by Eq. (2.15) to be slow or even intractable in practice.

To alleviate this problem, the idea of contrastive divergence [Hinton, 2002] is to initialize Gibbs samplers with given training examples, which are then only run for one or a few number of iterations.

The derivative of the log-likelihood function (Eq. (2.15))

$$\begin{aligned}\frac{\partial \ell(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} &\propto \left\langle \frac{\partial \log q(\mathbf{x};\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{\mathbf{X}} - \left\langle \frac{\partial \log q(\mathbf{x};\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{p} \\ &\propto \left\langle \frac{\partial \log q(\mathbf{x};\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{p^0} - \left\langle \frac{\partial \log q(\mathbf{x};\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{p^\infty}\end{aligned} \tag{2.16}$$

is proportional to the difference between the expected log-derivatives of the data and model distributions. The training set $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}\}$ can equally be denoted as p^0 if we imagine T Gibbs samplers, each initialized with one of training examples $\mathbf{x}^{(t)}$ but run for zero iterations. Using the same analogy, p^∞ denotes the same set of Gibbs samplers run until convergence to obtain samples from the model distribution.

Each iteration of the Gibbs sampler, the random variables move further away from the training examples and become more similar to samples from the model distribution. Intuitively, if a few iterations of the Gibbs sampler hardly change the value of the random variables, then the initial values were already appropriate samples from the model.

Hence, a suitable surrogate for the ML learning rule from Eq. (2.15) is the contrastive divergence

$$T\left[\left\langle \frac{\partial \log q(\mathbf{x};\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{p^0} - \left\langle \frac{\partial \log q(\mathbf{x};\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \right\rangle_{p^k}\right], \tag{2.17}$$

where k denotes the number of iterations of the Gibbs sampler, which can be chosen much smaller than necessary for convergence, often even $k = 1$.

This is obviously only an intuitive explanation of contrastive divergence to better understand the remainder of this work. We refer the interested reader to the literature.

2.3.4 Score matching

Score matching (SM) was originally proposed by Hyvärinen [2005] as a computationally inexpensive way to estimate the parameters of non-normalized statistical models from training examples, in the case of continuous-valued variables defined over $\mathbb{R}^n$. The motivation for this new procedure, as well as many related methods, is to avoid evaluation of the mostly intractable partition function of the model pdf. Hyvärinen suggests to minimize the expected squared distance between the gradient of the log-density of the data and the gradient of the log-density of the model. The gradient of log-density is loosely called the *score function* – hence the term *score matching*. Estimation of the data score function would be a very challenging problem itself, mostly infeasible when dealing with high-dimensional data in practice. However, Hyvärinen proves the surprising result that this is not required in order to evaluate the SM objective function in a reformulated form, only involving computations of the score function and its derivative. For this to hold, some regularity conditions are assumed to hold for the model and data densities.

Score matching has also been shown to be (locally) consistent [Hyvärinen, 2005], i.e. convergence is guaranteed if the model density follows the data density. In subsequent work [Hyvärinen, 2008], it is suggested that SM is actually the optimal estimator in an "empirical Bayes" setting under various assumptions, which we will briefly summarize below. Another desirable property of SM is that the objective function can be obtained in closed form for certain exponential families [Hyvärinen, 2007b].

Lyu [2009] has shown a formal relation between score matching and maximum likelihood by demonstrating that the SM objective function is the derivative of the ML objective function – the Kullback-Leibler (KL) divergence [Kullback and Leibler, 1951] – in the scale space of probability density functions w.r.t. the scale factor. He suggests the interpretation that in the presence of small amounts of noise in the training

data, the SM objective seeks optimal parameters that lead to the least changes (i.e., stability) in the KL-divergence, whereas ML tries to maximize it. This however only indicates that score matching is seeking a solution that is less affected by small amounts of noise, not that minimizing the reformulated objective function by some algorithm like gradient descent is less sensitive to noise in the training data. We will suggest that rather the opposite is true.

Hyvärinen [2007a] has shown a relation between contrastive divergence and score matching, and even equality in a special case of a specific Monte Carlo method. Sohl-Dickstein et al. [2009] even introduced a new estimation framework called "Minimum Probability Flow Learning", of which score matching and certain forms of contrastive divergence are shown to be special cases of. Their technique works by first establishing (random walk) system dynamics that would transform the observed training data into the model distribution. The initial flow of probability away from the data distribution is then minimized as the objective function.

Below, we will first introduce score matching formally and then review a theoretical property of interest. In Chapter 4, we will demonstrate some practical properties of score matching in comparison to maximum likelihood in simple univariate experiments. Our findings suggest that SM has some problems with heavy-tailed densities, a form of "non-smooth" model. Hyvärinen already stated in the concluding remarks of his 2005 paper that one main assumption of score matching is that "the model pdf is smooth enough".

Basic method

We consider, as in Section 2.3.1, the case of a continuous model density

$$p(\mathbf{x};\boldsymbol{\Theta}) = \frac{1}{Z(\boldsymbol{\Theta})} q(\mathbf{x};\boldsymbol{\Theta}) \tag{2.18}$$

with $\mathbf{x} \in \mathbb{R}^D$ whose parameters $\boldsymbol{\Theta}$ we want to estimate from observed i.i.d. data $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots\}$ which is distributed according to some unknown distribution $p_{\mathbf{X}}(\mathbf{x})$. We further assume that the partition function

$$Z(\boldsymbol{\Theta}) = \int q(\mathbf{x};\boldsymbol{\Theta})\, d\mathbf{x}, \tag{2.19}$$

required to normalize the model density, is not known in closed-form. Hence, the integral $Z(\boldsymbol{\Theta})$ has to be approximated or evaluated numerically in order to do maximum likelihood estimation of $\boldsymbol{\Theta}$. Since direct numerical evaluation is almost always impossible in practice (mostly even for $D \geq 3$), one usually needs to resort to MCMC methods, as described above.

Hyvärinen focuses on the gradient of log-density, called score function, since it does not depend on $Z(\boldsymbol{\Theta})$. He defines it as

$$\boldsymbol{\psi}(\mathbf{x};\boldsymbol{\Theta}) = \begin{pmatrix} \dfrac{\partial \log p(\mathbf{x};\boldsymbol{\Theta})}{\partial x_1} \\ \vdots \\ \dfrac{\partial \log p(\mathbf{x};\boldsymbol{\Theta})}{\partial x_D} \end{pmatrix} = \begin{pmatrix} \psi_1(\mathbf{x};\boldsymbol{\Theta}) \\ \vdots \\ \psi_D(\mathbf{x};\boldsymbol{\Theta}) \end{pmatrix} = \nabla_{\mathbf{x}} \log p(\mathbf{x};\boldsymbol{\Theta}) = \nabla_{\mathbf{x}} \log q(\mathbf{x};\boldsymbol{\Theta}) \tag{2.20}$$

Likewise, the score function of the observed data $\mathbf{X}$ is denoted as

$$\boldsymbol{\psi}_{\mathbf{X}}(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_{\mathbf{X}}(\mathbf{x}). \tag{2.21}$$

He proposes to estimate the model parameters by minimizing the expected squared distance between the model score function and the data score function

$$J(\boldsymbol{\Theta}) = \frac{1}{2}\int \|\boldsymbol{\psi}(\mathbf{x};\boldsymbol{\Theta}) - \boldsymbol{\psi}_{\mathbf{X}}(\mathbf{x})\|^2 \, d\mathbf{x}. \tag{2.22}$$

In order to compute $J(\boldsymbol{\Theta})$, the unknown data density $p_{\mathbf{X}}(\mathbf{x})$ could be estimated by non-parametric density estimation, but this is a challenging and computationally demanding task itself, especially when dealing with many dimensions. Hyvärinen however proves the surprising result that $p_{\mathbf{X}}(\mathbf{x})$ is not required to minimize $J(\boldsymbol{\Theta})$; he shows that Eq. (2.22) can be rewritten as

$$J(\boldsymbol{\Theta}) = \frac{1}{2}\int \sum_{d=1}^{D} \left[\psi_d'(\mathbf{x};\boldsymbol{\Theta}) + \frac{1}{2}\psi_d(\mathbf{x};\boldsymbol{\Theta})^2 \right] d\mathbf{x} + \text{const.}, \tag{2.23}$$

where the constant term does not depend on $\boldsymbol{\Theta}$, assuming that:

1. $\boldsymbol{\psi}(\mathbf{x};\boldsymbol{\Theta})$ and $\boldsymbol{\psi}_{\mathbf{X}}(\mathbf{x})$ are differentiable,
2. the expectations $\left\langle \|\boldsymbol{\psi}(\mathbf{x};\boldsymbol{\Theta})\|^2 \right\rangle_{\mathbf{X}}$ and $\left\langle \|\boldsymbol{\psi}_{\mathbf{X}}(\mathbf{x})\|^2 \right\rangle_{\mathbf{X}}$ are finite for any $\boldsymbol{\Theta}$,
3. $p_{\mathbf{X}}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{x};\boldsymbol{\Theta})$ goes to zero for any $\boldsymbol{\Theta}$ when $\|\mathbf{x}\| \to \infty$.

In practice, where we want to estimate $\boldsymbol{\Theta}$ from given training examples $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}\}$, the sample version of Eq. (2.23) becomes

$$\tilde{J}(\boldsymbol{\Theta}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{d=1}^{D} \left[\psi_d'(\mathbf{x}^{(t)};\boldsymbol{\Theta}) + \frac{1}{2}\psi_d(\mathbf{x}^{(t)};\boldsymbol{\Theta})^2 \right] + \text{const.}, \tag{2.24}$$

where the constant term and scalar $1/T$ can be ignored since they do not change $\arg\min_{\boldsymbol{\Theta}} \tilde{J}(\boldsymbol{\Theta})$.

Optimal denoising

Consider the setting as described in Section 2.1.2, i.e. we learn (a point estimate of the) model parameters $\boldsymbol{\Theta}$ prior to doing inference with the model. Further suppose that we train our model to do image denoising by inferring the uncorrupted image $\mathbf{x}$, given the noisy image $\mathbf{y}$, where we assume the Gaussian likelihood

$$p(\mathbf{y}|\mathbf{x}) \propto \mathcal{N}(\mathbf{y};\mathbf{x},\sigma^2\mathbf{I}). \tag{2.25}$$

Specifically, we are interested in the MAP estimate

$$\hat{\mathbf{x}}_{\text{MAP}} = \arg\max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{x};\boldsymbol{\Theta}) \tag{2.26}$$

of the denoised image.

As previously stated, maximum likelihood is the most popular and widely used criteria to learn the model parameters $\boldsymbol{\Theta}$. ML learning minimizes the error in the estimate $\boldsymbol{\Theta}$ of seeing the data in the training set, which is directly related to minimizing the KL-divergence between the empirical distribution of the training set and the model distribution. In circumstances like this, however, we are usually interested in denoising performance, i.e. we want to minimize the error in the MAP estimate of $\mathbf{x}$. Hence, ML may not be the best way to estimate $\boldsymbol{\Theta}$, since a small error in the estimate of $\boldsymbol{\Theta}$ does not imply a small error in the MAP estimate of $\mathbf{x}$. Consequently, the "optimal estimator" for $\boldsymbol{\Theta}$ should be based on minimizing the error in the MAP estimate of $\mathbf{x}$.

Hyvärinen [2008] demonstrates that score matching, as defined above, is the optimal estimator in terms of minimizing the (squared) error in the MAP estimate of $\mathbf{x}$ under these circumstances. However, this only holds true for a Gaussian likelihood as in Eq. (2.25) where $\sigma \to 0$. In other words, the corrupted image $\mathbf{y}$ is assumed to be generated by adding Gaussian noise with infinitesimal variance to $\mathbf{x}$. Noise of infinitesimal variance allows the author to do first-order approximations which are the core of his analysis. This is a purely theoretical result, however, and it has to be shown how the assumption of infinitesimal variance relates to denoising performance in practice, when score matching is used to learn $p(\mathbf{x}; \boldsymbol{\Theta})$. We could not confirm this theoretical result in our experiments (Chapter 6).

3 Flexible MRF Model and Efficient Sampling

Our MRF prior stays within the Fields of Experts (FoE) framework [Roth and Black, 2009], a high-order MRF whose clique potentials are expressed as Products of Experts [Hinton, 1999] that model the responses to a bank of linear filters $\mathbf{w}_i$. The probability density of an image $\mathbf{x}$ under the FoE is written as

$$p(\mathbf{x};\boldsymbol{\Theta}) = \frac{1}{Z(\boldsymbol{\Theta})} e^{-\epsilon\|\mathbf{x}\|^2/2} \prod_{k=1}^{K} \prod_{i=1}^{N} \phi\left(\mathbf{w}_i^{\mathrm{T}} \mathbf{x}_{(k)}; \boldsymbol{\alpha}_i\right), \tag{3.1}$$

where $\mathbf{x}_{(k)}$ are the pixels of the k^{th} maximal clique, ϕ is an expert function, $\boldsymbol{\alpha}_i$ are the expert parameters for linear filter $\mathbf{w}_i$, and $Z(\boldsymbol{\Theta})$ is the partition function that depends on all model parameters $\boldsymbol{\Theta} = \{\mathbf{w}_i, \boldsymbol{\alpha}_i | i = 1, \dots, N\}$.

The very broad Gaussian factor $e^{-\epsilon\|\mathbf{x}\|^2/2}$ with $\epsilon = 10^{-8}$ guarantees the model to be normalizable because we generally use zero-mean filters (Chapter 5) that do not fully constrain the image space (cf. Weiss and Freeman [2007]); the values $p(\mathbf{x};\boldsymbol{\Theta})$ and $p(\mathbf{x} + \text{const.};\boldsymbol{\Theta})$ would be equal if we were not using such a factor – which would also imply $Z(\boldsymbol{\Theta}) = \infty$ and not being able to sample from our model.

Following Weiss and Freeman [2007], we use flexible Gaussian scale mixtures (GSMs) as experts[1]

$$\phi(\mathbf{w}_i^{\mathrm{T}} \mathbf{x}_{(k)}; \boldsymbol{\alpha}_i) = \sum_{j=1}^{J} \beta_{ij} \cdot \mathcal{N}(\mathbf{w}_i^{\mathrm{T}} \mathbf{x}_{(k)}; 0, \sigma_i^2 / s_j) \tag{3.2}$$

where

$$\beta_{ij} = \frac{\exp(\alpha_{ij})}{\sum_{j'=1}^{J} \exp(\alpha_{ij'})} \tag{3.3}$$

is the weight of the Gaussian component with scale s_j and base variance σ_i^2. This definition ensures positive mixture weights that are normalized, i.e. $\sum_j \beta_{ij} = 1$. In contrast to Weiss and Freeman [2007], however, we use a different GSM for each filter $\mathbf{w}_i$ and learn the respective parameters $\boldsymbol{\alpha}_i$ together with the filter coefficients, instead of fixing them beforehand.

GSMs can represent a wide variety of well-known heavy-tailed distributions, including Student-t [Roth and Black, 2009] and generalized Laplacians [Tappen et al., 2003]. More importantly, they support a much broader range of shapes when using suitable scales – which we do by choosing exponentially-spaced scales (together with a fixed base variance).

Our MRF model from Eq. (3.1) subsumes a variety of FoE-based models [Roth and Black, 2009; Samuel and Tappen, 2009; Weiss and Freeman, 2007] and pairwise MRFs [Lan et al., 2006; Levin et al., 2009; Tappen et al., 2003]. For the pairwise case, we simply define a single fixed filter $\mathbf{w}_1 = [1, -1]^{\mathrm{T}}$ and let the maximal cliques be all pairs of horizontal and vertical neighbors.

3.1 Auxiliary-variable Gibbs Sampler

A fast and rapidly-mixing sampling procedure is crucial for analyzing the generative properties of MRF priors through samples, and also required for efficient training via ML/CD. Direct sampling is not possible

[1] Note that we sometimes use the terms potential and expert interchangeably, depending on the context.

due to the intractable partition function; hence, we resort to Markov chain Monte Carlo methods. Single-site Gibbs samplers [Geman and Geman, 1984; Zhu and Mumford, 1997], which update the value of one pixel at a time, are very slow as they need many iterations for the image vector to reach the equilibrium distribution.

We exploit here that our potentials use Gaussian Scale Mixtures which allows us to rather naturally equip our MRF model with a set of (hidden) auxiliary random variables $\mathbf{z}$, which are similar to the indicator variables of a regular mixture model. The joint distribution $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\Theta})$ of $\mathbf{x}$ and auxiliary mixture coefficients $\mathbf{z}$ can then be defined such that $\sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}|\boldsymbol{\Theta}) = p(\mathbf{x}|\boldsymbol{\Theta})$. This well-known strategy of adding variables to improve Gibbs sampling is sometimes called *data augmentation* [Gelman et al., 2004].

Welling et al. [2003] already showed in the context of Products of Experts [Hinton, 1999] that augmenting the model with hidden random variables $\mathbf{z}$ lends to a rapidly mixing Gibbs sampler that alternates between sampling

$$\mathbf{z}^{(t+1)} \sim p(\mathbf{z}|\mathbf{x}^{(t)}; \boldsymbol{\Theta}) \quad \text{and} \quad \mathbf{x}^{(t+1)} \sim p(\mathbf{x}|\mathbf{z}^{(t+1)}; \boldsymbol{\Theta}), \tag{3.4}$$

where t denotes the current iteration. After convergence, the $\mathbf{z}$s can be discarded since we usually only care about obtaining samples of $\mathbf{x}$. The whole image vector can be sampled at once, which significantly speeds up convergence to the equilibrium distribution as compared to updating one pixel at a time in single-site Gibbs samplers. Levi [2009] applied this technique to MRFs with arbitrary Gaussian mixture potentials where $\mathbf{z} \in \{1, \dots, J\}^{N \times K}$, one indicator variable for each expert and clique.

We can apply this to our case and first rewrite the model density from Eqs. (3.1) and (3.2) as

$$p(\mathbf{x}; \boldsymbol{\Theta}) = \sum_{\mathbf{z}} \frac{1}{Z(\boldsymbol{\Theta})} e^{-\epsilon \|\mathbf{x}\|^2/2} \prod_{k=1}^{K} \prod_{i=1}^{N} p(z_{ik}) \cdot \mathcal{N}(\mathbf{w}_i^{\mathrm{T}} \mathbf{x}_{(k)}; 0, \sigma_i^2 / s_{z_{ik}}), \tag{3.5}$$

where we treat the scales indices $\mathbf{z} \in \{1, \dots, J\}^{N \times K}$ for each expert and clique as random variables with $p(z_{ik}) = \beta_{i z_{ik}}$ (i.e. the normalized GSM mixture weights). Instead of marginalizing out the scale indices, we can also retain them explicitly and define the joint distribution (cf. Welling et al. [2003])

$$p(\mathbf{x}, \mathbf{z}; \boldsymbol{\Theta}) = \frac{1}{Z(\boldsymbol{\Theta})} e^{-\epsilon \|\mathbf{x}\|^2/2} \prod_{k=1}^{K} \prod_{i=1}^{N} p(z_{ik}) \cdot \mathcal{N}(\mathbf{w}_i^{\mathrm{T}} \mathbf{x}_{(k)}; 0, \sigma_i^2 / s_{z_{ik}}). \tag{3.6}$$

Since the scale indices are conditionally independent given the image, the conditional distribution $p(\mathbf{z}|\mathbf{x}; \boldsymbol{\Theta})$ is fully defined by

$$p(z_{ik}|\mathbf{x}; \boldsymbol{\Theta}) \propto p(z_{ik}) \cdot \mathcal{N}(\mathbf{w}_i^{\mathrm{T}} \mathbf{x}_{(k)}; 0, \sigma_i^2 / s_{z_{ik}}). \tag{3.7}$$

Sampling from these discrete distributions is straightforward and efficient.

The conditional distribution $p(\mathbf{x}|\mathbf{z}; \boldsymbol{\Theta})$ can be derived as the multivariate Gaussian

$$\begin{aligned}
p(\mathbf{x}|\mathbf{z}; \boldsymbol{\Theta}) &\propto p(\mathbf{x}, \mathbf{z}; \boldsymbol{\Theta}) \\
&\propto e^{-\epsilon \|\mathbf{x}\|^2/2} \prod_{k=1}^{K} \prod_{i=1}^{N} \exp\left(-\frac{s_{z_{ik}}}{2\sigma_i^2} \left(\mathbf{w}_i^{\mathrm{T}} \mathbf{x}_{(k)} \right)^2 \right) \\
&\propto \exp\left(-\frac{\epsilon}{2} \|\mathbf{x}\|^2 + \sum_{i=1}^{N} \sum_{k=1}^{K} -\frac{s_{z_{ik}}}{2\sigma_i^2} \left(\mathbf{w}_{ik}^{\mathrm{T}} \mathbf{x} \right)^2 \right) \\
&\propto \exp\left(-\frac{1}{2} \mathbf{x}^{\mathrm{T}} \left(\epsilon \mathbf{I} + \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{s_{z_{ik}}}{\sigma_i^2} \mathbf{w}_{ik} \mathbf{w}_{ik}^{\mathrm{T}} \right) \mathbf{x} \right) \\
&\propto \mathcal{N}\left(\mathbf{x}; \mathbf{0}, \left(\epsilon \mathbf{I} + \sum_{i=1}^{N} \mathbf{W}_i \mathbf{Z}_i \mathbf{W}_i^{\mathrm{T}} \right)^{-1} \right),
\end{aligned} \tag{3.8}$$

where the $\mathbf{w}_{ik}$ are defined such that $\mathbf{w}_{ik}^{\mathrm{T}}\mathbf{x}$ is the result of applying filter $\mathbf{w}_i$ to the k^{th} clique of the image $\mathbf{x}$. $\mathbf{Z}_i = \operatorname{diag}\{s_{z_{ik}}/\sigma_i^2\}$ are diagonal matrices with entries for each clique, and $\mathbf{W}_i$ are filter matrices that correspond to a convolution of the image with filter $\mathbf{w}_i$, i.e. $\mathbf{W}_i^{\mathrm{T}}\mathbf{x} = [\mathbf{w}_{i1}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{w}_{iK}^{\mathrm{T}}\mathbf{x}]^{\mathrm{T}} = [\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(1)}, \ldots, \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(K)}]^{\mathrm{T}}$. The broad Gaussian factor $e^{-\epsilon\|\mathbf{x}\|^2/2}$ guarantees positive definiteness of the covariance matrix. Since the conditional distribution of the image given the scale indices in Eq. (3.8) is Gaussian, the only difficulty for sampling arises from the fact that the (inverse) covariance matrix is huge when the image is large, which prevents an explicit Cholesky decomposition as used in Welling et al. [2003]. Levi [2009] showed that this can be circumvented by rewriting the covariance as the matrix product

$$\boldsymbol{\Sigma} = \left(\epsilon\mathbf{I} + \sum_{i=1}^{N}\mathbf{W}_i\mathbf{Z}_i\mathbf{W}_i^{\mathrm{T}}\right)^{-1} = \left([\mathbf{W}_1, \ldots, \mathbf{W}_N, \mathbf{I}]\begin{bmatrix}\mathbf{Z}_1 & & \cdots & 0\\ & \ddots & & \vdots\\ \vdots & & \mathbf{Z}_N & \\ 0 & \cdots & & \epsilon\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{W}_1^{\mathrm{T}}\\ \vdots\\ \mathbf{W}_N^{\mathrm{T}}\\ \mathbf{I}\end{bmatrix}\right)^{-1} = \left(\mathbf{W}\mathbf{Z}\mathbf{W}^{\mathrm{T}}\right)^{-1} \tag{3.9}$$

and sample $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to obtain a sample $\mathbf{x}$ from $p(\mathbf{x}|\mathbf{z}; \boldsymbol{\Theta})$ by solving the least-squares problem

$$\mathbf{W}\mathbf{Z}\mathbf{W}^{\mathrm{T}}\mathbf{x} = \mathbf{W}\sqrt{\mathbf{Z}}\mathbf{y}. \tag{3.10}$$

By using the well-known property

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \Rightarrow \mathbf{A}\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{A}\mathbf{I}\mathbf{A}^{\mathrm{T}}), \tag{3.11}$$

it follows that

$$\begin{aligned}\mathbf{x} = \left(\mathbf{W}\mathbf{Z}\mathbf{W}^{\mathrm{T}}\right)^{-1}\mathbf{W}\sqrt{\mathbf{Z}}\mathbf{y} &\sim \mathcal{N}\left(\mathbf{x}; \mathbf{0}, \left(\left(\mathbf{W}\mathbf{Z}\mathbf{W}^{\mathrm{T}}\right)^{-1}\mathbf{W}\sqrt{\mathbf{Z}}\right)\mathbf{I}\left(\left(\mathbf{W}\mathbf{Z}\mathbf{W}^{\mathrm{T}}\right)^{-1}\mathbf{W}\sqrt{\mathbf{Z}}\right)^{\mathrm{T}}\right)\\ &\sim \mathcal{N}\left(\mathbf{x}; \mathbf{0}, \left(\mathbf{W}\mathbf{Z}\mathbf{W}^{\mathrm{T}}\right)^{-1}\right)\end{aligned} \tag{3.12}$$

is indeed a valid sample from the conditional distribution as derived in Eq. (3.8). Since solving this sparse linear system of equations is much more efficient than a Cholesky decomposition, this leads to an efficient sampling procedure with rapid mixing (see Fig. 5.18).

3.1.1 Conditional sampling

In subsequent chapters, we will make use of conditional sampling in order to avoid extreme values at the less constrained boundary pixels [Norouzi et al., 2009] during learning and model analysis, or to perform inpainting of missing pixels given the known ones. In particular, we sample the pixels $\mathbf{x}_A$ given fixed $\mathbf{x}_B$ and scales $\mathbf{z}$ according to the conditional Gaussian distribution

$$p(\mathbf{x}_A|\mathbf{x}_B, \mathbf{z}; \boldsymbol{\Theta}), \tag{3.13}$$

where A and B denote the index sets of the respective pixels. Without loss of generality, we assume that

$$\mathbf{x} = \begin{bmatrix}\mathbf{x}_A\\ \mathbf{x}_B\end{bmatrix}, \quad \boldsymbol{\Sigma} = \left(\mathbf{W}\mathbf{Z}\mathbf{W}^{\mathrm{T}}\right)^{-1} = \begin{bmatrix}\mathbf{A} & \mathbf{C}\\ \mathbf{C}^{\mathrm{T}} & \mathbf{B}\end{bmatrix}^{-1}, \tag{3.14}$$

where the square sub-matrix $\mathbf{A}$ has as many rows and columns as the vector $\mathbf{x}_A$ has elements; the same applies to matrix $\mathbf{B}$ with respect to vector $\mathbf{x}_B$. The size of the matrix $\mathbf{C}$ is therefore determined by both $\mathbf{x}_A$ and $\mathbf{x}_B$. The conditional distribution of interest can now be derived as

$$\begin{aligned} p(\mathbf{x}_A|\mathbf{x}_B,\mathbf{z};\boldsymbol{\Theta}) &\propto p(\mathbf{x}|\mathbf{z};\boldsymbol{\Theta}) \\ &\propto \exp\left(-\frac{1}{2}\begin{bmatrix}\mathbf{x}_A\\ \mathbf{x}_B\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}\mathbf{A} & \mathbf{C}\\ \mathbf{C}^{\mathrm{T}} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{x}_A\\ \mathbf{x}_B\end{bmatrix}\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\mathbf{x}_A^{\mathrm{T}}\mathbf{A}\mathbf{x}_A + 2\mathbf{x}_A^{\mathrm{T}}\mathbf{C}\mathbf{x}_B + \mathbf{x}_B^{\mathrm{T}}\mathbf{B}\mathbf{x}_B\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\mathbf{x}_A + \mathbf{A}^{-1}\mathbf{C}\mathbf{x}_B\right)^{\mathrm{T}}\mathbf{A}\left(\mathbf{x}_A + \mathbf{A}^{-1}\mathbf{C}\mathbf{x}_B\right)\right) \\ &\propto \mathcal{N}\left(\mathbf{x}_A; -\mathbf{A}^{-1}\mathbf{C}\mathbf{x}_B, \mathbf{A}^{-1}\right). \end{aligned} \tag{3.15}$$

The matrices $\mathbf{A}$ and $\mathbf{C}$ are given by the appropriate sub-matrices of $\mathbf{W}_i$ and $\mathbf{Z}_i$, and allow for the same efficient sampling scheme. The mean $\boldsymbol{\mu} = -\mathbf{A}^{-1}\mathbf{C}\mathbf{x}_B$ can also be computed by solving the least squares problem $\mathbf{A}\boldsymbol{\mu} = -\mathbf{C}\mathbf{x}_B$ and does not require matrix inversion of $\mathbf{A}$.

Sampling the conditional distribution of scales $p(\mathbf{z}|\mathbf{x}_A,\mathbf{x}_B;\boldsymbol{\Theta}) = p(\mathbf{z}|\mathbf{x};\boldsymbol{\Theta})$ remains as before.

3.1.2 Convergence analysis

Assessing convergence of MCMC samplers is a long-standing issue that unfortunately has no definitive solution. Convergence to the equilibrium distribution is important, because only fair samples should be used to estimate the quantities of interest. Although our auxiliary-variable Gibbs sampler mixes rapidly, as can be seen in Figure 5.18, it is still advantageous to use a quantitative measure for monitoring convergence.

To that end, we use the popular approach by Gelman and Rubin [1992], which has also found its way into the well-received textbook *Bayesian Data Analysis* [Gelman, Carlin, Stern, and Rubin, 2004]. It relies on running several Markov chains in parallel which are initialized with different over-dispersed starting points. The basic idea is to compare the within-sequence variance W and the between-sequence variance B of scalar estimands of interest (we use the model energy[2]) – and to declare convergence when W roughly equals B. Concretely, convergence is determined by estimating the potential scale reduction (EPSR)

$$\hat{R} = \sqrt{\left((n-1)W + B\right)/\left(nW\right)}, \tag{3.16}$$

where n is the number of iterations per chain. If $\hat{R}$ is large, further iterations will probably improve our inference about the scalar estimands. If $\hat{R}$ is near 1, however, we can assume approximate convergence; we stop the sampler when $\hat{R} < 1.1$ in particular. Starting the chains at different over-dispersed starting points is crucial for this method to work. For computing $\hat{R}$, we always conservatively discard the first half of the samples. We refer to Gelman and Rubin [1992]; Gelman et al. [2004] for details.

[2] The model energy is the negative log of Eq. (3.1), ignoring the normalization constant $Z(\boldsymbol{\Theta})$.

4 Learning Heavy-tailed Distributions

The heavy-tailed marginal distributions of natural image derivatives (and even random zero-mean filters) motivate the use of heavy-tailed potentials in MRFs. In this chapter, we investigate learning the parameters of such heavy-tailed potentials in a simple univariate setting, before we tend to learning the more complicated MRF models in Chapter 5. The results of the experiments in this chapter indicate that score matching is rather unsuitable for estimating the parameters of heavy-tailed distributions from "noisy" training data.

In the following we will use

$$\tilde{J}(\boldsymbol{\Theta}) = \sum_{t=1}^{T} S(x^{(t)}; \boldsymbol{\Theta}) \tag{4.1}$$

with

$$S(x; \boldsymbol{\Theta}) = \psi'(x; \boldsymbol{\Theta}) + \frac{1}{2}\psi(x; \boldsymbol{\Theta})^2, \qquad \psi(x; \boldsymbol{\Theta}) = \frac{\mathrm{d}}{\mathrm{d}x} \log \phi(x; \boldsymbol{\Theta}) \tag{4.2}$$

as the score matching objective function for the univariate parametric distribution $\phi(x; \boldsymbol{\Theta})$, given i.i.d. training data $x^{(1)}, \ldots, x^{(T)}$.

Gaussian distribution. It is illuminating to first take a look at the SM estimator for the Gaussian distribution, which Hyvärinen [2005] showed to coincide with ML estimation. The Gaussian distribution seems particularly suitable for SM since the gradient of log-density is a straight line, i.e. the log-pdf is very smooth. The SM objective $\tilde{J}_{\mathrm{G}}(\sigma)$ for the zero-mean Gaussian distribution

$$\phi_{\mathrm{G}}(x; \sigma) \propto \mathcal{N}(x; 0, \sigma^2) \tag{4.3}$$

is given by

$$S_{\mathrm{G}}(x; \sigma) = \psi'_{\mathrm{G}}(x; \sigma) + \frac{1}{2}\psi_{\mathrm{G}}(x; \sigma)^2 = -\frac{1}{\sigma^2} + \frac{x^2}{2\sigma^4}. \tag{4.4}$$

Figure 4.1(a) shows a plot of $S_{\mathrm{G}}(x; \sqrt{0.5})$. Note that function values further away from the mode are increasing rapidly because the tails of the Gaussian are falling off quickly. Hence, values of x at the tails will substantially contribute to the cost function $\tilde{J}_{\mathrm{G}}(\sigma)$. Also note that S_{G} is a smooth function, i.e. small changes in x do not result in big changes in S_{G}.

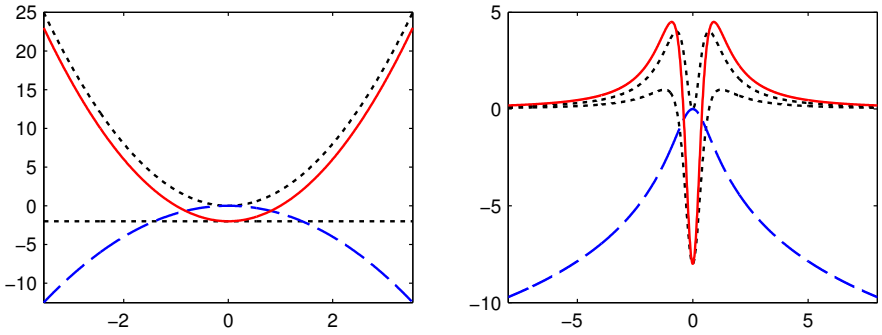
4.1 Student-t Distribution

The heavy-tailed Student-t distribution is popular in the literature and has been used by Roth and Black [2009] in the FoE. We define the distribution here as

$$\phi_{\mathrm{St}}(x; \sigma, \alpha) = \left(1 + \frac{x^2}{2\sigma^2}\right)^{-\alpha} \tag{4.5}$$

with parameters σ and α. The SM estimator is given by

$$S_{\mathrm{St}}(x; \sigma, \alpha) = \psi'_{\mathrm{St}}(x; \sigma) + \frac{1}{2}\psi_{\mathrm{St}}(x; \sigma)^2 = \frac{2\alpha(x^2 - 2\sigma^2)}{(2\sigma^2 + x^2)^2} + \frac{1}{2}\left(-\frac{2\alpha x}{2\sigma^2 + x^2}\right)^2. \tag{4.6}$$



(a) Gaussian, $S_{\mathrm{G}}(x;\sqrt{0.5})$ (b) Student-t, $S_{\mathrm{St}}(x;0.5,2)$

Figure 4.1: Contribution $S(x;\boldsymbol{\Theta})$ (solid red) to the SM objective $\tilde{J}(\boldsymbol{\Theta})$. The black dotted lines denote $\frac{1}{2}\psi(x;\boldsymbol{\Theta})^2$ and $\psi'(x;\boldsymbol{\Theta})$; the log-density is shown in dashed blue.

Figure 4.1(b) shows a plot of $S_{\mathrm{St}}(x;0.5,2)$; the function varies greatly with x around zero, i.e. small changes in x result in big changes of the function value. This suggests that SM estimation will be susceptible to noise in the training data. Note that function values further away from the mode are essentially negligible for the SM cost function. This is to be expected for all heavy-tailed densities with a somewhat sharp peak.

This property is amplified when the density becomes more peaky, i.e. σ becomes smaller. Then, S_{St} can take on its extreme values in a very small interval, while values outside this interval are essentially not contributing to the cost function. The minimum is always at $x_{\min}=0$ and equal to

$$S_{\mathrm{St}}(x_{\min};\sigma,\alpha)=-\frac{\alpha}{\sigma^2}. \tag{4.7}$$

The two maxima are at $x_{\max}=\pm\left(\sqrt{2}\sqrt{(\alpha+1)(\alpha+3)}\sigma\right)\left(\alpha+1\right)^{-1}$ and take on the value of

$$S_{\mathrm{St}}(x_{\max};\sigma,\alpha)=\frac{\alpha(\alpha+1)^2}{4\sigma^2(\alpha+2)}. \tag{4.8}$$

Hence, for $\sigma\to 0$ and moderate values of α, the function $S_{\mathrm{St}}(x;\sigma,\alpha)$ goes to $-\infty$ and $+\infty$ in a very small interval around $x=0$.

We observed this to be a problem in practice when using SM to learn the parameters of a Student-t potential in a pairwise MRF (from natural image patches). When σ approached 0 the magnitude of the gradient "exploded" at some point, "catapulting" the parameters far away from their previous value. We were unable to solve this problem by choosing a particularly small learning rate. The general problem is that the "peakedness" of the Student-t distribution cannot be controlled, so that it can essentially become a δ-like function with heavy tails. We will argue in the next section that these kinds of densities are problematic to estimate with score matching.

4.2 Gaussian Scale Mixtures

By choosing appropriate scales, Gaussian Scale Mixtures (GSMs) allow more control over the shape that the mixture model can take. We exploit this to conduct experiments with increasingly more heavy-tailed distribution shapes.

Similar to Chapter 3, we define the GSM as

$$\phi_{\mathrm{GSM}}(x;\boldsymbol{\alpha})=\sum_{j=1}^{J}\beta_j\cdot\mathcal{N}(x;0,\sigma^2/s_j) \tag{4.9}$$

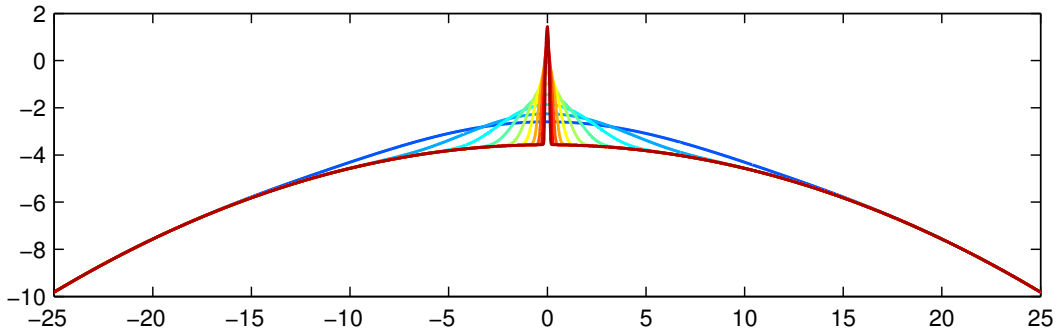


Figure 4.2: $\log \phi_{\mathrm{GSM}}(x;\boldsymbol{\alpha})$, $\gamma = 1$ (blue), $\ldots$, 10 (dark red).

with normalized mixture weight

$$\beta_j = \frac{\exp(\alpha_j)}{\sum_{j'=1}^{J} \exp(\alpha_{j'})} \tag{4.10}$$

for the Gaussian component with scale s_j and base variance σ^2. The SM estimator can easily be derived and is given by

$$\begin{aligned} S_{\mathrm{GSM}}(x;\boldsymbol{\alpha}) &= \psi'_{\mathrm{GSM}}(x;\boldsymbol{\alpha}) + \frac{1}{2}\psi_{\mathrm{GSM}}(x;\boldsymbol{\alpha})^2 \\ &= \frac{\phi''_{\mathrm{GSM}}(x;\boldsymbol{\alpha})}{\phi_{\mathrm{GSM}}(x;\boldsymbol{\alpha})} - \left(\frac{\phi'_{\mathrm{GSM}}(x;\boldsymbol{\alpha})}{\phi_{\mathrm{GSM}}(x;\boldsymbol{\alpha})}\right)^2 + \frac{1}{2}\left(\frac{\phi'_{\mathrm{GSM}}(x;\boldsymbol{\alpha})}{\phi_{\mathrm{GSM}}(x;\boldsymbol{\alpha})}\right)^2. \end{aligned} \tag{4.11}$$

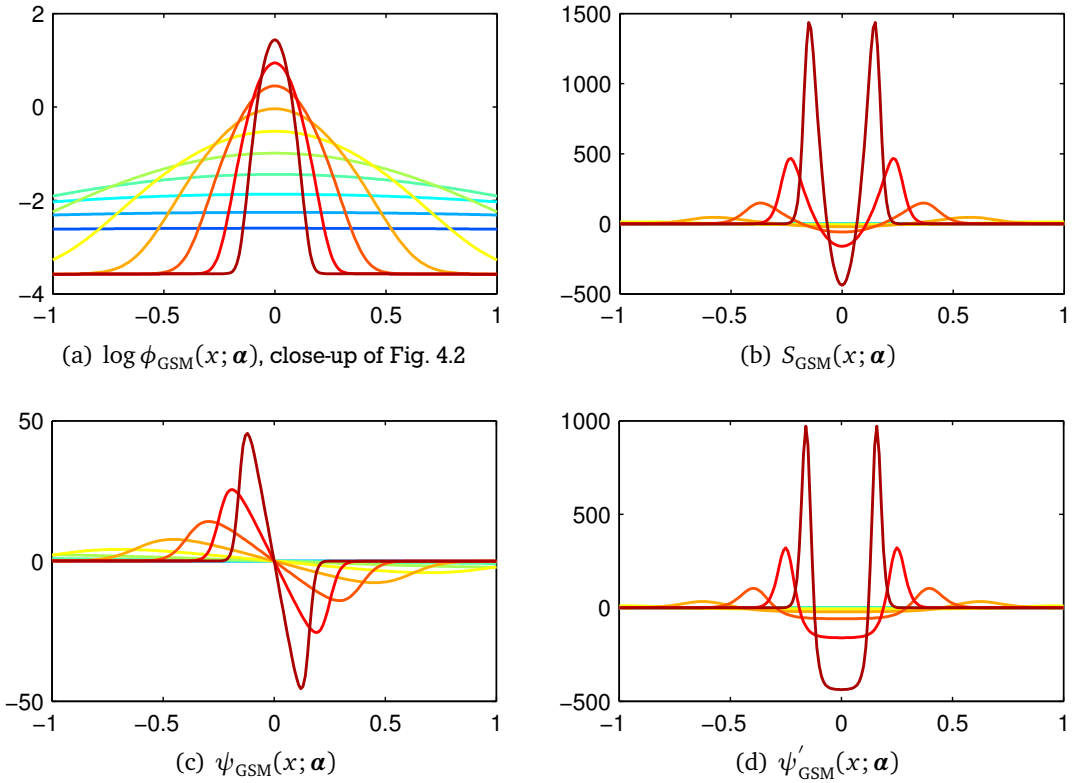
We let SM compete against maximum likelihood (ML), which does not require sampling here because the GSM from Eq. (4.9) integrates to 1. Hence, we can easily compute the log-likelihood function

$$\ell_{\mathrm{GSM}}(\boldsymbol{\alpha}) = \log \prod_{t=1}^{T} \phi_{\mathrm{GSM}}(x^{(t)};\boldsymbol{\alpha}) = \sum_{t=1}^{m} \log \phi_{\mathrm{GSM}}(x^{(t)};\boldsymbol{\alpha}) \tag{4.12}$$

and its derivatives w.r.t. the model parameters $\boldsymbol{\alpha}$. We minimized the SM objective function and the negative log-likelihood using conjugate gradients, based on the implementation of Rasmussen [2006].

We set $\sigma^2 = 50$, $\boldsymbol{\beta} = [0.5, 0.5]^{\mathrm{T}}$, and $s = \left(1, e^{\gamma}\right)$, varying γ to alter the shape of the mixture model. Figures 4.2 and 4.3(a) show how the shape of the GSM becomes more peaky with increasing γ, substantially influencing the function $S_{\mathrm{GSM}}(x;\boldsymbol{\alpha})$ whose "oscillations" become steeper and increase magnitude with larger values of γ (Fig. 4.3(b)). It could be argued that the almost δ-like distribution shapes for larger values of γ are rarely used in practice. We however find similar expert shapes in the FoE which lend to good generative properties (e.g. Figure 5.15(a)).

We performed experiments for $\gamma = 1, \ldots, 10$, where we generated 100000 samples from the model and used SM and ML to estimate $\boldsymbol{\alpha}$. Since the weights sum to one, the task was essentially to estimate a single parameter. We repeated each experiment ten times with different samples and start weights for the conjugate gradient method. The estimation error was evaluated as the KL-divergence between the ground truth distribution and the estimated GSM over the interval from -25 to 25 with a step size of 0.01. Fig. 4.4 shows the outcome of four kinds of experiments we performed. The plots show the average error over the 10 runs for every value of γ, where error-bars denote the minimal and maximal observed values.



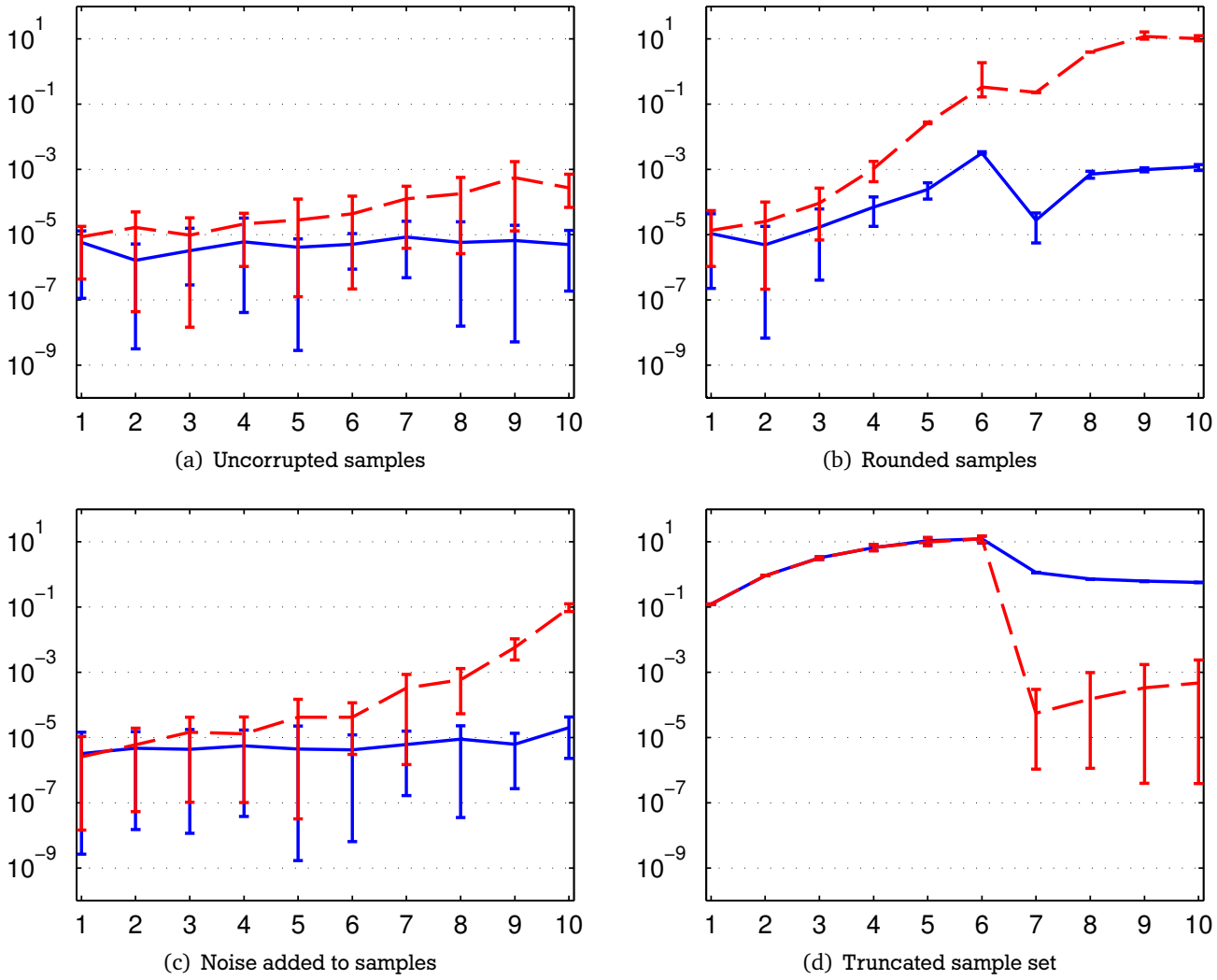
(a) $\log \phi_{\mathrm{GSM}}(x;\boldsymbol{\alpha})$, close-up of Fig. 4.2 (b) $S_{\mathrm{GSM}}(x;\boldsymbol{\alpha})$

(c) $\psi_{\mathrm{GSM}}(x;\boldsymbol{\alpha})$ (d) $\psi'_{\mathrm{GSM}}(x;\boldsymbol{\alpha})$

Figure 4.3: GSM model with uniform weights for $\gamma = 1$ (blue), ..., 10 (dark red). See text for description.

Experiments. The first experiment (Fig. 4.4(a)) was carried out just as described above. It can be seen that the estimation error for SM slowly rises with γ, whereas the error made by ML stays roughly constant. For the second experiment (Fig. 4.4(b)), we rounded the samples to the nearest integer. SM performs significantly worse, especially for larger values of γ. This is to be expected when looking at the shape of $S_{\mathrm{GSM}}(x;\boldsymbol{\alpha})$ (Fig. 4.3(b)). ML also performs 1-2 orders of magnitude worse, but the error is not growing as rapidly with increasing γ. For $\gamma > 7$ the error made by SM is greater than 1, whereas ML makes an error of about 10^{-3}. For the next experiment (Fig. 4.4(c)), we added zero-mean Gaussian noise with variance $\sigma^2 = 1/1600$ to the generated samples. While the performance of ML hardly changes at all, SM is significantly affected by this infinitesimal amount of noise for larger values of γ. In the last experiment (Fig. 4.4(d)), we discarded all samples outside the interval $(-1, 1)$. The point is to demonstrate that SM is not using the discarded samples for larger values of γ. It can be seen that SM performs just as in the first experiment for $\gamma = 7, \ldots, 10$ while ML performs much worse.

In summary, score matching's susceptibility to "noise" in the training data may pose a serious problem for real world applications, especially when using very heavy-tailed densities which we will argue are required to adequately model natural images with Markov Random Fields (Chapter 5). Image intensity values are often rounded or computed from rounded RGB values, and noise in natural images cannot entirely be avoided. Score matching's deficiencies for heavy-tailed distributions in these simple univariate experiments foreshadow its poor performance in the context of MRFs (Chapter 5).



(a) Uncorrupted samples

(b) Rounded samples

(c) Noise added to samples

(d) Truncated sample set

Figure 4.4: Experimental results for the four kinds of experiments we performed, please see the text for details. The horizontal axis indicates the value of γ and the vertical axis the estimation error on a logarithmic scale. The estimation results for maximum likelihood are shown in solid blue, whereas score matching performance is depicted with red dashed lines.

5 Learning MRFs and Generative Evaluation

Evaluation of MRF priors often takes place in the context of a particular application – image denoising in case of MRF models of natural images [Roth and Black, 2009; Tappen et al., 2003] – and is also dependent on the specific inference method used. Additionally, probabilistically trained generative models have often required ad-hoc modifications to perform well in practice [Roth and Black, 2009]. Hence, evaluation in a setting like this at best allows indirect conclusions about the inherent quality of the model. Despite these apparent disadvantages, it is largely the only choice: computing the likelihood of MRFs is usually intractable, and likelihood bounds are often not tight enough to allow comparison of different models (as in our case).

Our efficient auxiliary-variable Gibbs sampler allows us to evaluate the generative properties of the model in a timely manner by means of drawing samples – independent of any application and inference method. This approach of evaluation was already proposed by Zhu and Mumford [1997], but has been largely ignored ever since due to its computational difficulty.

After introducing and deriving the "competing" estimators, we train MRF models with contrastive divergence and score matching, and compare their generative properties. In particular, we compare the marginals of the MRF features (i.e., filters) and use the marginal KL-divergence as a quantitative measure. We consider pairwise MRF models first, which remain popular until today due to their simplicity, before we turn to the more powerful Fields of Experts. Section 5.4 addresses the problem of boundary handling in MRF models: we train and evaluate MRFs with alternative boundary handling and obtain our best generative models – which compare favorably to other popular MRF priors that show poor generative properties despite their good application performance in the context of MAP estimation.

In all experiments, we used stochastic gradient descent (SGD, cf. Bottou [2004]) with a mini-batch size of 20 image patches to train the MRFs. Unless otherwise noted, the GSM weights of the clique potentials have been initialized uniformly and the filter coefficients in the FoE models have been initialized from a zero-mean unit-variance Gaussian. Our training set contained 1000 training image patches of 30×30 pixels (Fig. 5.5(a)), sampled uniformly from a subset[1] of the training images of the Berkeley image segmentation dataset [Martin et al., 2001]. We converted the color images to the YCbCr color space using the MATLAB command `rgb2ycbcr` and used the Y channels as the gray scale images. Unfortunately, we only realized later that we incorrectly converted the images not using the full range of luminance levels. All findings here should nevertheless equally apply to images using the full range of intensity values.

Until Section 5.4, the marginal statistics are computed using 10000 30×30 image patches and samples drawn from the MRF. We did not employ the EPSR convergence criteria (Section 3.1.2) for simplicity, instead always used 20 iterations of the Gibbs sampler to generate a single sample. In order to quantitatively compare the marginal statistics without problems, we always added one count to each of the 401 bins of the histograms[2]. The marginal KL-divergence is then computed between the multinomial distributions which are given by the normalized histograms.

5.1 Deriving the Estimators

For the following derivations, it will be advantageous to look at the FoE model density

$$p(\mathbf{x};\boldsymbol{\Theta}) = \frac{1}{Z(\boldsymbol{\Theta})} \exp\left(-E(\mathbf{x};\boldsymbol{\Theta})\right) \tag{5.1}$$

[1] See `http://www.gris.informatik.tu-darmstadt.de/~sroth/research/foe/train.txt` for the list of file names.

[2] This can be interpreted as using a Dirichlet prior.

in terms of its energy

$$E(\mathbf{x};\boldsymbol{\Theta}) = \frac{\epsilon}{2}\|\mathbf{x}\|^2 - \sum_{k=1}^{K}\sum_{i=1}^{N}\log\phi\left(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i\right), \tag{5.2}$$

and to write the GSM experts

$$\phi(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i) = \frac{1}{\sum_{j=1}^{J}\omega_{ij}}\sum_{j=1}^{J}\omega_{ij}\cdot\mathcal{N}(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};0,\sigma_i^2/s_j) = (\boldsymbol{\omega}_i^{\mathrm{T}}\mathbf{1})^{-1}\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}) \tag{5.3}$$

as vector products for conciseness of notation, where $\boldsymbol{\omega}_i = [\omega_{i1},\ldots,\omega_{iJ}]^{\mathrm{T}}$ with $\omega_{ij} = \exp(\alpha_{ij})$, $\mathbf{1}$ denotes the J-dimensional 1-vector, and $\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}) = \left\{\mathcal{N}(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};0,\sigma_i^2/s_j)\,|\,j=1,\ldots,J\right\}$ is a vector-valued function where we denote vectors of element-wise derivatives with $\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})$, $\boldsymbol{\varphi}_i''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})$, etc.

Maximum likelihood. As already introduced in Section 2.3.1, we want to maximize the log-likelihood function

$$\ell(\boldsymbol{\Theta}) = \log\prod_{t=1}^{T}p(\mathbf{x}^{(t)};\boldsymbol{\Theta}) = \sum_{t=1}^{T}\log p(\mathbf{x}^{(t)};\boldsymbol{\Theta}) = \sum_{t=1}^{T} -\log Z(\boldsymbol{\Theta}) - E(\mathbf{x}^{(t)};\boldsymbol{\Theta}), \tag{5.4}$$

where $\mathbf{X} = \{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(T)}\}$ is a set of i.i.d. training data. We do this by taking the derivatives (cf. Eq. (2.15))

$$\begin{aligned}\frac{\partial\ell(\boldsymbol{\Theta})}{\partial\boldsymbol{\Theta}} &= -T\frac{\partial\log Z(\boldsymbol{\Theta})}{\partial\boldsymbol{\Theta}} - \sum_{t=1}^{T}\frac{\partial E(\mathbf{x}^{(t)};\boldsymbol{\Theta})}{\partial\boldsymbol{\Theta}}\\ &= T\left[\left\langle\frac{\partial E(\mathbf{x};\boldsymbol{\Theta})}{\partial\boldsymbol{\Theta}}\right\rangle_p - \left\langle\frac{\partial E(\mathbf{x};\boldsymbol{\Theta})}{\partial\boldsymbol{\Theta}}\right\rangle_{\mathbf{X}}\right]\end{aligned} \tag{5.5}$$

w.r.t. the model parameters $\boldsymbol{\Theta} = \{\mathbf{w}_i,\boldsymbol{\alpha}_i|i=1,\ldots,N\}$, where we rely on sampling to approximate the expected derivative w.r.t. the model. For contrastive divergence, we initialize the samples with the training data $\mathbf{X}$ and only take a few MCMC steps, instead of computing relatively expensive equilibrium samples.

Concretely, the derivative w.r.t. the GSM parameters is

$$\begin{aligned}\frac{\partial E(\mathbf{x};\boldsymbol{\Theta})}{\partial\alpha_{ij}} &= -\sum_{k=1}^{K}\frac{\partial\log\phi(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i)}{\partial\alpha_{ij}}\\ &= -\sum_{k=1}^{K}\frac{\partial\left(-\log(\boldsymbol{\omega}_i^{\mathrm{T}}\mathbf{1}) + \log(\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}))\right)}{\partial\alpha_{ij}}\\ &= \sum_{k=1}^{K}\frac{\omega_{ij}}{\boldsymbol{\omega}_i^{\mathrm{T}}\mathbf{1}} - \frac{\omega_{ij}\mathcal{N}(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};0,\sigma_i^2/s_j)}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\\ &= \frac{\omega_{ij}}{\boldsymbol{\omega}_i^{\mathrm{T}}\mathbf{1}}\left[K - \sum_{k=1}^{K}\frac{\mathcal{N}(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};0,\sigma_i^2/s_j)}{\phi(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i)}\right]\end{aligned} \tag{5.6}$$

and in case of the FoE we also need the derivative w.r.t. all filter coefficients:

$$\begin{aligned}\frac{\partial E(\mathbf{x};\boldsymbol{\Theta})}{\partial \mathrm{w}_{im}} &= -\sum_{k=1}^{K} \frac{\partial \log \phi(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i)}{\partial \mathrm{w}_{im}} \\ &= -\sum_{k=1}^{K} \frac{\phi'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i)}{\phi(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i)} \cdot \frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial \mathrm{w}_{im}} \\ &= -\sum_{k=1}^{K} \frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})} \cdot \left[\mathbf{x}_{(k)}\right]_m.\end{aligned} \tag{5.7}$$

Score matching. We want to minimize the score matching cost function

$$\tilde{J}(\boldsymbol{\Theta}) = \sum_{t=1}^{T}\sum_{d=1}^{D} \psi_d'(\mathbf{x}^{(t)};\boldsymbol{\Theta}) + \frac{1}{2}\psi_d(\mathbf{x}^{(t)};\boldsymbol{\Theta})^2 \tag{5.8}$$

w.r.t. $\boldsymbol{\Theta}$, where each training example $\mathbf{x}^{(t)} \in \mathbb{R}^D$. The objective function comprises the score function

$$\begin{aligned}\psi_d(\mathbf{x};\boldsymbol{\Theta}) &= \frac{\partial \log p(\mathbf{x};\boldsymbol{\Theta})}{\partial x_d} \\ &= -\epsilon x_d + \sum_{\substack{k=1 \\ x_d \in \mathbf{x}_{(k)}}}^{K} \sum_{i=1}^{N} \frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d} \cdot \frac{\phi'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i)}{\phi(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i)} \\ &= -\epsilon x_d + \sum_{\substack{k=1 \\ x_d \in \mathbf{x}_{(k)}}}^{K} \sum_{i=1}^{N} \frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d} \cdot \frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\end{aligned} \tag{5.9}$$

and its derivative

$$\begin{aligned}\psi_d'(\mathbf{x};\boldsymbol{\Theta}) &= \frac{\partial^2 \log p(\mathbf{x};\boldsymbol{\Theta})}{\partial x_d^2} \\ &= -\epsilon + \sum_{\substack{k=1 \\ x_d \in \mathbf{x}_{(k)}}}^{K} \sum_{i=1}^{N} \left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d}\right]^2 \left[\frac{\phi''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i)}{\phi(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i)} - \left(\frac{\phi'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i)}{\phi(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i)}\right)^2\right] \\ &= -\epsilon + \sum_{\substack{k=1 \\ x_d \in \mathbf{x}_{(k)}}}^{K} \sum_{i=1}^{N} \left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d}\right]^2 \left[\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})} - \left(\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\right)^2\right].\end{aligned} \tag{5.10}$$

We also compute the derivatives

$$\frac{\partial \tilde{J}(\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} = \sum_{t=1}^{T}\sum_{d=1}^{D} \frac{\partial \psi_d'(\mathbf{x}^{(t)};\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} + \psi_d(\mathbf{x}^{(t)};\boldsymbol{\Theta}) \frac{\partial \psi_d(\mathbf{x}^{(t)};\boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \tag{5.11}$$

w.r.t. the model parameters $\boldsymbol{\Theta} = \{\mathbf{w}_i, \boldsymbol{\alpha}_i | i = 1, \ldots, N\}$, which require the following:

$$\frac{\partial \psi_d(\mathbf{x};\boldsymbol{\Theta})}{\partial \alpha_{ij}} = \sum_{\substack{k=1 \\ x_d \in \mathbf{x}_{(k)}}}^{K} \left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d}\right] \left[\frac{\omega_{ij}\mathcal{N}'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};0,\sigma_i^2/s_j)}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})} - \frac{\omega_{ij}\mathcal{N}(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};0,\sigma_i^2/s_j) \cdot \boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\left(\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})\right)^2}\right] \tag{5.12}$$

$$\frac{\partial \psi_d'(\mathbf{x};\boldsymbol{\Theta})}{\partial \alpha_{ij}} = \sum_{\substack{k=1 \\ x_d \in \mathbf{x}_{(k)}}}^{K} \left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d}\right]^2 \left[\frac{\omega_{ij}\mathcal{N}''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};0,\sigma_i^2/s_j)}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})} + \frac{2\omega_{ij}\mathcal{N}(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};0,\sigma_i^2/s_j)\cdot\left(\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})\right)^2}{\left(\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})\right)^3} - \frac{2\omega_{ij}\mathcal{N}'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};0,\sigma_i^2/s_j)\cdot\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}) + \omega_{ij}\mathcal{N}(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};0,\sigma_i^2/s_j)\cdot\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\left(\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})\right)^2}\right] \tag{5.13}$$

$$\frac{\partial \psi_d(\mathbf{x};\boldsymbol{\Theta})}{\partial \mathrm{w}_{im}} = \sum_{\substack{k=1 \\ x_d \in \mathbf{x}_{(k)}}}^{K} \left[\frac{\partial^2 \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d \partial \mathrm{w}_{im}}\right]\left[\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\right] + \left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d}\right]\left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial \mathrm{w}_{im}}\right]\left[\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})} - \left(\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\right)^2\right] \tag{5.14}$$

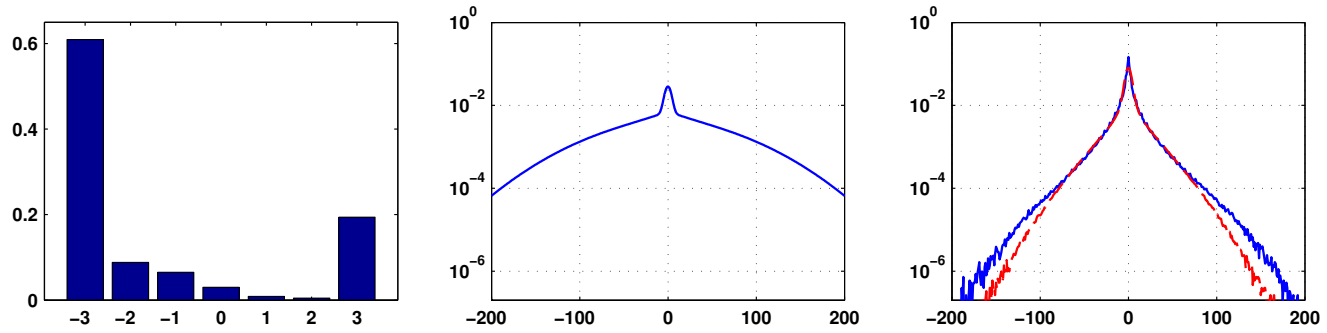
$$= \sum_{\substack{k=1 \\ x_d \in \mathbf{x}_{(k)}}}^{K} \left[\frac{\partial^2 \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial \mathrm{w}_{im} \partial x_d}\right]\left[\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\right] + \left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d}\right]\left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial \mathrm{w}_{im}}\right]\left[\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})} - \left(\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\right)^2\right] \tag{5.15}$$

$$= \sum_{\substack{k=1 \\ x_d \in \mathbf{x}_{(k)}}}^{K} \left[\frac{\partial \left[\mathbf{x}_{(k)}\right]_m}{\partial x_d}\right]\left[\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\right] + \left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d}\right]\left[\mathbf{x}_{(k)}\right]_m\left[\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})} - \left(\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\right)^2\right] \tag{5.16}$$

$$\frac{\partial \psi_d'(\mathbf{x};\boldsymbol{\Theta})}{\partial \mathrm{w}_{im}} = \sum_{\substack{k=1 \\ x_d \in \mathbf{x}_{(k)}}}^{K} 2\left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d}\right]\left[\frac{\partial^2 \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d \partial \mathrm{w}_{im}}\right]\left[\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})} - \left(\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\right)^2\right] + \left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d}\right]^2\left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial \mathrm{w}_{im}}\right]\left[\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})} + 2\left(\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\right)^3 - 3\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})\cdot\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\left(\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})\right)^2}\right] \tag{5.17}$$

$$= \sum_{\substack{k=1 \\ x_d \in \mathbf{x}_{(k)}}}^{K} 2\left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d}\right]\left[\frac{\partial \left[\mathbf{x}_{(k)}\right]_m}{\partial x_d}\right]\left[\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})} - \left(\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\right)^2\right] + \left[\frac{\partial \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}}{\partial x_d}\right]^2\left[\mathbf{x}_{(k)}\right]_m\left[\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})} + 2\left(\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}\right)^3 - 3\frac{\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i'(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})\cdot\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i''(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})}{\left(\boldsymbol{\omega}_i^{\mathrm{T}}\boldsymbol{\varphi}_i(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})\right)^2}\right] \tag{5.18}$$

The required computations are obviously more complicated in comparison to ML, but they do not require to draw samples from the MRF.



(a) GSM weights (b) GSM potential (c) Marginals, KLD = 0.0442

Figure 5.1: Learned pairwise MRF using CD-ML and scales from e^{-3} to e^{3}. *(a)* GSM weight distribution of log-scales. *(b)* Semi-log plot of GSM potential. *(c)* Marginal semi-log derivative histogram for natural images (solid blue) and samples drawn from the pairwise MRF (dashed red).

5.2 Pairwise MRFs

We trained pairwise MRFs with fixed horizontal and vertical derivative filters and a single GSM potential. We set the GSM base variance to the empirical variance of the image derivatives of the set of training image patches ($\sigma^2 \approx 250$).

5.2.1 Natural images

For a first experiment we chose scales

$$s = (e^{-3}, e^{-2}, e^{-1}, e^{0}, e^{1}, e^{2}, e^{3}) \tag{5.19}$$

and estimated the GSM weights by first running CD with one iteration of the Gibbs sampler until progress went slow. We then ran 15 iteration CD, more resembling ML, to further tune the parameters until convergence (by optical inspection). We will call this strategy "CD-ML" after Carreira-Perpiñán and Hinton [2005], who suggested it. We note that the 15-step CD tuning only slightly improved the results. Figure 5.1(b) shows the learned potential; note that most of the weight is put on the smallest scale (Fig. 5.1(a)). The marginal derivative statistics of samples drawn from the model (Fig. 5.1(c)) match those of natural images quite well, but are not pointed enough. This suggests larger scales are required to better fit the data. The tails are barely wide enough; the large weight on the smallest scale suggests to further expand the range of scales in this direction as well.

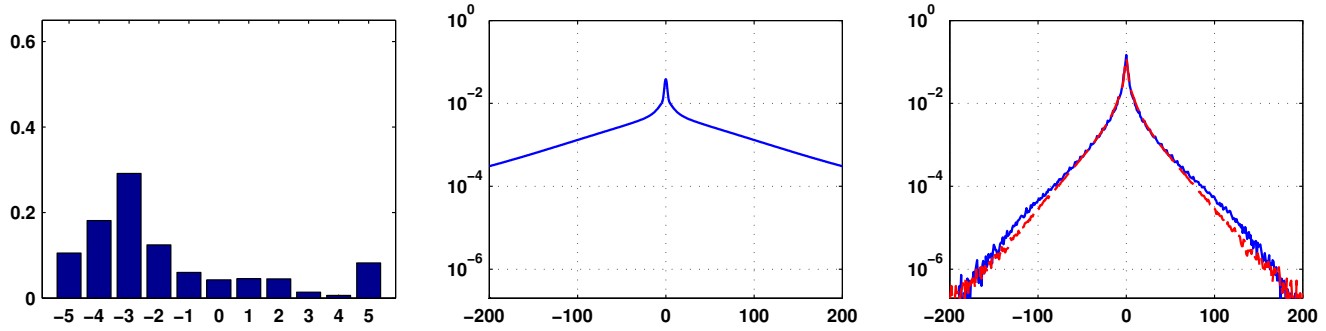
Note that before computing the marginal statistics, we always trim each sample to 28×28 pixels, ignoring the underconstrained boundary pixels which are overlapped by fewer cliques than pixels in the interior. The influence of these boundary pixels seems to be negligible in pairwise MRFs; they are however a problem in the Fields of Experts (Section 5.3).

Motivated by the findings of this first experiment, we added both smaller and larger scales and set

$$s = (e^{-5}, e^{-4}, e^{-3}, e^{-2}, e^{-1}, e^{0}, e^{1}, e^{2}, e^{3}, e^{4}, e^{5}). \tag{5.20}$$

We trained the MRF as in the previous experiment, and the results are shown in Figure 5.2. The weight distribution of scales is now more spread out. It can clearly be seen that the marginal statistics are a better match, which is also expressed in terms of improved KL-divergence (KLD).

Note that the learned GSM potential is significantly heavier-tailed than the marginal derivative statistics and can be considered *optimal* for generative pairwise MRF image models (using first derivatives)



(a) GSM weights (b) GSM potential (c) Marginals, KLD = 0.0107

Figure 5.2: Learned pairwise MRF using CD-ML and scales from e^{-5} to e^{5}. *(a)* GSM weight distribution of log-scales. *(b)* Semi-log plot of GSM potential. *(c)* Marginal semi-log derivative histogram for natural images (solid blue) and samples drawn from the pairwise MRF (dashed red).

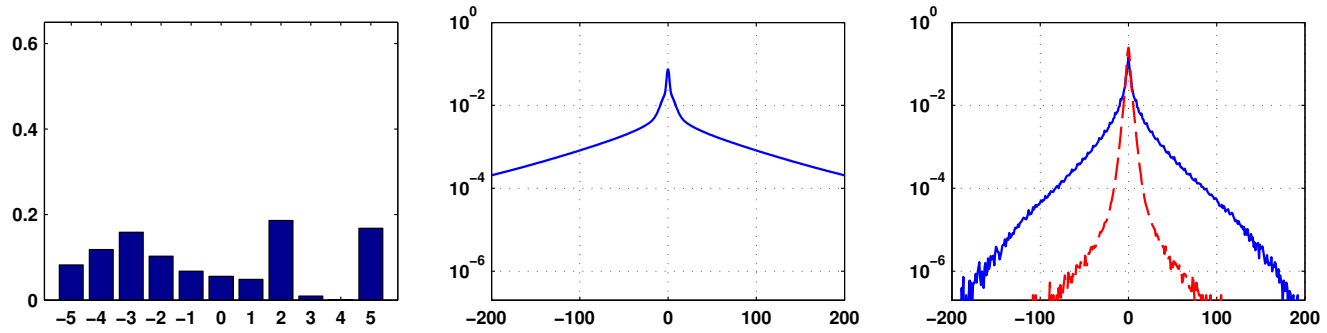
due to the maximum entropy model interpretation of pairwise MRFs (cf. Zhu and Mumford [1997]). To the best of our knowledge, this is the first time that such an optimal pairwise potential has been reported. We will also show in Section 5.4.2 that fitting GSM potentials directly to the empirical derivative marginals, similar to Scharr et al. [2003]; Weiss and Freeman [2007], does not capture the marginal derivative statistics of natural images correctly.

Having successfully learned a GSM potential with CD-ML and shown which GSM scales are suitable, we tried score matching under the same circumstances. We found that SM fails to produce good results, even when initialized with the optimal parameters learned via CD-ML. The experimental results for SM are shown in Figure 5.3. We observed that SM quickly increases the weight of the largest scale at the beginning of the learning progress, resulting in a significant drop of the cost function. After convergence for this weight is slow, the other weights slowly change as well. Their contribution to the cost function is presumably rather small. Note that the ratio of the weights for the smaller scales has not changed much from their initialization, they just all "lost" to scales e^{3} and e^{5}. It is also noteworthy that the variance of the two largest weights during learning is high, but small for all other weights. We also tried SM using smaller scales but essentially observed similar behavior. When using a larger scale, such as e^{10}, and natural images with integer intensity values (e.g. from P.O. Hoyer's ImageICA[3] package), SM puts almost all weight (≈ 0.99) on the largest scale.

Efficiency. We observed SM to be actually slower than 1-step CD in our experiments, due to our efficient Gibbs sampler and the comparatively more complex SM objective function. We deem this noteworthy since SM was originally proposed as a computationally inexpensive estimator that avoids costly MCMC-based sampling techniques. While this may be true in general, 1-step CD can actually be faster when using efficient MCMC-samplers.

One pass over the 1000 image patches in our training set (50 groups of 20 images) took on average 18 seconds with 1-step CD, 145 seconds with 15-step CD, and 41 seconds with SM; these numbers are from the previous two experiments using 15 scales, both run on the same computer with comparable (simple) MATLAB implementations. Additionally, we found SM to be quite sensitive to the learning rate, whereas CD was rather robust to it. Hence, we were forced to use a small learning rate for SM, effectively requiring many more iterations than CD to converge. For CD-ML, we could use a rather large step-size with 1-step CD and then relatively few 15-step CD iterations to tune the weights. Also, we re-iterate that 15-step CD is not crucial to learn a pairwise MRF with approximately correct derivative marginals.

[3] Available at `http://www.cs.helsinki.fi/u/phoyer/imageica.tar.gz`.



(a) GSM weights (b) GSM potential (c) Marginals, KLD = 0.9508

Figure 5.3: Learned pairwise MRF using SM and scales from e^{-5} to e^{5}, initialized with the weights shown in Fig. 5.2(a). *(a)* GSM weight distribution of log-scales. *(b)* Semi-log plot of GSM potential. *(c)* Marginal semi-log derivative histogram for natural images (solid blue) and samples drawn from the pairwise MRF (dashed red).

Likelihood bounds. We also computed the likelihood bounds devised by Weiss and Freeman [2007], but found them to be uninformative to compare our learned models – which may be due to our broad selection of exponentially-spaced scales. Note that we generalized the likelihood bounds (Appendix B) to fit our model definition from Chapter 3.

We compute the average log-likelihood

$$\bar{\ell}(\boldsymbol{\Theta}) = \frac{1}{T}\log\prod_{t=1}^{T} p(\mathbf{x}^{(t)};\boldsymbol{\Theta}) = \frac{1}{T}\sum_{t=1}^{T} -\log Z(\boldsymbol{\Theta}) - E(\mathbf{x}^{(t)};\boldsymbol{\Theta}) = -\log Z(\boldsymbol{\Theta}) - \langle E(\mathbf{x};\boldsymbol{\Theta})\rangle_{\mathbf{X}}, \tag{5.21}$$

where $\mathbf{X} = \{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(T)}\}$ is a test set of 1000 30×30 image patches. Table 5.1 shows the relevant values for the three pairwise MRFs learned so far.

	$\log Z(\boldsymbol{\Theta})$		$\langle E(\mathbf{x};\boldsymbol{\Theta})\rangle_{\mathbf{X}}$	$\bar{\ell}(\boldsymbol{\Theta})$	
MRF model	lower	upper		lower	upper
From Figure 5.1 (CD-ML, 7 scales)	−5712	−2022	7371	−5349	−1659
From Figure 5.2 (CD-ML, 11 scales)	−9606	−600	7462	−6862	2144
From Figure 5.3 (SM, 11 scales)	−10024	513	6812	−7325	3212

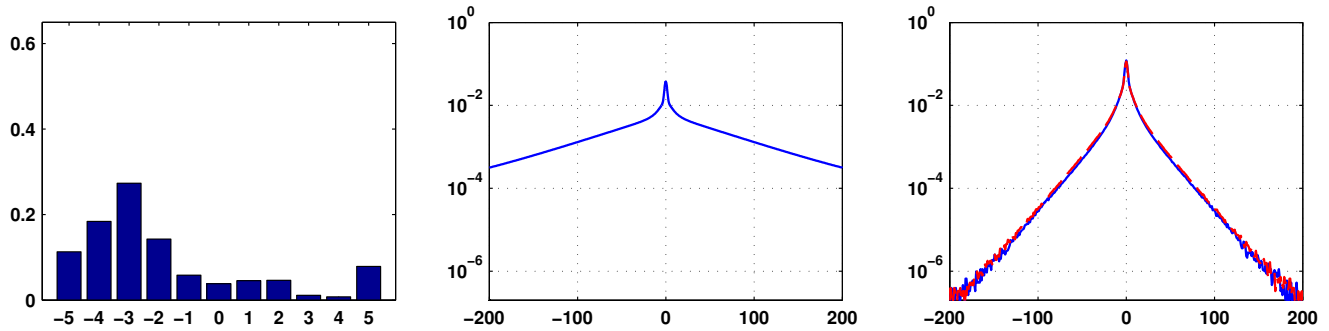
Table 5.1: Bounds on log partition function and average log-likelihood for learned pairwise MRFs.

5.2.2 Synthetic images

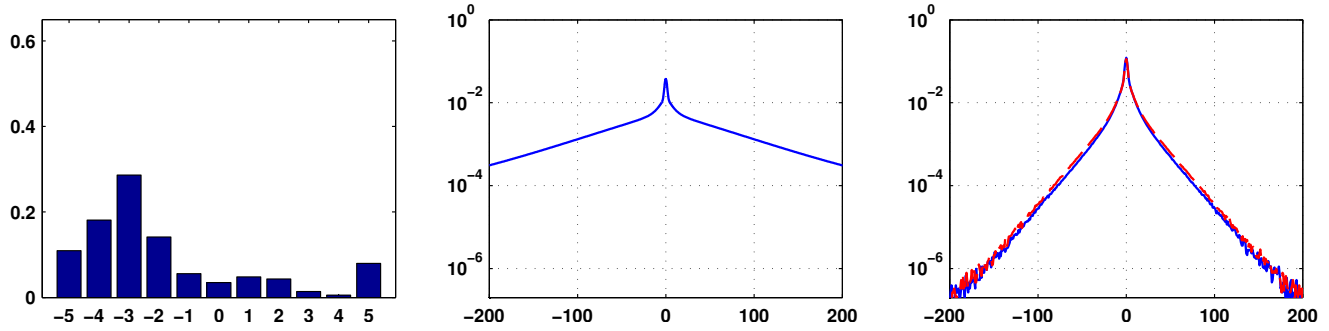
We repeated the previous experiment with CD-ML and SM, but instead of learning from natural images we used samples drawn from the MRF ("synthetic images", Figure 5.5(b)) using the optimal potential learned via CD-ML (Figure 5.2(b)). The advantage is twofold: First, it allows us to obtain "perfect" training examples, free from any noise and other structure irrelevant to our model; second, we know the ground truth MRF that has been used to generate the samples.

Interestingly, SM and CD-ML perform equally well when learning from these synthetic training examples (Figure 5.4). Also, both estimation methods show a local optimum when learning is initialized with the ground truth weights. We observed that convergence for SM is significantly slower and that increasing the learning rate must be done with caution, since we also managed to end up with bad results when setting the learning rate too high.

This raises the question what the difference between natural image patches and these synthetic samples is, as far as the SM estimator is concerned. Both training sets are almost perfectly equal in terms of

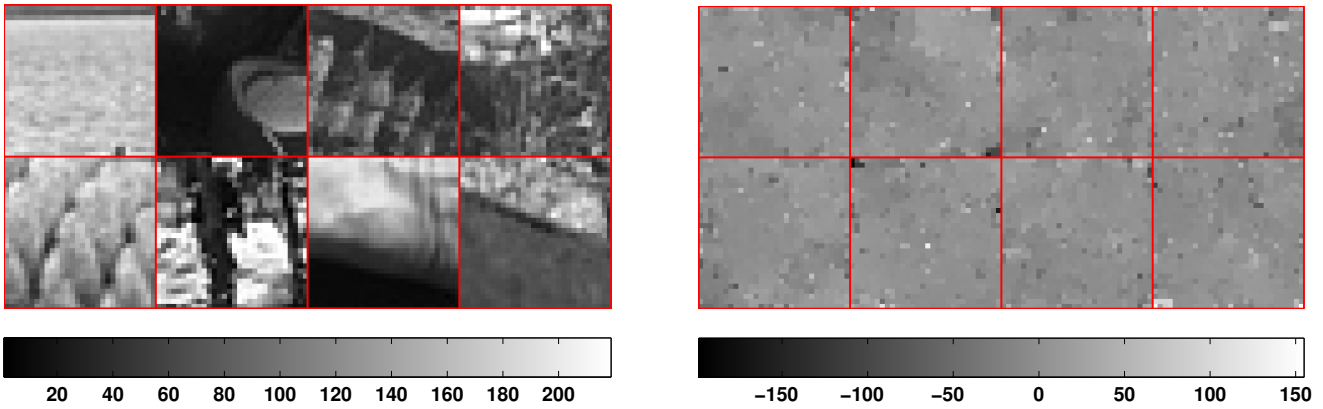


(a) CD-ML / GSM weights (b) CD-ML / GSM potential (c) CD-ML / Marginals, KLD = 0.0024



(d) SM / GSM weights (e) SM / GSM potential (f) SM / Marginals, KLD = 0.0037

Figure 5.4: Learned pairwise MRFs from synthetic images using CD-ML *(a–c)* and SM *(d–f)*. *(a, d)* GSM weight distribution of log-scales. *(b, e)* Semi-log plot of GSM potential. *(c, f)* Marginal semi-log derivative histogram for synthetic images (solid blue) and samples drawn from the learned pairwise MRF (dashed red).

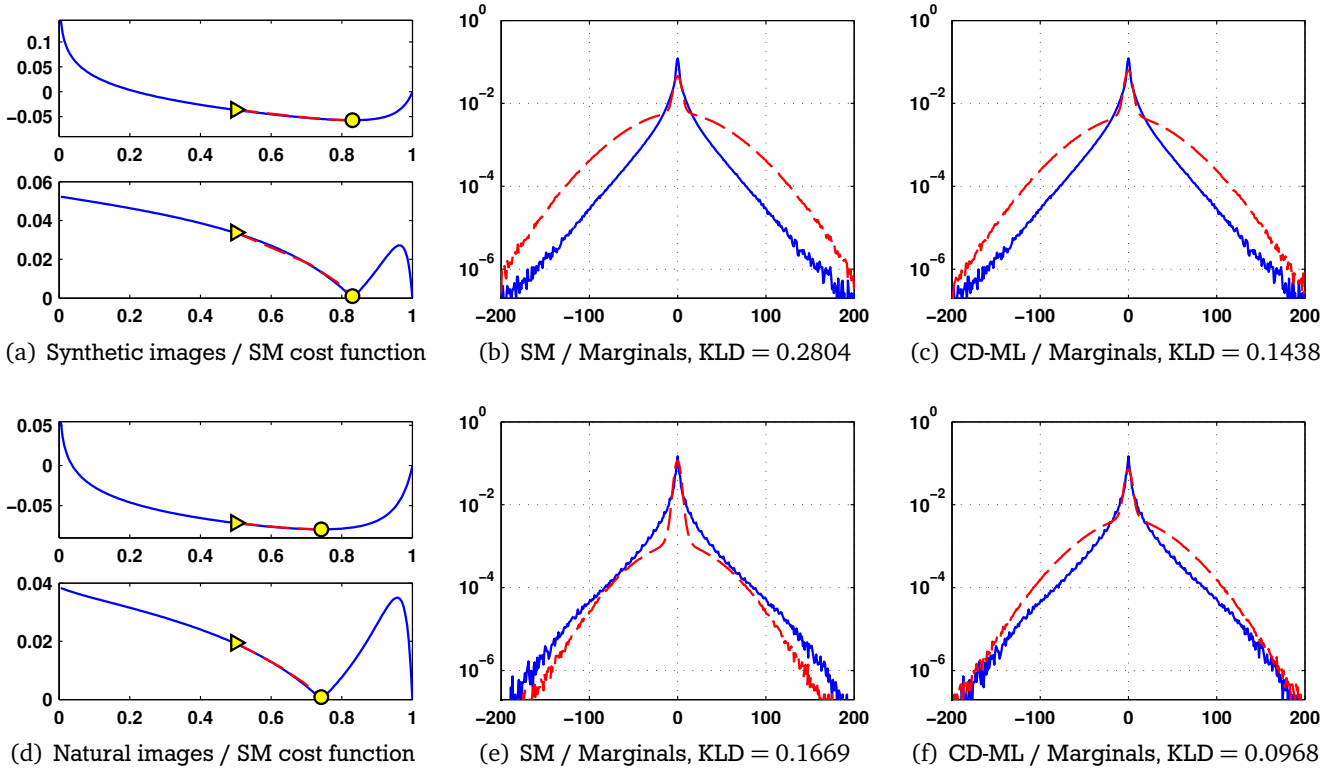


(a) Natural image patches

(b) Samples drawn from pairwise MRF, using the potential shown in Figure 5.2(b). The mean value from each sample has been subtracted for better visualization.

Figure 5.5: Subset of training data used in our experiments. The red lines separate individual 30×30 pixel training examples.

derivative marginals, which is the only feature that pairwise MRFs model. Relating to our univariate experiments in Chapter 4, we could speculate here that SM is working well with perfect training examples, but showing problems otherwise.



(a) Synthetic images / SM cost function (b) SM / Marginals, KLD = 0.2804 (c) CD-ML / Marginals, KLD = 0.1438

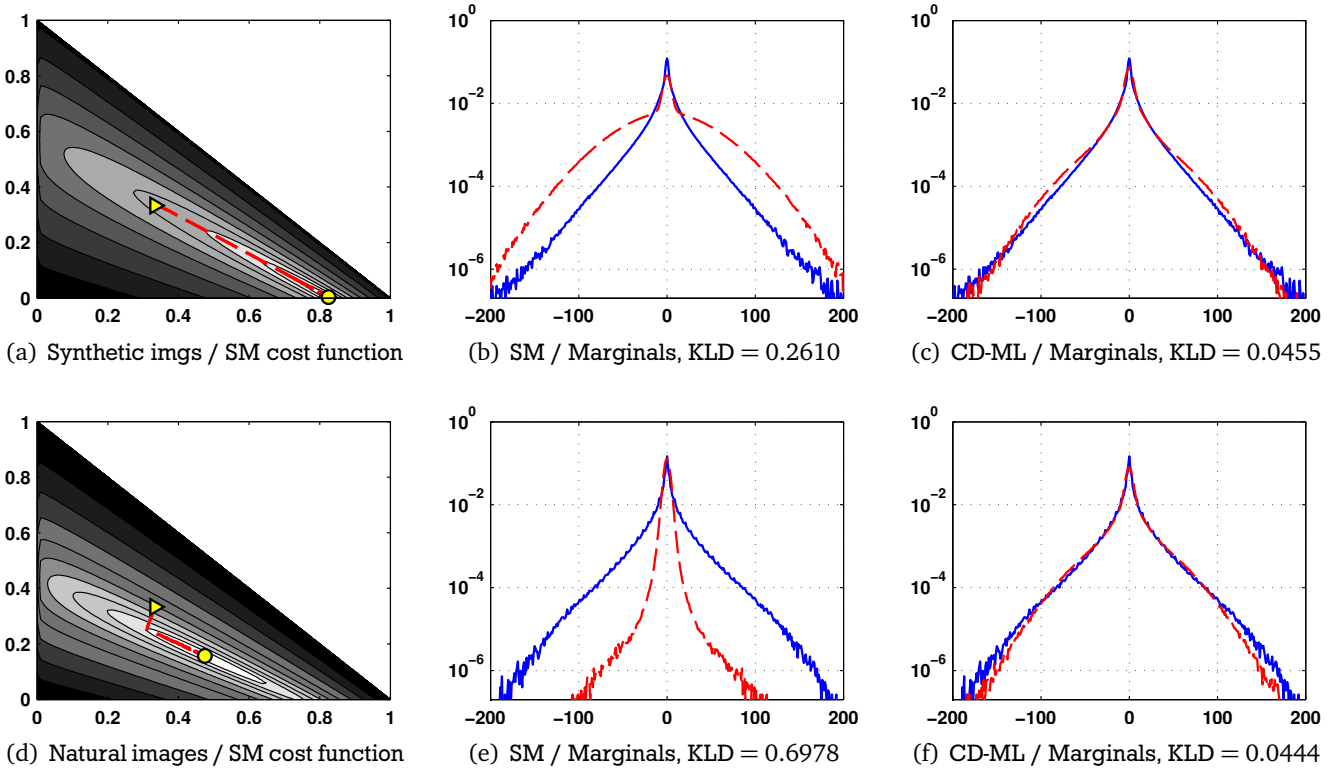
(d) Natural images / SM cost function (e) SM / Marginals, KLD = 0.1669 (f) CD-ML / Marginals, KLD = 0.0968

Figure 5.6: Experiments with 2 scales for synthetic images *(a–c)* and natural images *(d–f)*. *(a, d)* Top: weight of first scale vs. SM cost function; bottom: weight of first scale vs. norm of gradient w.r.t. GSM weights; super-imposed weight progress during SM learning (circle denotes final result). *(b, c, e, f)* Marginal semi-log derivative histogram for images (solid blue) and samples drawn from the learned pairwise MRF (dashed red). The GSM base variance was always fit to the training data prior to learning.

5.2.3 Visualization in a simplified setting

In order to better understand the results obtained by SM, we computed and visualized the SM cost function when using GSMs with 2 and 3 scales only. Using scales $s = (e^{-3}, e^{3})$ and $s = (e^{-3}, e^{0}, e^{3})$ should suffice to obtain rather good results, as can be assumed from our first experiment (Figure 5.1). For both, natural image patches and synthetic images from the MRF (Figure 5.5; having virtually equal derivative marginals), we exhaustively computed the SM cost function with a weight step width of 0.005 in case of 2 scales, and 0.01 in case of 3 scales. We also learned the weights using SGD as in the other experiments, and superimposed the weight progress on top of the cost function; we carried out CD-ML learning for comparison as well.

The results can be seen in Figures 5.6 and 5.7. We see that SM does not get stuck in a local minimum and indeed converges to the global minimum. In case of 3 scales, we observe a ridge in the cost function where values are very similar, and hence requires to choose a very small learning rate for gradient-based methods. Using more scales does greatly improve the results obtained by CD-ML, but not for SM – the results are similar or even worse. This is obviously a very simplified setting, but the different results of SM when learning from natural image patches and samples from the MRF remain. We could speculate that SM shows weak performance when the training data is not actually from the model distribution – which would make SM very fragile to choosing the correct model for the data, and unsuitable for many practical applications.



(a) Synthetic imgs / SM cost function (b) SM / Marginals, KLD = 0.2610 (c) CD-ML / Marginals, KLD = 0.0455

(d) Natural images / SM cost function (e) SM / Marginals, KLD = 0.6978 (f) CD-ML / Marginals, KLD = 0.0444

Figure 5.7: Experiments with 3 scales for synthetic images *(a–c)* and natural images *(d–f)*. *(a, d)* Weight of first and second scale vs. SM cost function (log scale, darker is higher), super-imposed with weight progress during SM learning (circle denotes final result). *(b, c, e, f)* Marginal semi-log derivative histogram for images (solid blue) and samples drawn from the learned pairwise MRF (dashed red). The GSM base variance was always fit to the training data prior to learning.

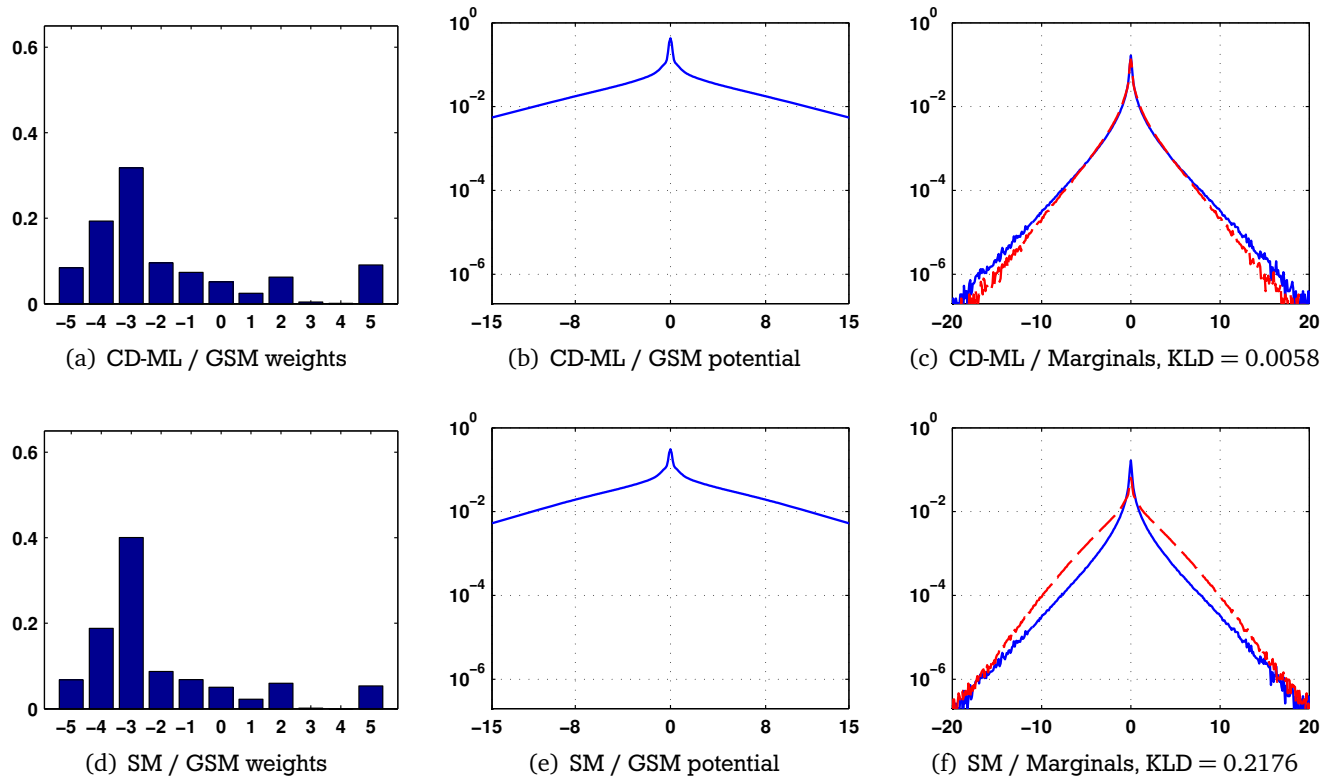
5.2.4 Whitened images

In a final set of pairwise MRF experiments, we considered whitened images to see if SM is able to perform better. We whitened the image patches using a zero-phase whitening filter (cf. Köster et al. [2009]), which comprises "normal" whitening in order to de-correlate the random variables; additionally the whitened data is rotated back to the original coordinate system to keep its spatial relation. The base variance for the potential has to be chosen entirely different since all pixel variables now have unit variance ($\sigma^2 \approx 2$, still set to the empirical variance of the image derivatives of the training set). The selection of scales remained unchanged.

SM behaves rather different in comparison to learning from "normal" (gamma-compressed) images. Whereas the sample statistics from Figure 5.3 show that SM overestimated the mode and underestimated the tails, the situation is reversed for whitened images; the results are shown in Figure 5.8. Our remarks to SM's behavior during learning from samples also apply to this experiment. Additionally, the learning rate for SM had to be decreased by a factor of about 100 to make it work; we did not need to adjust the learning rate for CD-ML though.

5.3 Fields of Experts

Moving towards more powerful models, we learned Fields of Experts with square filter sizes of 3×3 and 5×5 pixels, constrained to have mean zero. The details of the learning procedure are similar to the pairwise case; we however set the base variance $\sigma^2 = 500$ for each GSM expert instead of fitting it to



(a) CD-ML / GSM weights (b) CD-ML / GSM potential (c) CD-ML / Marginals, KLD = 0.0058

(d) SM / GSM weights (e) SM / GSM potential (f) SM / Marginals, KLD = 0.2176

Figure 5.8: Learned pairwise MRFs learned from whitened images using CD-ML *(a–c)* and SM *(d–f)*. *(a, d)* GSM weight distribution of log-scales. *(b, e)* Semi-log plot of GSM potential. *(c, f)* Marginal semi-log derivative histogram for whitened images (solid blue) and samples drawn from the pairwise MRF (dashed red). SM learning was initialized with the weights shown in *(a)*.

the empirical derivative marginals of the training set. Since we are also learning the filters of the MRF, we extended the range of scales $s = (e^{-9}, e^{-7}, e^{-5}, e^{-4}, e^{-3}, e^{-2}, e^{-1}, e^{0}, e^{1}, e^{2}, e^{3}, e^{4}, e^{5}, e^{7}, e^{9})$ to allow for greater flexibility of the GSM experts. We only used 1-step CD and did not employ CD-ML due to computational demands.

5.3.1 Natural images

We were unable to learn an FoE with SM due to numerical problems, no matter how small we chose the learning rate. At some point the filter coefficients took on extreme values and the GSM weights oscillated heavily. This behavior was somewhat delayed when using smaller scales, we however showed that a broad range of scales is already necessary for the special case of pairwise MRFs. Initializing SM learning with a solution previously obtained by CD did not help either.

Hence, we only learned FoEs using CD with 8 3×3 filters and 24 5×5 filters; Figures 5.9 and 5.10 show the FoEs and their generative properties – which we analyze by looking at the marginal distributions of filter responses (each model w.r.t. its own learned bank of filters). We notice that the learned MRFs heavily overfit on the boundary pixels, which are less constrained than pixels in the image interior because they are overlapped by fewer cliques in the MRF. As Norouzi et al. [2009], we also observe that this leads to extreme values at the boundary pixels when sampling the model. Hence, we find that the filter marginals of our learned FoEs fit those of natural images very well when including the boundary pixels of samples; they are however a poor match when those pixels are left out.

This comes down to the question what it is that we want to model. Natural images do not have a distinct boundary, we therefore think a model of natural images should not rely on extreme values at the

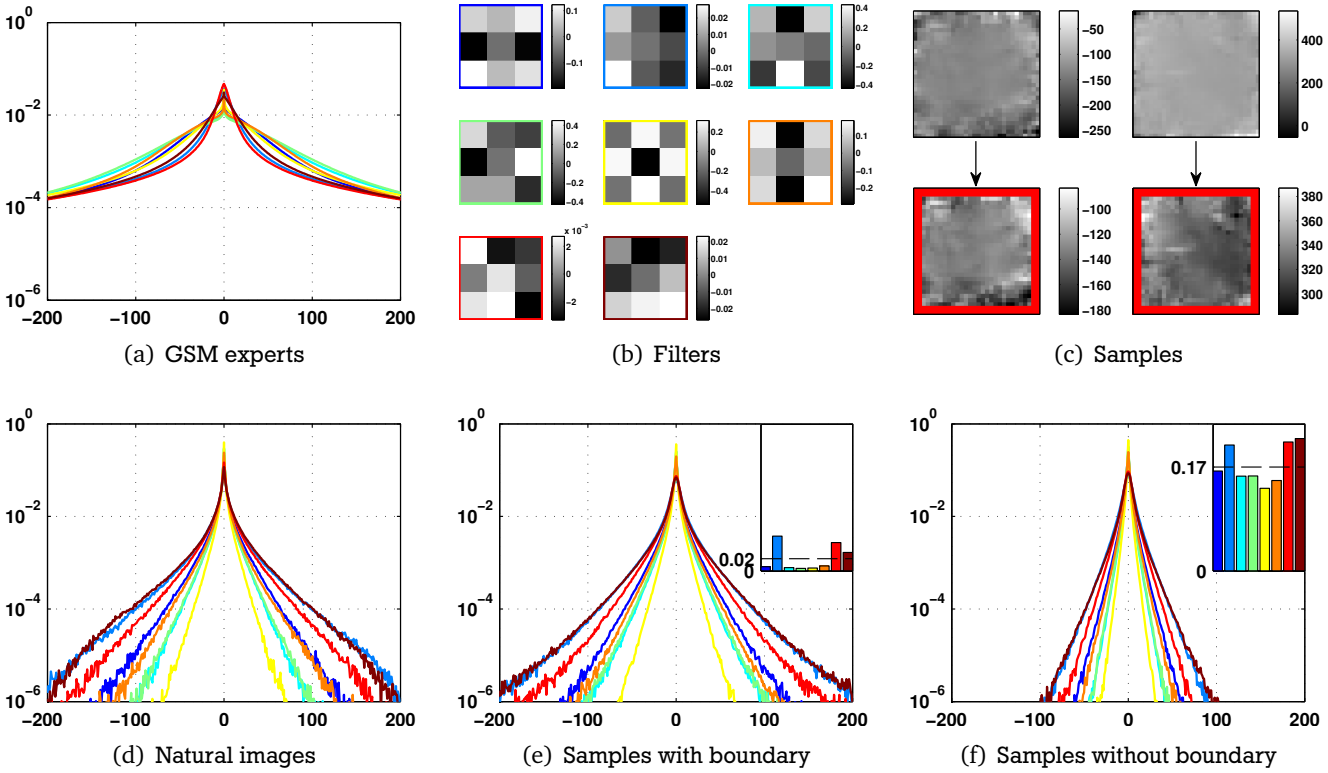


Figure 5.9: Learned 3×3 FoE using CD. *(a, b)* Learned experts and filters. *(c)* Example of MRF samples with and without boundary pixels. *(d–f)* Filter marginals (filters are normalized for ease of display). The bar charts show the marginal KL-divergence of each feature; same color across sub-figures denotes same expert/filter.

boundary pixels to match the statistics of natural images. We will address this problem and a possible solution in Section 5.4.

5.3.2 Synthetic images

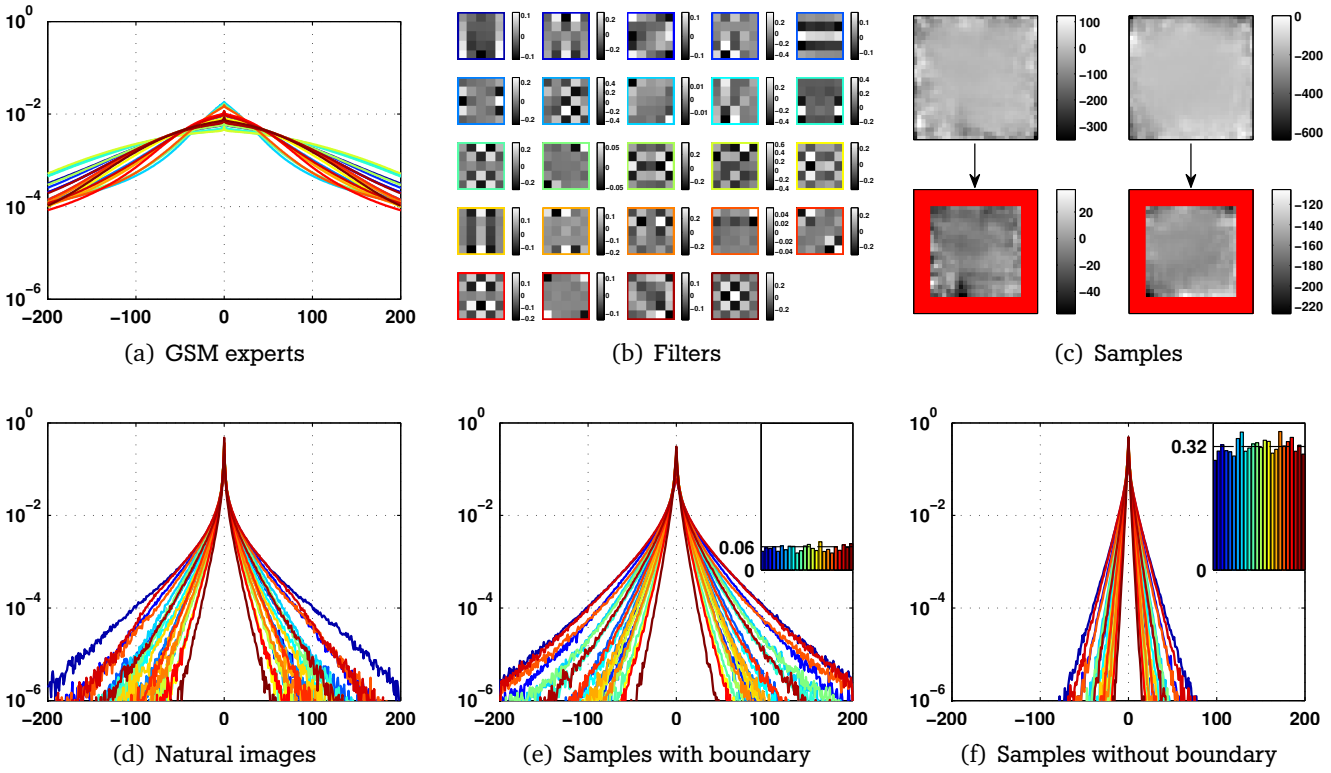
We generally think it is important to get SM working well in the simpler pairwise MRFs first before investigating its application to the more complicated FoEs. We however did some simple experiments and found that at least learning was possible from synthetic images; SM was however not able to recover the MRF (that produced the samples) very well. When initialized with the ground truth MRF, however, SM showed a local optimum not far from the ground truth weights. SM worked much better when we tried learning from samples drawn from an MRF with smooth GSM experts – which supports our hypothesis that SM does not work well with heavy-tailed distributions.

5.3.3 Whitened images

As in the pairwise MRF, we also did experiments with learning from whitened images. We were especially interested in understanding the results of Köster et al. [2009], who were the first to learn an FoE with SM. As Köster et al. [2009][4] , we used the fixed expert function

$$\phi(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)}) = \cosh(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)})^{-1} \tag{5.22}$$

[4] This was revealed to us upon request, it is not mentioned in the paper.



(a) GSM experts (b) Filters (c) Samples

(d) Natural images (e) Samples with boundary (f) Samples without boundary

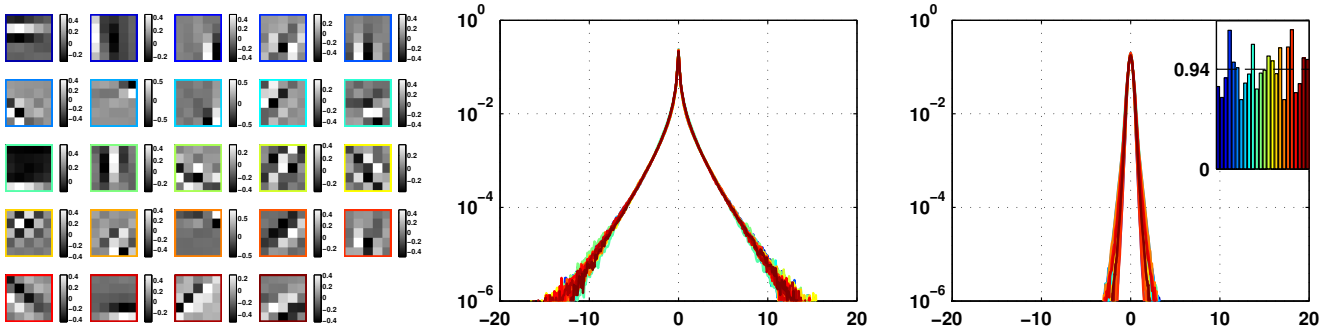
Figure 5.10: Learned 5×5 FoE using CD. *(a, b)* Learned experts and filters. *(c)* Example of MRF samples with and without boundary pixels. *(d–f)* Filter marginals (filters are normalized for ease of display). The bar charts show the marginal KL-divergence of each feature; same color across sub-figures denotes same expert/filter.

for all filters and cliques of the FoE, in contrast however only learned 24 filters of size 5×5 instead of their 144 12×12 filters. The training image patches were whitened as described for the pairwise MRF.

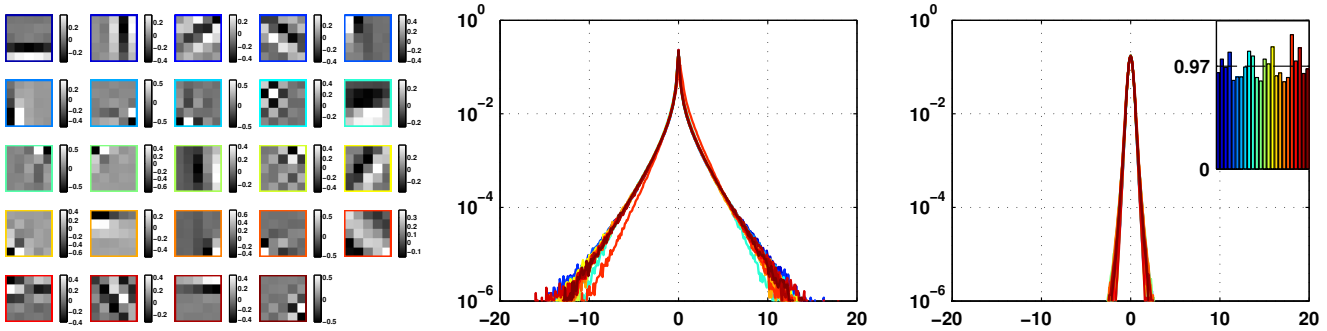
We learned the filters of the 5×5 FoE using both SM and CD, the latter using a GSM approximation of the expert from Eq. (5.22) in order to use our efficient Gibbs sampler. We also constrained all filters to unit-norm. Interestingly, we get qualitatively similar results to those of Köster et al. [2009], as far as it possible to tell, using both learning approaches. The results are shown in Figure 5.11. The filter marginals of samples from both learned FoEs are very similar, although they do not fit those of natural images, even when including the sample boundary pixels. It is interesting that SM gives similar results to CD under these circumstances, which again supports our theory that SM has problems with very heavy-tailed distributions, since the expert from Eq. (5.22) is not especially heavy-tailed and does not exhibit a sharp peak like the learned GSM experts (via CD) so far (which lead to good generative properties).

Like Köster et al. [2009], we also observed (not shown) that most of the filters went to zero when the filter norm was not constrained to 1, although a few filters became quite large. The question is why the norm has be restricted since there seems to be no good reason for doing it. When using an FoE with GSM experts, we do not encounter this problem. In fact, we repeated the same experiment (using CD) while also learning the weights of the GSM experts and found the marginal statistics to be much better (although still overfitting on the boundary pixels); the results are shown in Figure 5.12. Hence, in terms of image modeling, using non-heavy-tailed potentials such as from Eq. (5.22) seems unsuitable.

It is also noteworthy that learning converges much faster on whitened data.

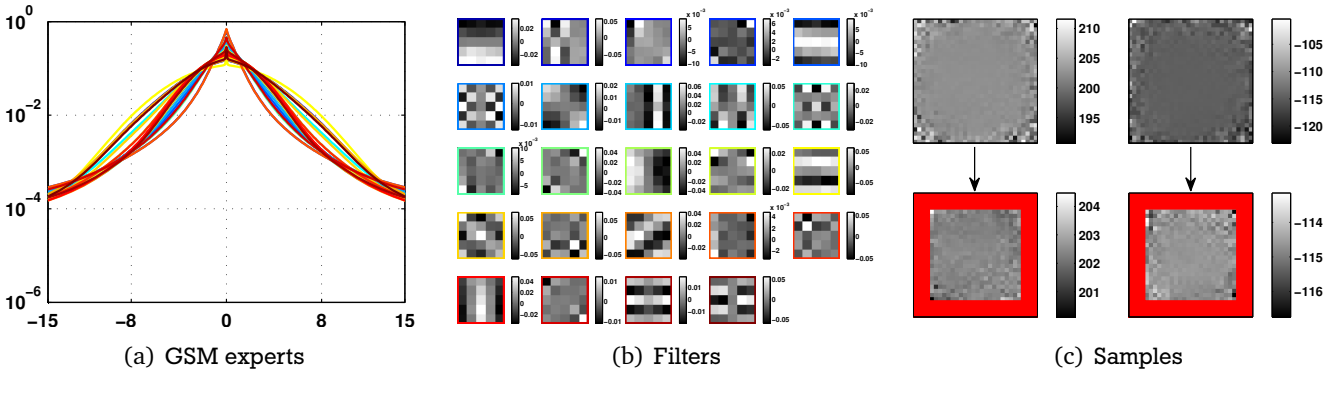


(a) CD / Filters (b) CD / Whitened images (c) CD / Samples with boundary

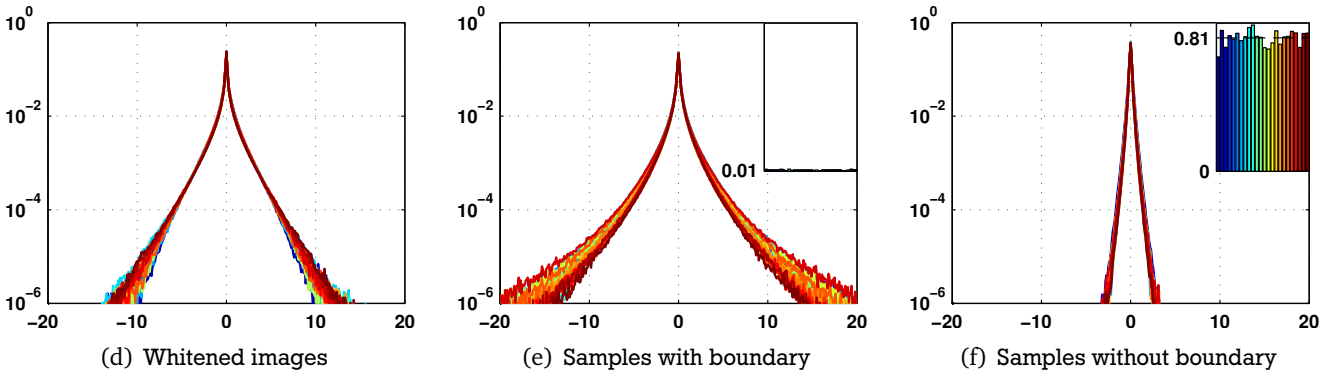


(d) SM / Filters (e) SM / Whitened images (f) SM / Samples with boundary

Figure 5.11: Learned 5×5 FoEs from whitened images with fixed experts and unit-norm filter constraint. *(a, d)* Learned filters, *(b, c, e, f)* filter marginals (filters are normalized for ease of display). The bar charts show the marginal KL-divergence of each feature; same color across horizontal sub-figures denotes same expert/filter.



(a) GSM experts (b) Filters (c) Samples



(d) Whitened images (e) Samples with boundary (f) Samples without boundary

Figure 5.12: Learned 5×5 FoE from whitened images using CD. *(a, b)* Learned experts and filters, *(c)* example of MRF samples with and without boundary pixels, *(d–f)* filter marginals (filters are normalized for ease of display). The bar charts show the marginal KL-divergence of each feature; same color across sub-figures denotes same expert/filter.

5.4 Using Boundary Handling

We demonstrated in the previous section that it is possible to learn FoEs whose filter marginals fit those of natural images very well, but unfortunately rely on including the boundary pixels of samples that take on extreme values. We found this to be no problem in the pairwise MRF, where the difference in the number overlapping cliques between interior and boundary pixels is small, and the filters are fixed. The problem of overfitting on the boundary pixels increases as the filter size grows, since the gap between interior and boundary pixels in terms of overlapping cliques widens.

We could use much larger training image patches to reduce the influence of the boundary pixels in the learning process. Another approach [Norouzi et al., 2009], which we pursue here, is to keep the less constrained pixels at the boundary, $\mathbf{x}^b$, fixed and conditionally sample the interior $\mathbf{x}^i$ according to $p(\mathbf{x}^i|\mathbf{x}^b, \mathbf{z}; \boldsymbol{\Theta})$. Since $p(\mathbf{x}|\mathbf{z}; \boldsymbol{\Theta})$ is Gaussian, the required conditional distribution is easy to derive, as shown in Section 3.1.1. We found that this conditional learning procedure reduces overfitting on the boundary pixels, yet is more efficient than simply training on larger image patches to achieve the same goal.

From Figures 5.9(c), 5.10(c), and 5.12(c), we can also see that the values of the boundary pixels influence the interior of the samples. Hence, we ignore an even larger boundary when computing the marginals in the following.

5.4.1 Pairwise MRF and FoE for natural images

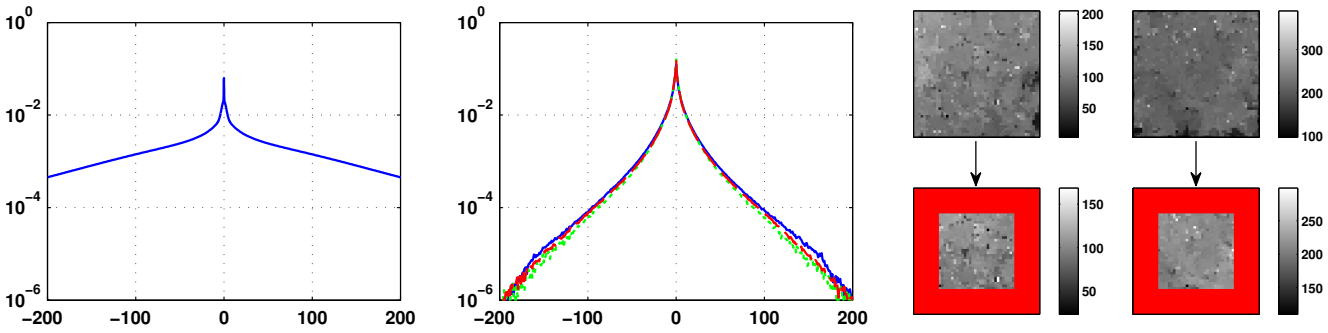
In contrast to the previous experiments in this chapter, we use a larger training set of 5000 grayscale 50×50 image patches, randomly cropped from all training images of the Berkeley image segmentation dataset [Martin et al., 2001]; we now correctly use the full range of graylevels. We employ conditional sampling during CD learning as described above, and use the extended range of scales $s = \exp(0, \pm 1, \pm 2, \pm 3, \pm 4, \pm 5, \pm 7, \pm 9)$, even for the pairwise MRF.

From now on, our natural image validation set consists of 3800 non-overlapping 30×30 patches, randomly cropped from grayscale versions of the test images of the Berkeley image segmentation dataset [Martin et al., 2001]. We randomly sample 3800 images of size 50×50 to evaluate the generative properties of the MRF models, but only use the 30×30 pixels in the middle to compute the sample statistics in order to reduce the influence of the boundary pixels. We employ conditional sampling to avoid boundary artifacts, where image boundaries from a separate set of 3800 images patches are used (cropped from the training images of Martin et al. [2001]). During sampling, the fixed boundaries are $m-1$ pixels wide/high, where m is the maximum extent of the largest clique – which causes every interior pixel to be overlapped by the same amount of cliques. Instead of using a fixed amount of iterations, we assess sampler convergence by estimating the potential scale reduction as described in Section 3.1.2, however using at least 21 but no more than 501 iterations. To draw a single sample from the model distribution, we set up three chains with over-dispersed starting points: the interior of the boundary image, a smooth median-filtered version, and a noisy version with Gaussian noise ($\sigma = 15$) added.

We again trained pairwise MRFs using CD-ML and SM[5], with fixed horizontal and vertical derivative filters and a single GSM potential, and an FoE using 1-step CD with 3×3 cliques and 8 GSM experts including filters. We were unable to learn an FoE with 5×5 filters that improves on the learned 3×3 FoE in terms of generative properties, even when using a different basis for the filters.

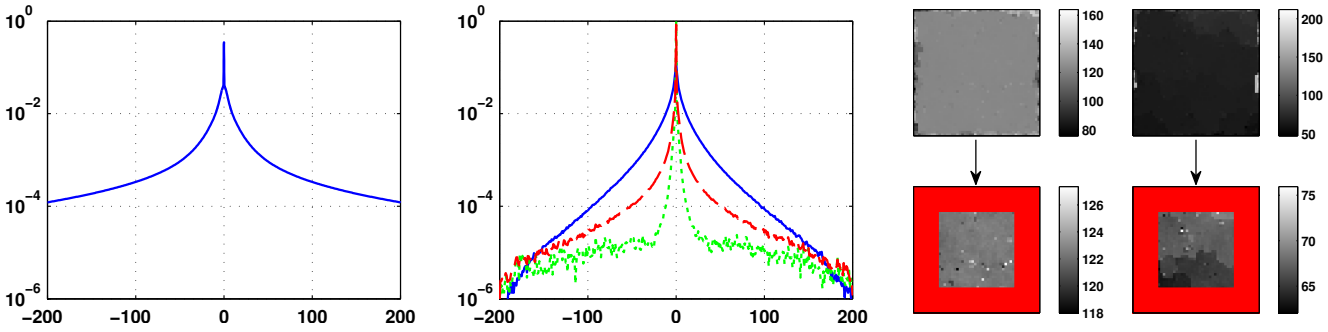
Figure 5.13(a) shows the learned pairwise potential via CD-ML, which in comparison to the previously trained potential (cf. Fig. 5.2(b)) exhibits an even stronger peak due to the extended range of scales. The important thing to note is that the influence of the sample boundary pixels on the derivative marginals is negligible, because pairwise MRF models do not suffer as much from the fact that boundary pixels

[5] We ignored the underconstrained boundary pixels in the SM cost function (and therefore the parameter gradient) for consistency with the other results here.



(a) GSM potential (b) Derivative marginals (c) Samples

Figure 5.13: Learned pairwise MRF using CD-ML with conditional sampling. *(a)* Learned GSM potential. *(b)* Derivative marginals of natural images (solid blue), samples with boundary (dashed red, KLD $= 0.0079$), and samples without boundary (dotted green, KLD $= 0.0106$). *(c)* Example of MRF samples with and without 10 boundary pixels.



(a) GSM potential (b) Derivative marginals (c) Samples

Figure 5.14: Learned pairwise MRF using SM with boundary handling. *(a)* Learned GSM potential. *(b)* Derivative marginals of natural images (solid blue), samples with boundary (dashed red, KLD $= 1.3135$), and samples without boundary (dotted green, KLD $= 2.6388$). *(c)* Example of MRF samples with and without 10 boundary pixels.

are constrained by fewer overlapping cliques. In comparison, the pairwise potential learned via SM (Fig. 5.14) exhibits a stronger peak and at the same time less heavy tails – which results in incorrect derivative marginals.

In case of the learned 3×3 FoE, we find *very broad experts with a small, narrow peak* (Fig. 5.15(a)), on close inspection even more heavy-tailed than the experts trained with "full" sampling (cf. Fig. 5.9(a)). Their almost δ-like shape differs from the experts used in the literature [Roth and Black, 2009; Weiss and Freeman, 2007]. Figure 5.15 shows that these learned experts significantly reduce the dependency on the sample boundary pixels compared to the other FoEs learned so far, even when ignoring a generous boundary of 10 pixels to compute the marginals. Further research is necessary, however, since the filter statistics are not perfectly captured yet.

5.4.2 Comparison with other MRFs

We converted other popular MRF priors to our model representation in order to use the efficient Gibbs sampler to analyze their generative properties via sampling. In particular, we fit GSM potentials to the target potentials by means of simple nonlinear optimization of the parameters α_{ij}. The GSMs are flexible enough to achieve good fits through a wide range of different shapes (KLD < 0.0002). We evaluated the

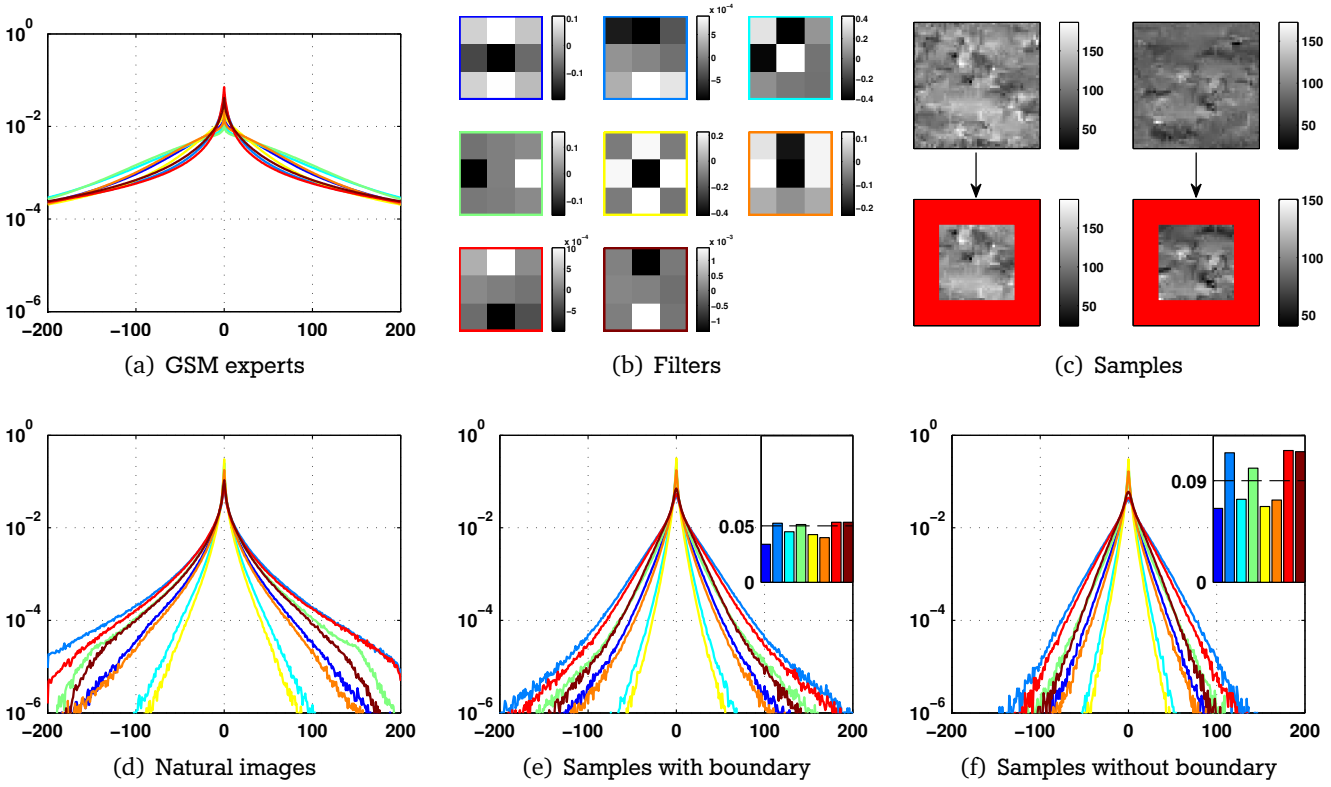


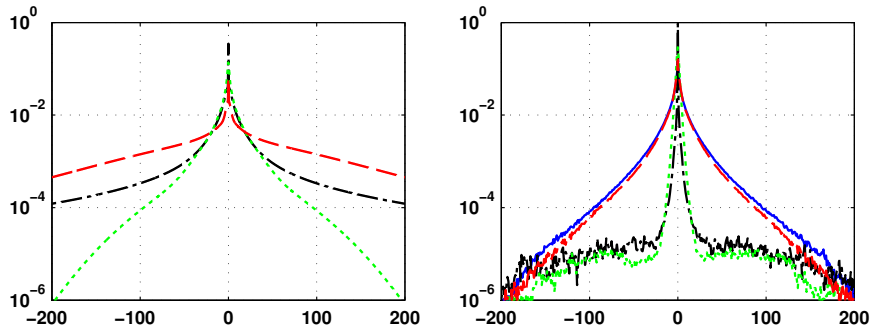
Figure 5.15: Learned 3×3 FoE using CD with conditional sampling. *(a, b)* Learned experts and filters. *(c)* Example of MRF samples with and without 10 boundary pixels. *(d–f)* Filter marginals (filters are normalized for ease of display). The bar charts show the marginal KL-divergence of each feature; same color across sub-figures denotes same expert/filter.

other MRFs like our learned models in the previous section, with the exception of using fewer samples for the FoE models.

The use of heavy-tailed potentials for pairwise MRFs with shapes similar to the empirical derivative statistics [Lan et al., 2006; Levin et al., 2009; Tappen et al., 2003] is directly motivated by the statistics of natural images. Potential functions have therefore been fit directly to the empirical derivative marginals [Scharr et al., 2003; Weiss and Freeman, 2007]. While some theoretical justification for fitting potentials to empirical marginals [Wainwright and Jordan, 2003] actually exists, there is no direct relation between potentials and marginals as in tree-structured graphical models [Wainwright and Jordan, 2003].

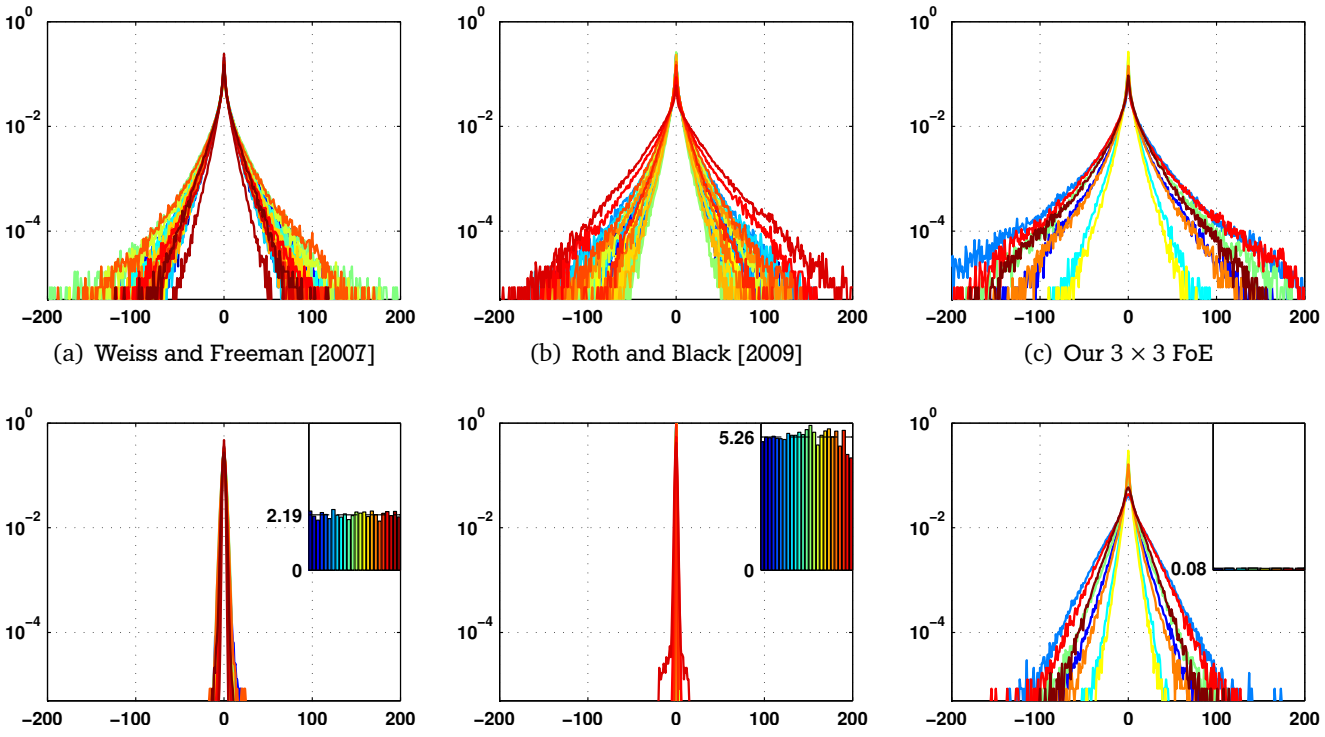
We fit a GSM potential directly to the empirical derivative marginals of our training set, similar to Scharr et al. [2003]; Weiss and Freeman [2007]; Figure 5.16 clearly demonstrates that the derivative statistics of natural images are not captured by a pairwise MRF with this potential. The model marginals are much too tightly peaked and the tails are too flat. We find that other, even less heavy-tailed, potentials like generalized Laplacians [Levin et al., 2009; Tappen et al., 2003] exhibit similar issues. Surprisingly, pairwise MRFs with similar potentials are widely used and have often shown good application performance in combination with MAP inference.

In case of the more powerful FoEs, we compared the generative properties of our 3×3 FoE with two other generatively-trained FoEs: the original FoE with Student-t experts [Roth and Black, 2009] and the GSM-based FoE model of Weiss and Freeman [2007]. Both, the original FoE model and the model of Weiss and Freeman [2007] do not capture the filter statistics (Fig. 5.17) of natural images. The model marginals are much too peaky for all filters, which manifests itself also in a high marginal KL-divergence. It is again surprising how widely used these models are, given their good application performance in the context of MAP estimation.



(a) Potentials (b) Derivative marginals

Figure 5.16: Pairwise MRF potentials and derivative marginals. *(a)* Fit of the marginals [Scharr et al., 2003] (dotted green), our learned GSM potential with CD-ML (dashed red), and our learned GSM potential with SM (dash-dotted black). *(b)* Derivative marginals of samples from MRFs with fit of the marginals potential (dotted green, KLD $= 1.45$), our learned GSM potential with CD-ML (dashed red, KLD $= 0.01$), and our learned GSM potential with SM (dash-dotted black, KLD $= 2.64$); statistics of natural images are shown in solid blue.



(a) Weiss and Freeman [2007] (b) Roth and Black [2009] (c) Our 3×3 FoE

(d) Weiss and Freeman [2007] (e) Roth and Black [2009] (f) Our 3×3 FoE

Figure 5.17: Filter statistics of natural images *(a–c)* and filter marginals of MRF models *(d–f)* (based on 300 samples without boundary, filters are normalized for ease of display). The bar charts show the marginal KL-divergence of each feature.

Figure 5.18 shows five subsequent samples (after reaching the equilibrium distribution) from all models compared here. Note how samples from pairwise MRFs generally allow for rather unrealistic single pixel discontinuities; the samples from the poor generative models in Figs. 5.18(b) and (c) are additionally too smooth. Samples from previous FoE models (Figs. 5.18(e) and (f)) are also too smooth, they are without large discontinuities.

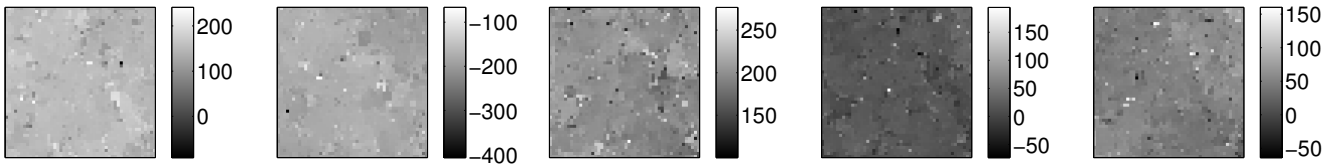
We can conclude that fitting experts to marginal statistics [Weiss and Freeman, 2007] is not appropriate, neither for pairwise MRFs nor FoEs. While we have not found optimal experts for FoEs, we can say

that the Student-t experts of Roth and Black [2009] are not heavy-tailed enough. Our learned models suggest that flexible potential functions and learning of all model parameters are key to achieving good generative properties.

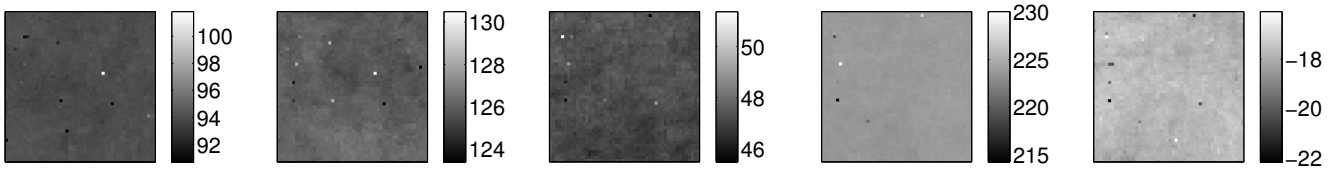
5.4.3 Further model analysis

We can gain further insight into our good generative models by inspecting additional statistical properties that the study of natural images has revealed. Already introduced in Section 2.2, we considered two characteristics of natural images here: first, the property that even random zero-mean filters of varying size exhibit heavy-tailed marginal statistics (Fig. 5.19(a)); and second, the scale invariance of derivative statistics [Srivastava et al., 2003] (Fig. 5.19(d)). We analyzed our models regarding these properties and also used the marginal KL-divergence as quantitative measure, going beyond Zhu and Mumford [1997] who only considered derivative statistics and did not perform quantitative measurements.

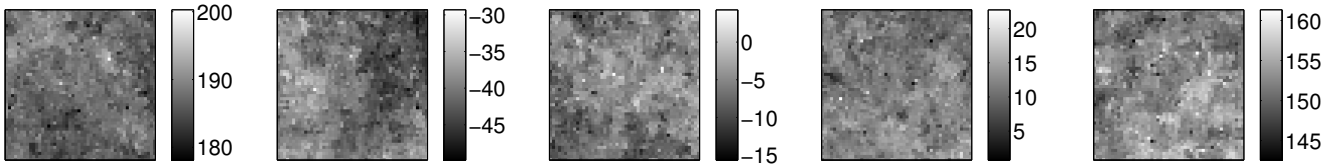
Our learned optimal pairwise MRF (via CD-ML) captures the statistics of small random 3×3 filters and derivatives at the smallest scale well (Figs. 5.19(b) and (e)), even slightly better than our learned high-order FoE. When it comes to larger random filters and large-scale derivatives, however, the model marginals tend toward being Gaussian. Figures 5.19(c) and (f) show the improved modeling power of our learned 3×3 FoE, which consistently captures the characteristics of natural images across a wider range of filter sizes and scales. This impression is also supported by visually comparing samples from both of our models (Fig. 5.20): samples from our pairwise MRF are locally uniform with large isolated discontinuities that look like "salt and pepper" noise; our high-order MRF produces more realistic samples that vary smoothly ("cloudy") with occasional edge-like discontinuities.



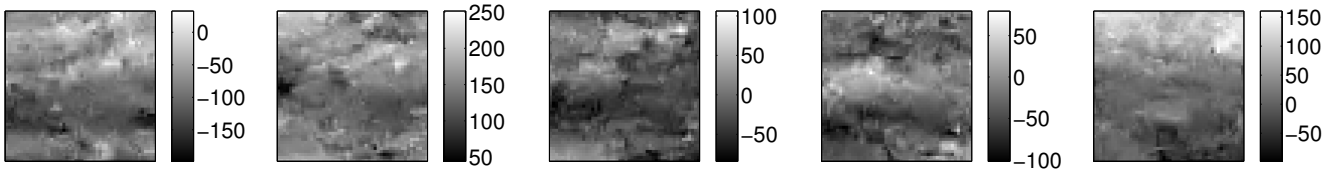
(a) Pairwise MRF (learned with CD-ML)



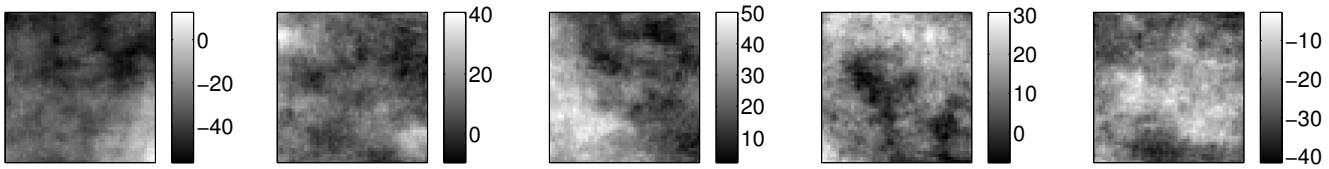
(b) Pairwise MRF (learned with SM)



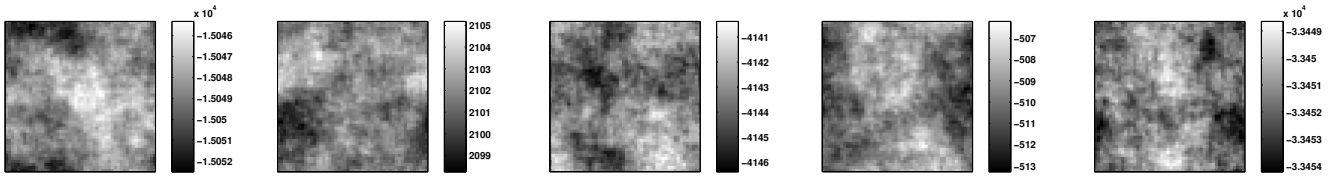
(c) Pairwise MRF (marginal fitting)



(d) 3×3 FoE (ours)

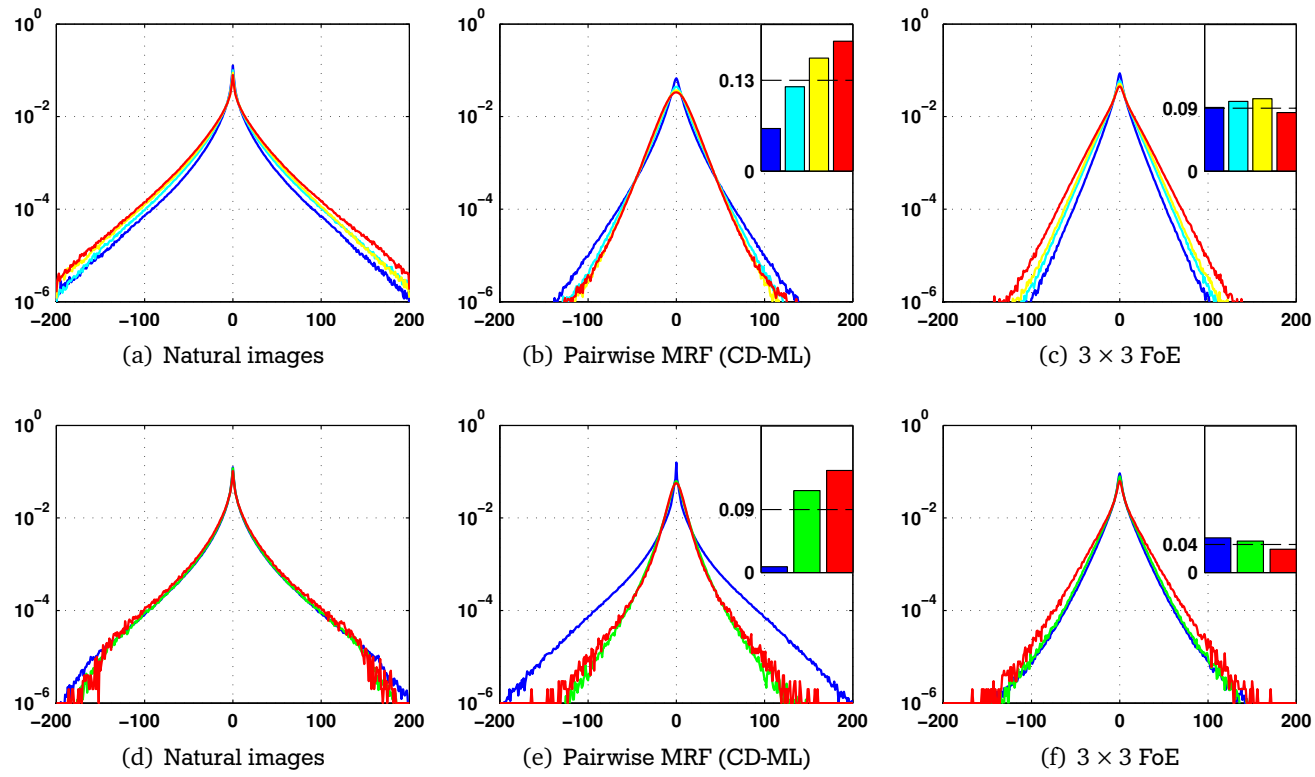


(e) 5×5 FoE [Roth and Black, 2009]



(f) 15×15 FoE [Weiss and Freeman, 2007] (convolution with circular boundary handling, no pixels removed)

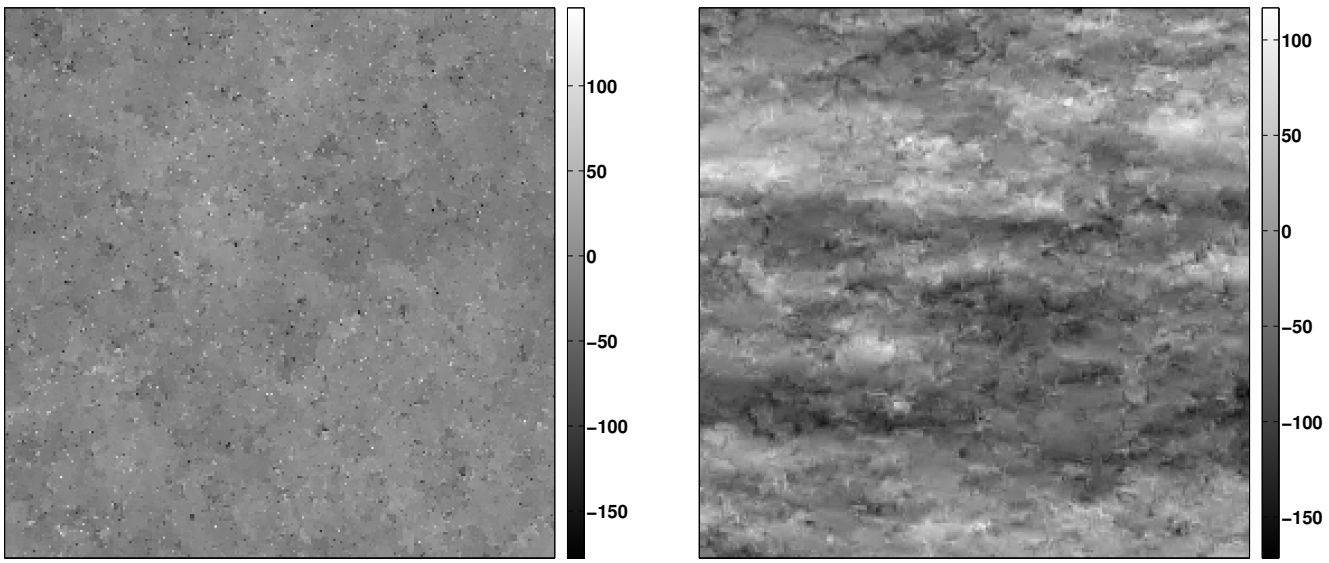
Figure 5.18: Five subsequent samples (left to right) from various MRF models after reaching the equilibrium distribution; the boundary pixels are removed for better visualization. Note that the auxiliary-variable Gibbs sampler mixes rapidly.



(a) Natural images (b) Pairwise MRF (CD-ML) (c) 3 × 3 FoE

(d) Natural images (e) Pairwise MRF (CD-ML) (f) 3 × 3 FoE

Figure 5.19: Random filter statistics and scale-invariant derivative statistics. *(a–c)* Average marginals of 8 random zero-mean unit-norm filters of various sizes (3 × 3 blue, 5 × 5 cyan, 7 × 7 yellow, 9 × 9 orange). *(d–f)* Derivative statistics at three spatial scales (1–blue, 2–green, 4–red; 1 refers to the original scale). The bar charts display the marginal KL-divergence of each feature. The learned pairwise MRF only captures short-range and small-scale statistics well. Our high-order FoE also models long-range and large-scale statistics.



(a) Pairwise MRF (CD-ML) (b) 3 × 3 FoE

Figure 5.20: 246 × 246 pixel sample from our learned models after reaching the equilibrium distribution. The boundary pixels are removed for better visualization.

6 Image Restoration

We compare our learned models with other popular MRF priors in image restoration tasks, specifically image denoising and image inpainting. Especially image denoising in the context of i.i.d. Gaussian noise with known standard deviation σ has become a benchmark for MRF priors of natural images, where performance is usually evaluated in terms of peak signal-to-noise ratio (PSNR); we additionally considered the perceptually more relevant structural similarity index (SSIM) [Wang et al., 2004]. We performed image denoising on two different test sets from the Berkeley segmentation dataset [Martin et al., 2001]: Detailed comparisons on a set of 10 images used by Lan et al. [2006], and more extensive experiments for our best performing models on a larger set of 68 images used by Roth and Black [2009]; Samuel and Tappen [2009].

Prior to denoising, we increased the size of the noisy images by mirroring the boundary pixels, 5 pixels when using pairwise MRFs and 9 pixels in case of FoEs (15 pixels for the model by Weiss and Freeman [2007] due to its large filter size). After denoising the enlarged image, we removed the mirrored boundary before computing the PSNR and SSIM values. We did this in order to decrease the influence of the underconstrained boundary pixels in the MRF, which can also affect denoising performance.

We rely on a user-defined mask for image inpainting, where we assume a flat likelihood for all missing pixels which are therefore filled in using the prior alone (cf. Section 2.2.4 and Roth and Black [2009]).

6.1 MAP Estimation

We first considered the commonly used maximum a-posteriori (MAP) estimation, which maximizes

$$p(\mathbf{x}|\mathbf{y};\boldsymbol{\Theta}) = p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{x};\boldsymbol{\Theta})^{\lambda} \tag{6.1}$$

w.r.t. $\mathbf{x}$, where $p(\mathbf{y}|\mathbf{x})$ is the application specific likelihood and λ is an optional regularization weight, which has often been required for MRFs to obtain good application performance (e.g. Roth and Black [2009]). We use conjugate gradients (CG) to maximize $p(\mathbf{x}|\mathbf{y};\boldsymbol{\Theta})$, in particular the implementation by Rasmussen [2006] with at most 5000 line searches.

We compare our learned models against a pairwise MRF whose potential has been fit to the derivative marginals of natural images (Figure 5.16(a)), as well as two Fields of Experts models [Roth and Black, 2009; Weiss and Freeman, 2007]. Table 6.1 shows that although our good generative models perform better than the other MRFs using MAP estimation and no regularization weight, they still do not outperform previous models when using a regularization weight λ (optimized w.r.t. PSNR on the test set for each model). Our SM-trained pairwise MRF shows no particularly good denoising performance using MAP without λ, despite the suggestion by Hyvärinen [2008] that SM might be the optimal learning method for the setting that we consider here (cf. Section 2.3.4).

The poor performance of our good generative models in the context of MAP estimation with optional regularization weight – the "gold standard" for evaluating image priors – may be an explanation why such models have not been used in the literature.

6.2 MMSE Estimation

We propose to perform image restoration with MRFs by computing the *Bayesian minimum mean-squared error estimate* (MMSE)

$$\hat{\mathbf{x}} = \arg\min_{\tilde{\mathbf{x}}} \int ||\tilde{\mathbf{x}} - \mathbf{x}||^2 p(\mathbf{x}|\mathbf{y};\boldsymbol{\Theta})\, d\mathbf{x} = E[\mathbf{x}|\mathbf{y}] \tag{6.2}$$

as an alternative, which is equal to the mean of the posterior distribution. MMSE estimation is desirable because it exploits the probabilistic nature of MRFs by using the uncertainty of the model to find the expected restored image; MAP estimation, on the other hand, will just seek the restored image with the highest probability.

Computing integrals over high-dimensional images is a difficult problem, which is the reason why the MMSE estimate is usually impractical (cf. Nikolova [2007]); although not in our case because the efficient auxiliary-variable Gibbs sampler can be extended to the posterior distribution. We perform image denoising in case of Gaussian noise by alternating between sampling the hidden scale indices $\mathbf{z}$ according to Eq. (3.7) and sampling the image according to

$$\begin{aligned}
p(\mathbf{x}|\mathbf{y},\mathbf{z};\boldsymbol{\Theta}) &\propto p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{x}|\mathbf{z};\boldsymbol{\Theta}) \\
&\propto \mathcal{N}\left(\mathbf{y};\mathbf{x},\sigma^2\mathbf{I}\right) \cdot \mathcal{N}(\mathbf{x};\mathbf{0},\boldsymbol{\Sigma}) \\
&\propto \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{x}\|^2\right) \cdot \exp\left(-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{x}\right) \\
&\propto \exp\left(-\frac{1}{2}\left(-2\mathbf{x}^{\mathrm{T}}\frac{\mathbf{y}}{\sigma^2} + \mathbf{x}^{\mathrm{T}}\left(\frac{\mathbf{I}}{\sigma^2} + \boldsymbol{\Sigma}^{-1}\right)\mathbf{x}\right)\right) \\
&\propto \exp\left(-\frac{1}{2}\left(\mathbf{x} - \widetilde{\boldsymbol{\Sigma}}\frac{\mathbf{y}}{\sigma^2}\right)^{\mathrm{T}} \widetilde{\boldsymbol{\Sigma}}^{-1}\left(\mathbf{x} - \widetilde{\boldsymbol{\Sigma}}\frac{\mathbf{y}}{\sigma^2}\right)\right) \\
&\propto \mathcal{N}\left(\mathbf{x};\widetilde{\boldsymbol{\Sigma}}\mathbf{y}/\sigma^2,\widetilde{\boldsymbol{\Sigma}}\right),
\end{aligned} \tag{6.3}$$

where σ^2 is the noise variance, $\widetilde{\boldsymbol{\Sigma}} = \left(\mathbf{I}/\sigma^2 + \boldsymbol{\Sigma}^{-1}\right)^{-1}$, and $\boldsymbol{\Sigma}$ is defined as in Eq. (3.9). MMSE estimation for image inpainting is possible through conditional sampling (cf. Section 3.1.1), where we sample the missing pixels conditioned on the known ones.

We compute the MMSE estimate from 4 independent Markov chains which we run in parallel to assess sampler convergence by estimating the potential scale reduction; the chains are initialized from over-dispersed starting points: the noisy image and smoothed versions from median, Wiener, and Gauss filtering. After the burn-in phase, we average all subsequent samples for each of the chains individually until the 4 average images are similar to one another; we stop when the difference is less than 1 grayvalue on average. The final MMSE estimate is computed by averaging the samples (at most 1000) from all chains. Figures 6.4 and 6.2(c) and (d) show example results for MMSE-based inpainting and denoising.

Although a single iteration of computing the MMSE via sampling is somewhat slower compared to gradient-based methods, the amount of change at each step is often greater when using a rapidly-mixing sampler such as ours. We find that our simple MATLAB implementation is already practical, and could significantly be sped up by using a more efficient linear equation solver and running even more chains in parallel. Employing multiple independent chains has the additional advantage of reducing the overall correlation of samples used for the MMSE estimate, effectively reducing the the total number of samples required to achieve the same accuracy. Another shortcoming of MAP-based denoising is that the PSNR at the (local) optimum of the posterior is often worse than the highest PSNR encountered during denoising, a problem which cannot be solved for all images by choosing a single regularization weight. We did not encounter this problem with MMSE-based denoising and our good generative models. We also want to remark that using only a single sample for image restoration tasks [Levi, 2009] is much inferior to computing the MMSE.

(a) PSNR in dB

Model	MAP ($\lambda = 1$)		MAP (opt. λ)		MMSE	
	$\sigma = 10$	$\sigma = 20$	$\sigma = 10$	$\sigma = 20$	$\sigma = 10$	$\sigma = 20$
Pairwise (marginal fitting)	28.41	23.99	31.02	26.93	29.73	24.85
Pairwise (ours, SM)	29.43	26.17	**31.44**	**26.98**	29.54	24.52
Pairwise (ours, CD-ML)	**30.45**	**26.57**	30.56	26.66	**32.07**	**28.32**
5×5 FoE [Roth and Black, 2009]	27.92	23.81	**32.63**	**28.92**	29.38	24.95
5×5 FoE [Weiss and Freeman, 2007]	22.51	20.45	32.27	28.47	23.22	21.47
3×3 FoE (ours)	**30.33**	**25.15**	32.19	27.98	**32.85**	**28.91**

(b) SSIM

Model	MAP ($\lambda = 1$)		MAP (opt. λ)		MMSE	
	$\sigma = 10$	$\sigma = 20$	$\sigma = 10$	$\sigma = 20$	$\sigma = 10$	$\sigma = 20$
Pairwise (marginal fitting)	0.789	0.600	0.873	0.748	0.835	0.637
Pairwise (ours, SM)	0.830	0.712	**0.890**	**0.754**	0.824	0.620
Pairwise (ours, CD-ML)	**0.860**	**0.725**	0.859	0.733	**0.904**	**0.809**
5×5 FoE [Roth and Black, 2009]	0.763	0.595	**0.913**	**0.833**	0.826	0.657
5×5 FoE [Weiss and Freeman, 2007]	0.515	0.445	0.903	0.820	0.564	0.489
3×3 FoE (ours)	**0.838**	**0.638**	0.909	0.798	**0.923**	**0.839**

Table 6.1: Average denoising results for 10 test images [Lan et al., 2006].

Model	Learning	Inference	PSNR in dB		SSIM	
			average	std. dev.	average	std. dev.
5×5 FoE [Roth and Black, 2009]	CD	MAP w/λ	27.44	2.36	0.746	0.080
5×5 FoE [Samuel and Tappen, 2009]	discrimin.	MAP	27.86	2.09	0.776	0.051
Pairwise (ours)	CD-ML	MMSE	27.55	2.11	0.761	0.048
3×3 FoE (ours)	CD	MMSE	**27.95**	2.30	**0.788**	0.059

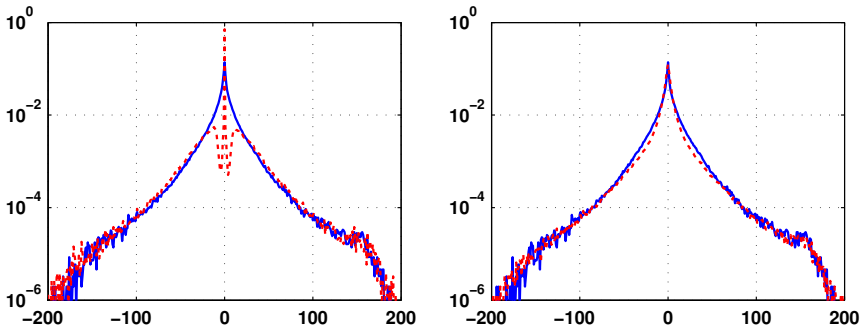
Table 6.2: Denoising results for 68 test images [Roth and Black, 2009; Samuel and Tappen, 2009] ($\sigma = 25$).

Table 6.1 compares MMSE estimation against MAP estimation, the latter with and without a regularization weight; Figure 6.2 shows denoising examples of all the considered models, each using the inference method that yields best performance. MMSE estimation applied to our good generative models outperforms MAP estimation even with an optimal regularization weight; note that MMSE estimation does not require a regularization weight.

These findings are supported by more extensive experiments on 68 test images [Roth and Black, 2009; Samuel and Tappen, 2009]; see Table 6.2 for the quantitative results and Figure 6.5 for a qualitative example. Using MMSE-based denoising, even our pairwise MRF (learned via CD-ML) outperforms the 5×5 FoE of Roth and Black [2009] using MAP with an optimal regularization weight; our learned 3×3 FoE even surpasses the performance of Samuel and Tappen [2009]. This is astonishing because the FoE by Samuel and Tappen [2009] is discriminatively trained to maximize MAP-based denoising performance, and additionally uses larger cliques and more experts. A revealing per-image comparison (Fig. 6.3) between the denoising results of their FoE (using MAP) and the results of our 3×3 FoE (using MMSE) shows a performance advantage for our approach, especially in terms of improved SSIM values.

Figures 6.6–6.11 show additional denoising examples for 6 of the 68 images (Tab. 6.2), which illustrate that MMSE estimation for our good generative models performs well on relatively smooth and strongly textured images. We can conclude that MMSE estimation allows application-neutral generative MRFs to compete with MAP-based denoising-specific discriminative MRFs.

Even more, MMSE-based image restoration solves another problem that has plagued MAP inference for a long time: the incorrect statistics of restored images. MAP solutions to image restoration tasks are often piece-wise constant, which manifests itself in incorrect image statistics (cf. Fig. 6.1(a) and Woodford



(a) MAP, KLD $= 1.05$ (b) MMSE, KLD $= 0.03$

Figure 6.1: Average derivative statistics of 10 denoised test images (dotted red, obtained with our pairwise MRF trained via CD-ML) and of corresponding clean originals (solid blue) for $\sigma = 10, 20$.

et al. [2009]). A recent work by Woodford et al. [2009] introduced a new statistical model that enforces certain statistical properties of the MAP estimate, by paying the price of abandoning the established MRF framework and having to use a rather complex inference procedure. Figure 6.1(b) shows that using MMSE estimation instead of MAP inference is already sufficient to obtain correct statistics of the restored image, and therefore solves this long-standing problem "for free".

(a) Original image

(b) Noisy image ($\sigma = 10$),
PSNR = 28.23dB, SSIM = 0.846

(c) Learned pairwise MRF (CD-ML),
MMSE, PSNR = 31.51dB, SSIM = 0.938

(d) Learned 3×3 FoE,
MMSE, PSNR = 32.40dB, SSIM = 0.947

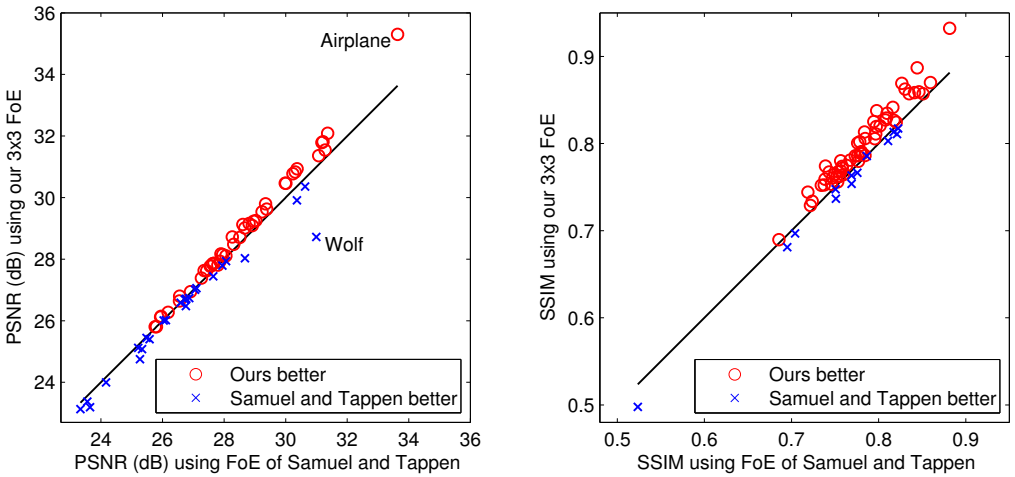
(e) Learned pairwise MRF (SM),
MAP w/λ, PSNR = 31.13dB, SSIM = 0.932

(f) Pairwise MRF (marginal fitting),
MAP w/λ, PSNR = 30.81dB, SSIM = 0.924

(g) 15×15 FoE [Weiss and Freeman, 2007],
MAP w/λ, PSNR = 31.44dB, SSIM = 0.925

(h) 5×5 FoE [Roth and Black, 2009],
MAP w/λ, PSNR = 32.40dB, SSIM = 0.946

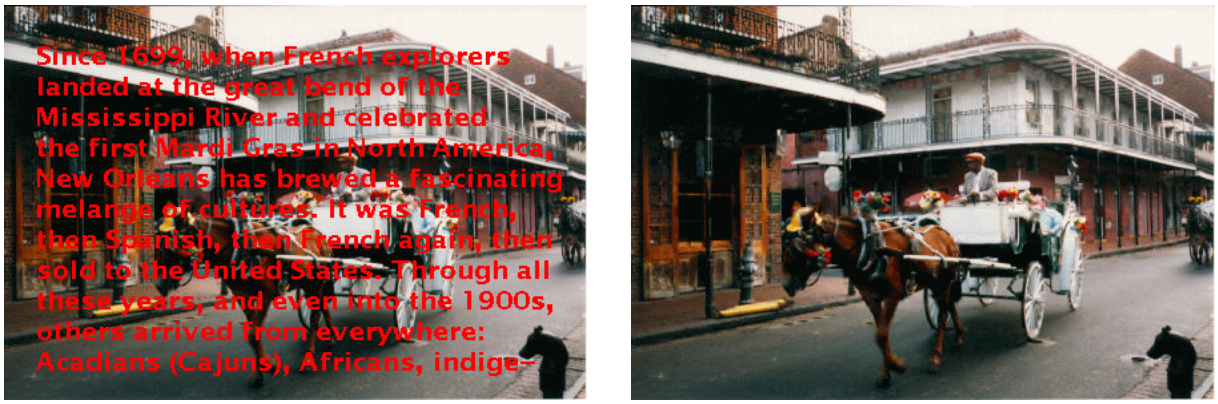
Figure 6.2: Image denoising example: Comparison of all models considered in Table 6.1, each using the inference method that yields best results.



(a) PSNR (b) SSIM

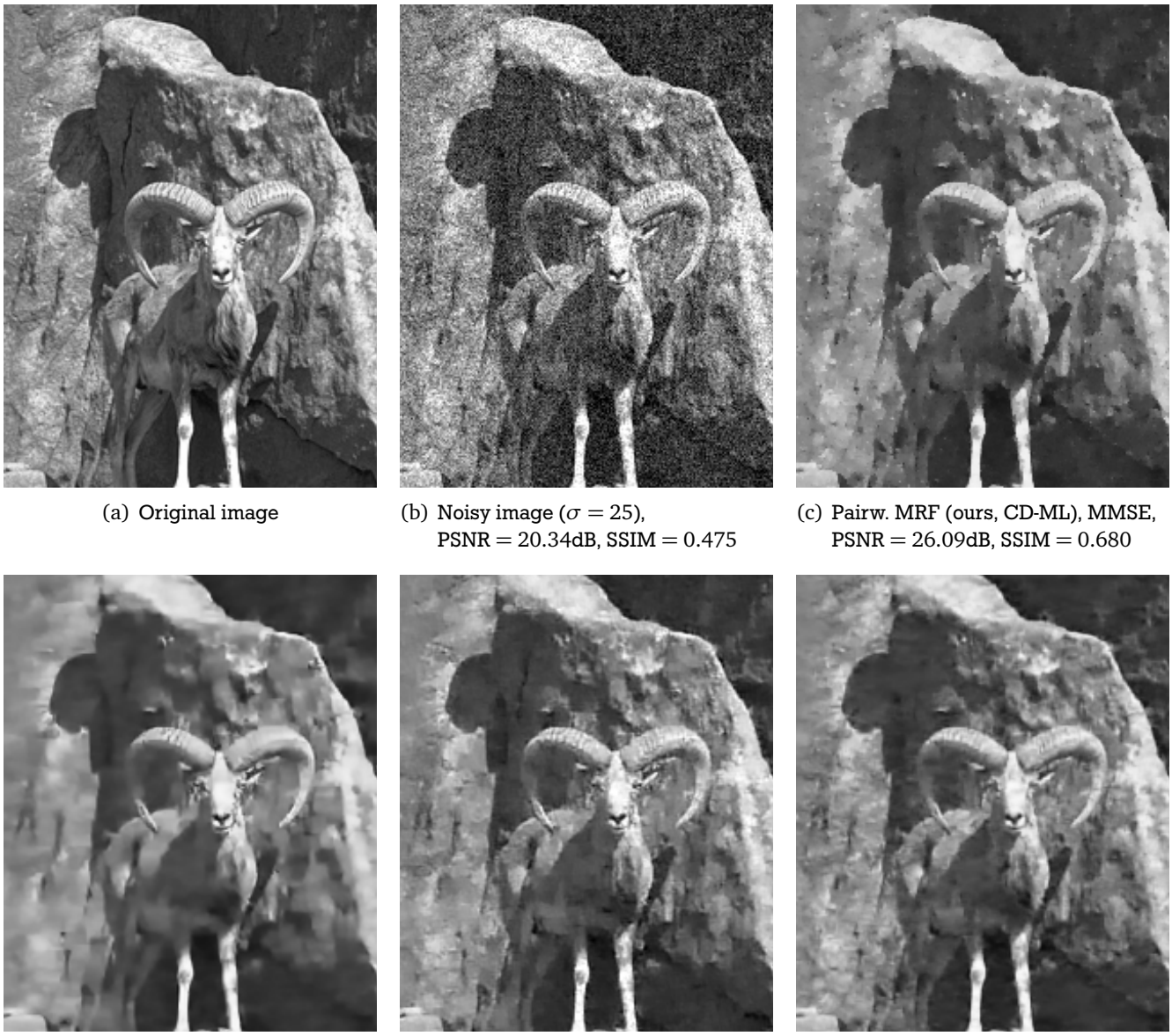
Figure 6.3: Comparing the denoising performance ($\sigma = 25$) in terms of *(a)* PSNR and *(b)* SSIM for 68 test images between our 3×3 FoE (using MMSE) and the 5×5 FoE from Samuel and Tappen [2009] (using MAP). A red circle above the black line means performance is better with our approach. The labels "Airplane" and "Wolf" refer to the respective test image names in Section 6.3.

(a) Original photograph (b) Restored with our pairwise MRF (CD-ML)



(c) Original photograph (d) Restored with our 3×3 FoE

Figure 6.4: MMSE-based image inpainting with our good generative models.



(a) Original image

(b) Noisy image ($\sigma = 25$), PSNR $= 20.34$dB, SSIM $= 0.475$

(c) Pairw. MRF (ours, CD-ML), MMSE, PSNR $= 26.09$dB, SSIM $= 0.680$

(d) 5×5 FoE [Roth and Black, 2009], MAP w/λ, PSNR $= 25.36$dB, SSIM $= 0.592$

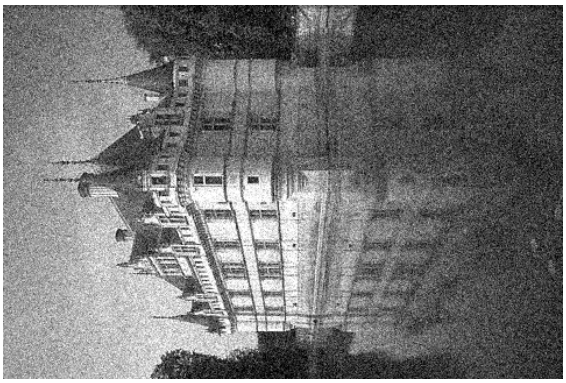
(e) 5×5 discrimin. FoE [Samuel and Tappen, 2009], MAP, PSNR $= 26.19$dB, SSIM $= 0.686$

(f) 3×3 FoE (ours), MMSE, PSNR $= 26.27$dB, SSIM $= 0.689$

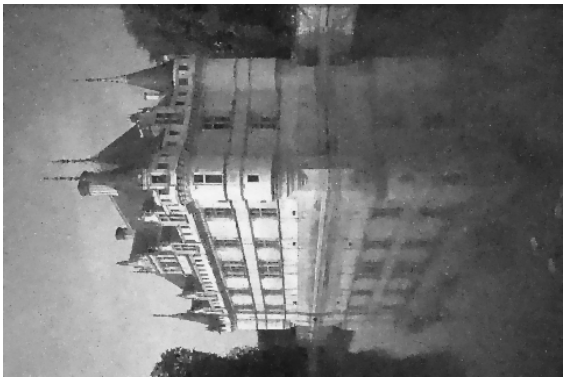
Figure 6.5: Image denoising example (cropped): State-of-the-art-performance from good generative models and MMSE estimation. Note how our learned models with MMSE preserve much of the rock texture, whereas the 5×5 FoE [Roth and Black, 2009] tends to oversmooth.

6.3 Additional Denoising Examples

(a) Original



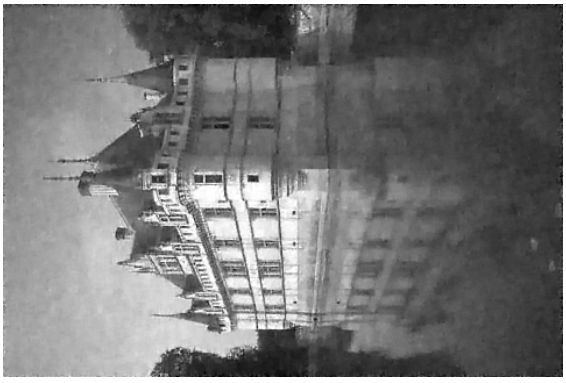
(b) Noisy ($\sigma = 25$), PSNR $= 20.29$dB, SSIM $= 0.310$



(c) Pairwise (ours), PSNR $= 28.38$dB, SSIM $= 0.788$

(d) 3×3 FoE (ours), PSNR $= 28.70$dB, SSIM $= 0.829$

(e) 5×5 FoE [Roth and Black, 2009], PSNR $= 28.52$dB, SSIM $= 0.816$



(f) 5×5 FoE [Samuel and Tappen, 2009], PSNR $= 28.51$dB, SSIM $= 0.809$

Figure 6.6: Denoising results for test image "Castle": *(c, d)* MMSE, *(f)* MAP w/λ, *(e)* MAP.

(a) Original

(b) Noisy ($\sigma = 25$), PSNR = 20.22dB, SSIM = 0.297

(c) Pairwise (ours), PSNR = 29.03dB, SSIM = 0.773

(d) 3×3 FoE (ours), PSNR = 29.79dB, SSIM = 0.820

(e) 5×5 FoE [Roth and Black, 2009], PSNR = 29.16dB, SSIM = 0.794

(f) 5×5 FoE [Samuel and Tappen, 2009], PSNR = 29.35dB, SSIM = 0.802

Figure 6.7: Denoising results for test image "Birds": *(c, d)* MMSE, *(f)* MAP w/λ, *(e)* MAP.

(a) Original

(b) Noisy ($\sigma = 25$), PSNR = 20.71dB, SSIM = 0.507

(c) Pairwise (ours), PSNR = 26.49dB, SSIM = 0.794

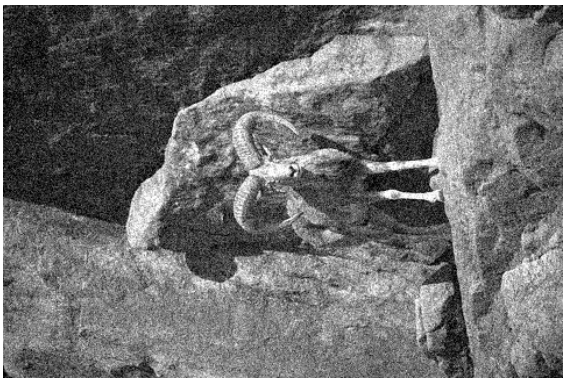
(d) 3×3 FoE (ours), PSNR = 27.00dB, SSIM = 0.813

(e) 5×5 FoE [Roth and Black, 2009], PSNR = 26.84dB, SSIM = 0.792

(f) 5×5 FoE [Samuel and Tappen, 2009], PSNR = 27.06dB, SSIM = 0.817

Figure 6.8: Denoising results for test image "LA": *(c, d)* MMSE, *(f)* MAP w/λ, *(e)* MAP.

(a) Original



(b) Noisy ($\sigma = 25$), PSNR $= 20.34$dB, SSIM $= 0.475$

(c) Pairwise (ours), PSNR $= 26.09$dB, SSIM $= 0.680$

(d) 3×3 FoE (ours), PSNR $= 26.27$dB, SSIM $= 0.689$

(e) 5×5 FoE [Roth and Black, 2009], PSNR $= 25.36$dB, SSIM $= 0.592$

(f) 5×5 FoE [Samuel and Tappen, 2009], PSNR $= 26.19$dB, SSIM $= 0.686$

Figure 6.9: Denoising results for test image "Goat": *(c, d)* MMSE, *(f)* MAP w/λ, *(e)* MAP.

(a) Original

(b) Noisy ($\sigma = 25$), PSNR = 22.44dB, SSIM = 0.278

(c) Pairwise (ours), PSNR = 28.79dB, SSIM = 0.830

(d) 3×3 FoE (ours), PSNR = 28.72dB, SSIM = 0.834

(e) 5×5 FoE [Roth and Black, 2009], PSNR = 28.52dB, SSIM = 0.820

(f) 5×5 FoE [Samuel and Tappen, 2009], PSNR = 30.99dB, SSIM = 0.810

Figure 6.10: Denoising results for test image "Wolf": *(c, d)* MMSE, *(f)* MAP w/λ, *(e)* MAP.

(a) Original

(b) Noisy ($\sigma = 25$), PSNR $= 20.21$dB, SSIM $= 0.136$

(c) Pairwise (ours), PSNR $= 33.89$dB, SSIM $= 0.848$

(d) 3×3 FoE (ours), PSNR $= 35.28$dB, SSIM $= 0.931$

(e) 5×5 FoE [Roth and Black, 2009], PSNR $= 35.00$dB, SSIM $= 0.938$

(f) 5×5 FoE [Samuel and Tappen, 2009], PSNR $= 33.63$dB, SSIM $= 0.881$

Figure 6.11: Denoising results for test image "Airplane": *(c, d)* MMSE, *(f)* MAP w/λ, *(e)* MAP.

7 Summary and Conclusions

We build increasingly more sophisticated models of the world. Parametric statistical models are no exception, which usually implies dealing with high-dimensional data and many model parameters that need to be tuned to fit the data. In order to be flexible, many probabilistic models in practice do not rely on standard distributions and are unnormalizable due to high-dimensional data. Choosing model parameters manually becomes increasingly more impractical as model complexity grows, but standard estimators like maximum likelihood only work with normalized statistical models, or require to draw samples from the model – which is often difficult and computationally demanding. Hence, there is a growing need for general purpose estimators, like score matching, to learn high-dimensional unnormalized statistical models from training data.

Assessing the inherent quality of a (learned) model – independent of a specific application – remains a problem in general, however, and will not be solved by the advent of new learning methods. Since we presume unnormalized statistical models, sampling is again largely the only standard approach to investigate the generative properties, i.e. how well the model actually represents the data.

All of the aforementioned applies to MRFs as well; the likelihood cannot be computed due to the intractable partition function, and likelihood-bounds can only give limited insight since they may not be tight enough to allow model comparison (as in our case). Sampling allowed us to compare MRFs through their generative properties and also made us realize that commonly used MRFs priors are poor generative models.

Using an MRF with flexible potentials, we find that contrastive divergence can be used to learn good generative models; we were however unable to accomplish the same using score matching. We hypothesize that SM is unsuitable to learn MRF image priors under realistic conditions, due to the required heavy-tailed potentials which are presumably not smooth enough to work well with this estimator. Another re-formulation or alteration of the SM objective function may alleviate the problems we observed. Score matching nevertheless is an interesting approach and further research needs to investigate under which conditions it is fruitful.

To the best of our knowledge, we report for the first time which potentials are optimal for generative pairwise MRF models of natural images. For high-order MRFs in the Fields of Experts framework, we learn significantly improved generative models – the statistics of the model features are however not perfectly captured yet. Hence, further research needs to address finding better parametric potentials for high-order MRFs, and specifically better experts in the context of FoEs.

We also showed the need to address boundary issues in high-order MRFs, a problem which requires further attention; it may be advantageous to learn distinct potential functions for cliques that encompass underconstrained boundary pixels.

We furthermore suggest that MAP estimation may largely to be blamed why good generative model have not been used in practice. Our good generative models performed on par or worse than poor generative models in the context of MAP estimation for image denoising (using a regularization weight) – which has become a standard benchmark for MRF image priors. When using suitable inference techniques like MMSE estimation, which make use of the uncertainty in probabilistic models, we showed that good generative properties can go hand-in-hand with state-of-the-art application results for image restoration tasks, and can even compete with application-specific discriminatively-trained MRFs. This is remarkable, given the relative simplicity of our model and the research community's focus on MAP-based inference in the recent past.

The excellent performance of the MMSE estimate makes another case for the ability to sample from MRFs. Even if alternative learning methods like score matching can be used to avoid sampling, sampling

may still be one of the best ways of posterior inference in a specific application. Additionally, we showed that the MMSE estimate for image denoising does not exhibit incorrect marginals statistics of the restored image, which is a problem that MAP estimation suffers from.

In the future, it would also be interesting to extend sampling-based MMSE estimation for posterior inference to other applications, such as super-resolution, which show poor results for gradient-based MAP estimation; sampling-based inference, in contrast to gradient-based methods, is not very sensitive to initialization if well-mixing samplers are used. Although we only considered image priors here, we would also expect many of the results of this work to generalize to other models of scenes, such as optical flow.

While the past few years have seen a tendency to move towards neglecting the probabilistic interpretation of MRFs, we think our findings are reason enough to justify further investigation in generative models for low-level vision.

A Mathematical Notation

Scalars	$a, b, c, \ldots, \alpha, \beta, \gamma, \ldots$
Vectors	$\mathbf{x}, \mathbf{y}, \mathbf{z}, \ldots, \boldsymbol{\alpha}_i, \mathbf{w}_{ik}, \mathbf{x}_{(k)} \ldots$
Matrices	$\mathbf{A}, \boldsymbol{\Sigma}, \mathbf{W}_i, \ldots$
Elements of vectors	$x_1, y_i, \alpha_{ij}, [\ldots]_k, \ldots$
Transposed vectors and matrices	$\mathbf{x}^{\mathrm{T}}, \boldsymbol{\omega}_i^{\mathrm{T}}, \ldots \quad \boldsymbol{\Sigma}^{\mathrm{T}}, \mathbf{W}_i^{\mathrm{T}}, \ldots$
Scalar-valued functions	$f(x), g(\mathbf{y}), \phi(x), \ldots$
Vector-valued functions	$\mathbf{f}(x), \boldsymbol{\phi}(x), \boldsymbol{\varphi}_i(x), \ldots$
First and higher-order derivatives of scalar function $f(x)$	$\frac{df(x)}{dx}$ or $f'(x)$, $\frac{d^2f(x)}{dx^2}$ or $f''(x), \ldots$
Derivatives of vector-valued function $\mathbf{f}(x)$	$\mathbf{f}'(x) = \left[f_1'(x), \ldots, f_n'(x)\right]^{\mathrm{T}}, \ldots$
First and higher-order partial derivative of scalar function $f(\mathbf{x})$	$\frac{\partial f(\mathbf{x})}{\partial x_i}, \frac{\partial}{\partial x_i} f(\mathbf{x}), \frac{\partial^2 f(\mathbf{x})}{\partial x_i^2}, \ldots$
Gradient of scalar function $f(\mathbf{x})$ w.r.t. $\mathbf{x}$	$\nabla_{\mathbf{x}} f(\mathbf{x})$
Probability density function (continuous)	$p(x), p(\mathbf{x})$
Conditional probability density of $\mathbf{x}$ given $\mathbf{y}$	$p(\mathbf{x}\|\mathbf{y})$
Probability density given parameters $\theta_1, \ldots, \theta_n$	$p(\ldots; \theta_1, \ldots, \theta_n)$
Expected value of $f(\mathbf{x})$ w.r.t. probability density $p(\mathbf{x})$	$E[f(\mathbf{x})], \langle f(\mathbf{x}) \rangle_{p(\mathbf{x})}, \langle f(\mathbf{x}) \rangle_p$
Expected value of $\mathbf{x}$ w.r.t. cond. probability density $p(\mathbf{x}\|\mathbf{y})$	$E[\mathbf{x}\|\mathbf{y}]$
Expected value of $f(\mathbf{x})$ w.r.t. empirical data $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}\}$	$\langle f(\mathbf{x}) \rangle_{\mathbf{X}}$
Normal distribution	$\mathcal{N}(x; \mu, \sigma^2), \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$

Table A.1: Commonly used mathematical notation.

B Likelihood Bounds for GSM-based FoEs

Let

$$\phi(x;\boldsymbol{\alpha}_i)=\sum_{j=1}^{J}\beta_{ij}\cdot\mathcal{N}(x;0,\sigma_i^2/s_j)=\sum_{j=1}^{J}\frac{\beta_{ij}}{\sigma_{ij}\sqrt{2\pi}}\exp\left(-\frac{x^2}{2\sigma_{ij}^2}\right) \tag{B.1}$$

be a Gaussian Scale Mixture (GSM) where $\sigma_{ij}=\sqrt{\sigma_i^2/s_j}$ and $\beta_{ij}=\exp(\alpha_{ij})/\sum_{j'=1}^{J}\exp(\alpha_{ij'})$ for conciseness of notation. Assume that the standard deviations are ordered in increasing magnitude $\sigma_{i1}\leq\sigma_{i2}\leq\cdots\leq\sigma_{iJ}$ and let $E(x;\boldsymbol{\alpha}_i)=-\log\phi(x;\boldsymbol{\alpha}_i)$ be the energy of the GSM. We can then use the *Energy Bound Lemma*[1] [Weiss and Freeman, 2007] to obtain lower and upper bounds

$$\frac{x^2}{2\sigma_{iJ}^2}-\log\left(\sum_{j=1}^{J}\frac{\beta_{ij}}{\sigma_{ij}\sqrt{2\pi}}\right)\leq E(x;\boldsymbol{\alpha}_i)\leq\frac{x^2}{2\sigma_{iJ}^2}-\log\left(\frac{\beta_{iJ}}{\sigma_{iJ}\sqrt{2\pi}}\right) \tag{B.2}$$

on the GSM's energy. Based on this Lemma, Weiss and Freeman derived likelihood bounds for GSM-based Fields of Experts. They however assumed that a single GSM expert is used for all filters of the FoE. In the following, we will derive a simple generalization of their result which drops that assumption, adjusted to our FoE definition from Chapter 3.

If we sum the energies of N GSM experts and use the Energy Bound Lemma from Eq. (B.2), we obtain

$$\left[\sum_{i=1}^{N}\frac{x^2}{2\sigma_{iJ}^2}\right]-\left[\sum_{i=1}^{N}\log\left(\sum_{j=1}^{J}\frac{\beta_{ij}}{\sigma_{ij}\sqrt{2\pi}}\right)\right]\leq\sum_{i=1}^{N}E(x;\boldsymbol{\alpha}_i)\leq\left[\sum_{i=1}^{N}\frac{x^2}{2\sigma_{iJ}^2}\right]-\left[\sum_{i=1}^{N}\log\left(\frac{\beta_{iJ}}{\sigma_{iJ}\sqrt{2\pi}}\right)\right]. \tag{B.3}$$

We then apply the Lemma for all filter responses of each expert's filter $\mathbf{w}_i$ and multiply by -1

$$\begin{aligned}\left[\sum_{k=1}^{K}\sum_{i=1}^{N}\log\left(\sum_{j=1}^{J}\frac{\beta_{ij}}{\sigma_{ij}\sqrt{2\pi}}\right)\right]-\left[\sum_{k=1}^{K}\sum_{i=1}^{N}\frac{\left(\mathbf{w}_{ik}^{\mathrm{T}}\mathbf{x}\right)^2}{2\sigma_{iJ}^2}\right]\geq\\ -\sum_{k=1}^{K}\sum_{i=1}^{N}E\left(\mathbf{w}_{ik}^{\mathrm{T}}\mathbf{x};\boldsymbol{\alpha}_i\right)\geq\\ \left[\sum_{k=1}^{K}\sum_{i=1}^{N}\log\left(\frac{\beta_{iJ}}{\sigma_{iJ}\sqrt{2\pi}}\right)\right]-\left[\sum_{k=1}^{K}\sum_{i=1}^{N}\frac{\left(\mathbf{w}_{ik}^{\mathrm{T}}\mathbf{x}\right)^2}{2\sigma_{iJ}^2}\right]\end{aligned} \tag{B.4}$$

where $\mathbf{w}_{ik}^{\mathrm{T}}\mathbf{x}$ is the result of applying filter $\mathbf{w}_i$ to the k^{th} maximal clique of the image vector $\mathbf{x}\in\mathbb{R}^D$. Exponentiating all sides and multiplying by $e^{-\epsilon\|\mathbf{x}\|^2/2}$ gives

$$\begin{aligned}\left[\prod_{k=1}^{K}\prod_{i=1}^{N}\left(\sum_{j=1}^{J}\frac{\beta_{ij}}{\sigma_{ij}\sqrt{2\pi}}\right)\right]\cdot\left[\exp\left(-\frac{\epsilon}{2}\|\mathbf{x}\|^2-\sum_{i=1}^{N}\sum_{k=1}^{K}\frac{\left(\mathbf{w}_{ik}^{\mathrm{T}}\mathbf{x}\right)^2}{2\sigma_{iJ}^2}\right)\right]\geq\\ e^{-\epsilon\|\mathbf{x}\|^2/2}\prod_{k=1}^{K}\prod_{i=1}^{N}\exp\left(-E\left(\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_i\right)\right)\geq\\ \left[\prod_{k=1}^{K}\prod_{i=1}^{N}\left(\frac{\beta_{iJ}}{\sigma_{iJ}\sqrt{2\pi}}\right)\right]\cdot\left[\exp\left(-\frac{\epsilon}{2}\|\mathbf{x}\|^2-\sum_{i=1}^{N}\sum_{k=1}^{K}\frac{\left(\mathbf{w}_{ik}^{\mathrm{T}}\mathbf{x}\right)^2}{2\sigma_{iJ}^2}\right)\right]\end{aligned} \tag{B.5}$$

[1] Please see Weiss and Freeman [2007] for a proof.

Finally, integrating all sides over $\mathbf{x}$

$$\left[\prod_{i=1}^{N}\left(\sum_{j=1}^{J}\frac{\beta_{ij}}{\sigma_{ij}\sqrt{2\pi}}\right)\right]^{K}\cdot\left[\int\exp\left(-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\left(\epsilon\mathbf{I}+\sum_{i=1}^{N}\sum_{k=1}^{K}\frac{\mathbf{w}_{ik}\mathbf{w}_{ik}^{\mathrm{T}}}{\sigma_{iJ}^{2}}\right)\mathbf{x}\right)d\mathbf{x}\right]\geq$$
$$\int e^{-\epsilon\|\mathbf{x}\|^{2}/2}\prod_{k=1}^{K}\prod_{i=1}^{N}\phi\left(\mathbf{w}_{i}^{\mathrm{T}}\mathbf{x}_{(k)};\boldsymbol{\alpha}_{i}\right)d\mathbf{x}\geq$$
$$\left[\prod_{i=1}^{N}\left(\frac{\beta_{iJ}}{\sigma_{iJ}\sqrt{2\pi}}\right)\right]^{K}\cdot\left[\int\exp\left(-\frac{1}{2}\mathbf{x}^{\mathrm{T}}\left(\epsilon\mathbf{I}+\sum_{i=1}^{N}\sum_{k=1}^{K}\frac{\mathbf{w}_{ik}\mathbf{w}_{ik}^{\mathrm{T}}}{\sigma_{iJ}^{2}}\right)\mathbf{x}\right)d\mathbf{x}\right] \tag{B.6}$$

results in

$$\left[\prod_{i=1}^{N}\left(\sum_{j=1}^{J}\frac{\beta_{ij}}{\sigma_{ij}\sqrt{2\pi}}\right)\right]^{K}\cdot Z_{\mathrm{GFoE}}(\boldsymbol{\Sigma})\geq Z_{\mathrm{GSMFoE}}(\boldsymbol{\Theta})\geq\left[\prod_{i=1}^{N}\left(\frac{\beta_{iJ}}{\sigma_{iJ}\sqrt{2\pi}}\right)\right]^{K}\cdot Z_{\mathrm{GFoE}}(\boldsymbol{\Sigma}) \tag{B.7}$$

where $Z_{\mathrm{GSMFoE}}(\boldsymbol{\Theta})$ is the intractable partition function of the GSM-based FoE. $Z_{\mathrm{GFoE}}(\boldsymbol{\Sigma})$ is the partition function of an FoE with Gaussian experts, which is a multivariate Gaussian distribution with mean zero and covariance matrix

$$\boldsymbol{\Sigma}=\left(\epsilon\mathbf{I}+\sum_{i=1}^{N}\sum_{k=1}^{K}\frac{\mathbf{w}_{ik}\mathbf{w}_{ik}^{\mathrm{T}}}{\sigma_{iJ}^{2}}\right)^{-1}=\left(\epsilon\mathbf{I}+\sum_{i=1}^{N}\frac{\mathbf{W}_{i}\mathbf{W}_{i}^{\mathrm{T}}}{\sigma_{iJ}^{2}}\right)^{-1} \tag{B.8}$$

where $\mathbf{W}_i$ are filter matrices that correspond to a convolution of the image with filter $\mathbf{w}_i$, i.e. $\mathbf{W}_i^{\mathrm{T}}\mathbf{x}=\left[\mathbf{w}_{i1}^{\mathrm{T}}\mathbf{x},\ldots,\mathbf{w}_{iK}^{\mathrm{T}}\mathbf{x}\right]^{\mathrm{T}}=\left[\mathbf{w}_{i}^{T}\mathbf{x}_{(1)},\ldots,\mathbf{w}_{i}^{T}\mathbf{x}_{(K)}\right]^{\mathrm{T}}$. In practice, we compute the logarithm of Eq. (B.7)

$$\left[K\sum_{i=1}^{N}\log\left(\sum_{j=1}^{J}\frac{\beta_{ij}}{\sigma_{ij}\sqrt{2\pi}}\right)\right]\cdot\log Z_{\mathrm{GFoE}}(\boldsymbol{\Sigma})\geq\log Z_{\mathrm{GSMFoE}}(\boldsymbol{\Theta})\geq\left[K\sum_{i=1}^{N}\log\left(\frac{\beta_{iJ}}{\sigma_{iJ}\sqrt{2\pi}}\right)\right]\cdot\log Z_{\mathrm{GFoE}}(\boldsymbol{\Sigma}) \tag{B.9}$$

since the partition function is usually too small to be represented as a double precision floating point number. The partition function $Z_{\mathrm{GFoE}}(\boldsymbol{\Sigma})$ of the multivariate Gaussian is obviously well known and its logarithm can be computed as

$$\log Z_{\mathrm{GFoE}}(\boldsymbol{\Sigma})=\log\left((2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}\right)=\frac{D}{2}\log(2\pi)+\frac{1}{2}\log|\boldsymbol{\Sigma}|=\frac{D}{2}\log(2\pi)-\frac{1}{2}\log\left|\epsilon\mathbf{I}+\sum_{i=1}^{N}\frac{\mathbf{W}_{i}\mathbf{W}_{i}^{\mathrm{T}}}{\sigma_{iJ}^{2}}\right|. \tag{B.10}$$

In general, the log-determinant

$$\log|\mathbf{A}|=\log|\mathbf{L}\mathbf{L}^{\mathrm{T}}|=\log\left(|\mathbf{L}||\mathbf{L}^{\mathrm{T}}|\right)=2\log|\mathbf{L}|=2\log\prod_{m}[\mathrm{diag}(\mathbf{L})]_{m}=2\sum_{m}\log[\mathrm{diag}(\mathbf{L})]_{m} \tag{B.11}$$

of a symmetric positive definite matrix $\mathbf{A}$ can be computed using the Cholesky decomposition $\mathbf{A}=\mathbf{L}\mathbf{L}^{\mathrm{T}}$ in order to avoid numerical problems, where $\mathbf{L}$ is a square lower triangular matrix and $\mathrm{diag}(\mathbf{L})$ denotes the vector of its diagonal elements. This can directly be applied to computing the log-determinant in Eq. (B.10) since $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{-1}$ are both symmetric and positive definite.

If convolution with circular boundary handling is used, the computation of $\log Z_{\mathrm{GFoE}}(\boldsymbol{\Sigma})$ can be made more efficient by employing Fourier transformations. We refer the interested reader to Weiss and Freeman [2007] and Lyu and Simoncelli [2009].

Bibliography

M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Computer Graphics (Proceedings of ACM SIGGRAPH)*, pages 417–424, New Orleans, Louisiana, July 2000. doi: 10.1145/344779.344972.

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

L. Bottou. Stochastic learning. In O. Bousquet and U. von Luxburg, editors, *Advanced Lectures on Machine Learning*, number 3176 in Lecture Notes in Artificial Intelligence, pages 146–168. Springer, Berlin, 2004.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, Nov. 2001. doi: 10.1109/34.969114.

M. Á. Carreira-Perpiñán and G. E. Hinton. On contrastive divergence learning. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, pages 33–40, Barbados, Jan. 2005.

J. Domke, A. Karapurkar, and Y. Aloimonos. Who killed the directed model? In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, Alaska, June 2008.

A. Gelman and D. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 2004.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, Nov. 1984.

G. L. Gimel'farb. Texture modeling by multiple pairwise pixel interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1110–1114, Nov. 1996. doi: 10.1109/34.544081.

G. E. Hinton. Products of experts. In *Proceedings of the Ninth International Conference on Artificial Neural Networks*, volume 1, pages 1–6, Edinburgh, UK, Sept. 1999.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, Aug. 2002. doi: 10.1162/089976602760128018.

J. Huang. *Statistics of Natural Images and Models*. PhD thesis, Brown University, 2000.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–708, Apr. 2005.

A. Hyvärinen. Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *IEEE Transactions on Neural Networks*, 18(5):1529–1531, 2007a.

A. Hyvärinen. Some extensions of score matching. *Computational Statistics and Data Analysis*, 51(5):2499–2512, 2007b.

A. Hyvärinen. Optimal approximation of signal priors. *Neural Computation*, 20(12):3087–3110, 2008.

U. Köster, J. T. Lindgren, and A. Hyvärinen. Estimating Markov random field potentials for natural images. In *Int. Conf. on Independent Component Analysis and Blind Source Separation (ICA)*, Paraty, Brazil, 2009.

S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22 (1):79–86, 1951.

X. Lan, S. Roth, D. P. Huttenlocher, and M. J. Black. Efficient belief propagation with learned higher-order Markov random fields. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Proceedings of the Ninth European Conference on Computer Vision*, volume 3952 of *Lecture Notes in Computer Science*, pages 269–282. Springer, 2006. doi: 10.1007/11744047_21.

A. Lee, D. Mumford, and J. Huang. Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. *International Journal of Computer Vision*, 41(1):35–59, 2001.

E. Levi. Using natural image priors – Maximizing or sampling? Master's thesis, The Hebrew University of Jerusalem, 2009.

A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami, Florida, June 2009. doi: 10.1109/CVPRW.2009.5206815.

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

S. Lyu. Interpretation and Generalization of Score Matching. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, Montreal, Canada, June 2009.

S. Lyu and E. P. Simoncelli. Modeling Multiscale Subbands of Photographic Images with Fields of Gaussian Scale Mixtures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):693–706, 2009. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.107.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, volume 2, pages 416–423, Vancouver, British Columbia, Canada, July 2001. doi: 10.1109/ICCV.2001.937655.

G. Matheron. Modèle séquentiel de partition aléatoire. Technical report, Centre de Morphologie Mathématique, 1968.

M. Nikolova. Model distortions in Bayesian MAP reconstruction. *AIMS Journal on Inverse Problems and Imaging*, 1(2):399–422, 2007.

M. Norouzi, M. Ranjbar, and G. Mori. Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, Nov. 2003. doi: 10.1109/TIP.2003.818640.

C. E. Rasmussen. `minimize.m` – Conjugate gradient minimization, April 2006. URL `http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize/`.

S. Roth. *High-Order Markov Random Fields for Low-Level Vision*. Ph.D. dissertation, Brown University, Department of Computer Science, Providence, Rhode Island, May 2007.

S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 860–867, San Diego, California, June 2005. doi: 10.1109/CVPR.2005.160.

S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, Apr. 2009. doi: 10.1007/s11263-008-0197-6.

D. L. Ruderman. Origins of scaling in natural images. *Vision Research*, 37(23):3385–3398, Dec. 1997. doi: 10.1016/S0042-6989(97)00008-4.

K. G. G. Samuel and M. F. Tappen. Learning optimized MAP estimates in continuously-valued MRF models. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami, Florida, June 2009.

H. Scharr, M. J. Black, and H. W. Haussecker. Image statistics and anisotropic diffusion. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 2, pages 840–847, Nice, France, Oct. 2003. doi: 10.1109/ICCV.2003.1238435.

J. Sohl-Dickstein, P. Battaglino, and M. R. DeWeese. Minimum Probability Flow Learning. 2009. URL `http://arxiv.org/abs/0906.4779`.

A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18(1):17–33, Jan. 2003. doi: 10.1023/A:1021889010444.

M. F. Tappen, B. C. Russell, and W. T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *Proceedings of the 3rd International Workshop on Statistical and Computational Theories of Vision*, Nice, France, Oct. 2003.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Technical Report 649, Department of Statistics, University of California, Berkeley, Sept. 2003.

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr. 2004. doi: 10.1109/TIP.2003.819861.

Y. Weiss and W. T. Freeman. What makes a good model of natural images? In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, June 2007.

M. Welling, G. E. Hinton, and S. Osindero. Learning sparse topographic representations with products of Student-t distributions. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 1359–1366, 2003.

O. J. Woodford, C. Rother, and V. Kolmogorov. A global perspective on MAP inference for low-level vision. In *Proceedings of the Thirteenth IEEE International Conference on Computer Vision*, Miami, Florida, June 2009.

J. W. Woods. Two-dimensional discrete Markovian fields. *IEEE Transactions on Information Theory*, 18 (2):232–240, Mar. 1972.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*, chapter 8, pages 239–236. Morgan Kaufmann Pub., 2003.

S. C. Zhu and D. Mumford. Prior learning and Gibbs reaction-diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1236–1250, Nov. 1997. doi: 10.1109/34.632983.