# Exploiting Inference for Approximate Parameter Learning in Discriminative Fields: An Empirical Study

Sanjiv Kumar, Jonas August, and Martial Hebert

The Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213 USA
{skumar,jonas,hebert}@cs.cmu.edu
http://www.cs.cmu.edu/~{skumar,jonas,hebert}

**Abstract.** Estimation of parameters of random field models from labeled training data is crucial for their good performance in many image analysis applications. In this paper, we present an approach for approximate maximum likelihood parameter learning in discriminative field models, which is based on approximating true expectations with simple piecewise constant functions constructed using inference techniques. Gradient ascent with these updates exhibits compelling limit cycle behavior which is tied closely to the number of errors made during inference. The performance of various approximations was evaluated with different inference techniques showing that the learned parameters lead to good classification performance so long as the method used for approximating the gradient is consistent with the inference mechanism. The proposed approach is general enough to be used for the training of, e.g., smoothing parameters of conventional Markov Random Fields (MRFs).

## 1 Introduction

In language processing, natural image analysis and many other applications, the input data show significant dependencies, which should be modeled appropriately to achieve good classification. In earlier work [1], we presented the Discriminative Random Field (DRF) model for image analysis, which discriminatively models the conditional distribution of the labels given the observed data directly as a Gibbs Field. DRFs allow one to relax the assumption of conditional independence of the observed data, which is invoked commonly in conventional generative MRF frameworks, and were shown to give better classification results than MRFs [1]. DRFs were inspired by Conditional Random Field (CRF), which was proposed by Lafferty et al. [2] and developed to analyze 1D sequence data for which exact maximum likelihood parameters can be computed efficiently, e.g., using iterative scaling [2], quasi-Newton methods [3], etc. Unfortunately, for graphs with loops, which are typical in image analysis, it is generally infeasible to exactly maximize the likelihood with respect to the parameters. Therefore,

a critical issue in applying discriminative fields is the design of effective parameter learning techniques that can operate on arbitrary graphs. The objective of this paper is to address this central issue.

In this work, we approximate the gradients of the log likelihood function directly using the inference techniques. Our experimental results may be summarized by the following two observations: First, *parameter learning* can be achieved by approximating the likelihood gradient using the label estimates obtained through methods such as Maximum A Posteriori (MAP) or Maximum Posterior Marginal (MPM) for the given conditional probability model. Second, good classification performance can be achieved by any of these approximations, so long as the method used for inference matches the method used for approximating the gradient in the parameter learning. We note that this *learning/inference coupling* is reasonable because the usual goal in classification problems is to minimize the number of errors, which is what our gradient approximation does, even though this may not necessarily maximize the likelihood. We also present a new experimental comparison of several learning and inference algorithm combinations for guiding what type of learning approximation should be adopted for a given choice of inference method.

## 2 Discriminative Random Field (DRF)

In this section, we review the formulation of discriminative fields. Although the formulation is general to arbitrary graphs with multiple class labels [4], we will discuss the problem of learning in the context of binary classification on 2D image lattices. Let $\boldsymbol{y}$ be the observed data from an input image, where $\boldsymbol{y} = \{\boldsymbol{y}_i\}_{i \in S}$, $\boldsymbol{y}_i$ is the data from $i^{th}$ site, and $S$ is the set of sites. Let the corresponding labels be given by $\boldsymbol{x} = \{x_i\}_{i \in S}$ where $x_i \in \{-1, 1\}$. The DRF formulation combines local discriminative models to capture the class associations at individual sites with the interactions in the neighboring sites as:

$$P(\boldsymbol{x}|\boldsymbol{y}) = \frac{1}{Z} \exp\left( \sum_{i \in S} \log P'(x_i|\boldsymbol{y}) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} \log P''(x_i, x_j|\boldsymbol{y}) \right), \qquad (1)$$

where $Z$ is the partition function (normalizing constant). Note that both the unary potential, $\log P'(x_i|\boldsymbol{y})$, and the pairwise potential, $\log P''(x_i, x_j|\boldsymbol{y})$, depend explicitly on all the observations $\boldsymbol{y}$. Unlike conventional generative MRFs, where the pairwise potential is a *data-independent* prior over the labels, the pairwise potential in DRFs depend on data $y$ and thus allow *data-dependent* interactions among the labels. Hence, DRFs capture much richer contexts in images. For instance, while the pairwise potential in MRF priors can model smoothness of the labels, DRFs can modulate this smoothness by using local image context, e.g., the smoothness can be deactivated at edges in the image.

In (1), $P'(x_i|\boldsymbol{y})$ and $P''(x_i, x_j|\boldsymbol{y})$ are arbitrary unary and pairwise discriminative classifiers. This view gives us the flexibility to choose domain-specific

discriminative classifiers suitable for specific tasks. In this paper, as in our previous work [1], we use a logistic link to give the local class posterior, i.e., $P'(x_i|\boldsymbol{y}) = \sigma(x_i \boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y}))$, where $\sigma(t) = 1/(1+e^{-t})$. Here $\boldsymbol{w}$ is a parameter vector, and $\boldsymbol{h}_i(\boldsymbol{y})$ is a sitewise feature vector. Similarly, to model $P''(x_i, x_j|\boldsymbol{y})$ we use a pairwise logistic classifier, which can be written in a simplified form as, $P''(x_i, x_j|\boldsymbol{y}) = x_i x_j \boldsymbol{v}^T \boldsymbol{\mu}_{ij}(\boldsymbol{y})$. Here $\boldsymbol{v}$ is a parameter vector, and $\boldsymbol{\mu}_{ij}(\boldsymbol{y})$ is a pairwise feature vector. Note that these choices of discriminative classifiers lead to forms of unary and pairwise potentials that are linear in parameters, similar to the CRFs given in [2]. Therefore, this particular DRF form can be seen as a 2D extension of 1D CRFs. Again, the loops in these 2D graphs require more elaborate parameter learning methods, which is the main concern of this paper. It is interesting to note that by ignoring the dependence of the pairwise potential on the observed data $\boldsymbol{y}$, we obtain the conventional MRF smoothing potential, $\beta x_i x_j$, also known as the Ising model.

## 3 Parameter learning approaches

We take a supervised training approach to learning the parameters of the DRF model. The data required are the observed training images and their corresponding ground-truth labeling (e.g., known segmentation). In this work we focus on the standard maximum likelihood approach to learning the parameters. In the case of DRFs, this implies maximization of the *conditional* likelihood, $\log P(\boldsymbol{x}|\boldsymbol{y}, \theta)$[1].

### 3.1 Maximum likelihood parameter learning

Let $\theta$ be the set of unknown DRF parameters, where $\theta = \{\boldsymbol{w}, \boldsymbol{v}\}$. Given $M$ i.i.d. labeled training images, the maximum likelihood estimates of the parameters are given by maximizing the log-likelihood $l(\theta) = \sum_{m=1}^{M} \log P(\boldsymbol{x}^m|\boldsymbol{y}^m, \theta)$, i.e.,

$$\widehat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{m=1}^{M} \left\{ \sum_{i \in S^m} \log \sigma(x_i^m \boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y}^m)) + \sum_{i \in S^m} \sum_{j \in \mathcal{N}_i} x_i^m x_j^m \boldsymbol{v}^T \boldsymbol{\mu}_{ij}(\boldsymbol{y}^m) - \log Z^m \right\}, \tag{2}$$

where the partition function for the $m^{th}$ image is,

$$Z^m = \sum_{\boldsymbol{x}} \exp \left\{ \sum_{i \in S^m} \log \sigma(x_i \boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y}^m)) + \sum_{i \in S^m} \sum_{j \in \mathcal{N}_i} x_i x_j \boldsymbol{v}^T \boldsymbol{\mu}_{ij}(\boldsymbol{y}^m) \right\}.$$

[1] Under the Bayesian view, maximum likelihood learning generally refers to the maximization of the joint distribution, $P(\boldsymbol{x}, \boldsymbol{y}; \theta) = P(\boldsymbol{y}|\boldsymbol{x}; \theta) P(\boldsymbol{x}; \theta)$, where $P(\boldsymbol{x}; \theta)$ is an explicit prior in a generative model.

Note that $Z^m$ is a function of the parameters $\theta$ and the observed data $\boldsymbol{y}^m$. For learning the parameters using gradient ascent, the derivatives of the log-likelihood are

$$\frac{\partial l(\theta)}{\partial \boldsymbol{w}} = \frac{1}{2}\sum_m \sum_{i \in S^m} (x_i^m - \langle x_i \rangle_{\theta;\boldsymbol{y}^m}) \boldsymbol{h}_i(\boldsymbol{y}^m), \tag{3}$$

$$\frac{\partial l(\theta)}{\partial \boldsymbol{v}} = \sum_m \sum_{i \in S^m} \sum_{j \in \mathcal{N}_i} (x_i^m x_j^m - \langle x_i x_j \rangle_{\theta;\boldsymbol{y}^m}) \boldsymbol{\mu}_{ij}(\boldsymbol{y}^m). \tag{4}$$

Here $\langle \cdot \rangle_{\theta;\boldsymbol{y}^m}$ denotes expectation with $P(\boldsymbol{x}|\boldsymbol{y}^m, \theta)$. Ignoring $\boldsymbol{\mu_{ij}}(\boldsymbol{y}^m)$, gradient ascent with (4) is exactly the learning problem for the smoothing parameter of the Ising model.

Generally the expectations in (3) and (4) cannot be computed analytically due to the combinatorial size of the label space. Sampling procedures such as Markov Chain Monte Carlo (MCMC) can be used to approximate the true expectations. Unfortunately, MCMC techniques have two main problems: a long 'burn-in' period (which makes them slow) and high variance in estimates [5]. Recently data-driven MCMC procedures have been proposed to address the problem of slow computation [6]. An alternative approach to avoid the above MCMC drawbacks, Contrastive Divergence (CD), was proposed by Hinton [5]. In CD, only a single MCMC move is made from the current empirical distribution of the data ($P^0$) leading to new distribution ($P^1$), thus eliminating the need for running the chain beyond burn-in. According to CD, $\langle x_i \rangle_{\theta;\boldsymbol{y}} \approx \langle x_i \rangle^{P^1}_{\theta;\boldsymbol{y}}$ and $\langle x_i x_j \rangle_{\theta;\boldsymbol{y}} \approx \langle x_i x_j \rangle^{P^1}_{\theta;\boldsymbol{y}}$. Even though CD is computationally simple and yields estimates with low variance, the bias in estimates can be a problem [7], which was also verified in our experiments in Section 6. However, this approximation of expectation using a single sample inspired the different approximations we propose in this work, as shown in the next section.

### 3.2 Coupling parameter learning and inference

The approximations defined in the previous section replace the exact gradient of (3) and (4) by $\boldsymbol{J(\theta)} = (\boldsymbol{J_1}(\theta), \boldsymbol{J_2}(\theta))$, where

$$\boldsymbol{J_1}(\theta) = \frac{1}{2}\sum_m \sum_{i \in S^m} (x_i^m - f_i(\theta; \boldsymbol{y}^m)) \boldsymbol{h}_i(\boldsymbol{y}^m), \tag{5}$$

$$\boldsymbol{J_2}(\theta) = \sum_m \sum_{i \in S^m} \sum_{j \in \mathcal{N}_i} (x_i^m x_j^m - g_{ij}(\theta; \boldsymbol{y}^m)) \boldsymbol{\mu}_{ij}(\boldsymbol{y}^m), \tag{6}$$

and $f_i$ and $g_{ij}$ are functions that approximate the true expectations in the gradient. Several approaches have been proposed that compute $f_i$ and $g_{ij}$ using pseudo-marginals [8][9]. In this work, we propose to directly construct $f_i$ and $g_{ij}$ using label estimates obtained through MAP and MPM inference at the current parameter estimates (Section 4.2 and 4.3).

Will the gradient ascent of the likelihood with such approximate gradients still converge? The answer is that, while the approximate gradient ascent is not strictly convergent in general, it is weakly convergent in that it oscillates within a set of good parameters, or converges to a good parameter with isolated large deviations, as shown experimentally in Section 5. But why should the parameters learned using a particular choice of approximating functions yield good classification performance? Informally, if we use for parameter learning the same approximating function $f_i$ that was obtained from inference (e.g., a MAP label estimate), then, given input training labels $\{x_i^m\}$,

$$N_E^\theta = \frac{1}{2} \sum_m \sum_{i \in S^m} |x_i^m - f_i(\theta; \boldsymbol{y}^m))| \tag{7}$$

can be interpreted as the number of errors in classification. Comparing (7) with (5) shows that the approximated gradient is directly related to the number of errors, so long as the *same approximation is used in both parameter learning and inference*. We provide more details in Section 7.1.

## 4 Candidate approximations

We first review the form of $f_i$ and $g_{ij}$ based on pseudo-marginals, and then introduce two approximations directly based on two different inference algorithms for estimating the labels: Maximum A Posteriori (MAP), and Maximum Posterior Marginal (MPM). Given our focus on binary DRFs, approximate MAP estimates were obtained using the min-cut/max-flow algorithms as explained in [1], and the MPM estimates were obtained using the sum-product version of loopy Belief Propagation (BP) [10]. The approximations described below are designed to match these two classes of inference techniques.

### 4.1 Pseudo-Marginal Approximation (PMA)

It is easy to see that if we had true marginal distributions $P_i(x_i|\boldsymbol{y}, \theta)$ at each site $i$ and $P_{ij}(x_i, x_j|\boldsymbol{y}, \theta)$ at each pair of sites $i$ and $j \in \mathcal{N}_i$, we could compute exact expectations using

$$\langle x_i \rangle_{\theta;\boldsymbol{y}} = \sum_{x_i} x_i P_i(x_i|\boldsymbol{y}, \theta) \quad \text{and} \quad \langle x_i x_j \rangle_{\theta;\boldsymbol{y}} = \sum_{x_i, x_j} x_i x_j P_{ij}(x_i, x_j|\boldsymbol{y}, \theta).$$

Since computing exact marginal distributions is in general infeasible, a standard approach is to replace the actual marginals by pseudo-marginals [9]. Here, we again used loopy BP to get these marginals. Since loopy BP assumes a tree approximation of the graph [10], it is expected to produce better approximations of these marginals than mean-field, which assumes the nodes in the graph to be disconnected. McCallum et al. [9] use a similar approximation, where pseudo-marginals estimated using Tree-based Reparametrization (TRP) were used for parameter learning in Factorial CRFs.

### 4.2 Learning with MAP inference: Saddle Point Approximation (SPA)

Here, we propose a very simple approximation inspired by CD [5], but uses MAP label estimates. It is based on approximating the partition function $Z$ with the Saddle Point Approximation (SPA) [11]. According to SPA, $Z$ is approximated such that the summation over all the label configurations $\boldsymbol{x}$ in $Z$ is replaced by the largest term in the sum, which occurs at the most probable label configuration[2]. In other words, if $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} P(\boldsymbol{x}|\boldsymbol{y}, \theta)$, then the SPA implies,

$$Z \approx \exp\left\{\sum_{i \in S} \log \sigma(\hat{x}_i \boldsymbol{w}^T \boldsymbol{h}_i(\boldsymbol{y})) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} \hat{x}_i \hat{x}_j \boldsymbol{v}^T \boldsymbol{\mu}_{ij}(\boldsymbol{y})\right\}.$$

This leads to a very simple approximation to the expectation, i.e., $\langle x_i \rangle_{\theta;\boldsymbol{y}} \approx \hat{x}_i$. Observe that this approximation would be exact if $\boldsymbol{x}$ were Gaussian. If we further assume mean-field decoupling, i.e., $\langle x_i x_j \rangle_{\theta;\boldsymbol{y}} = \langle x_i \rangle_{\theta;\boldsymbol{y}} \langle x_j \rangle_{\theta;\boldsymbol{y}}$, it also follows that $\langle x_i x_j \rangle_{\theta;\boldsymbol{y}} \approx \hat{x}_i \hat{x}_j$. It is interesting to note that with the saddle point approximation of $Z$, the gradient ascent updates are similar to the perceptron-learning type updates used in [12] and [13] in nonprobabilistic settings.

### 4.3 Learning with MPM inference: Maximum Marginal Approximation (MMA)

This is the second approximation based on BP inference in which Maximum Posterior Marginal (MPM) label estimates are used for approximating the expectations. Following the arguments of SPA-based parameter learning in the previous section, one can make a similar approximation of $Z$ such that all the mass of $Z$ is assumed to be concentrated on the maximum marginal configuration, $\tilde{x}_i = \arg\max_{x_i} P_i(x_i|\boldsymbol{y}, \theta)$. The expectations in this case can be written as $\langle x_i \rangle_{\theta;\boldsymbol{y}} \approx \tilde{x}_i$ and $\langle x_i x_j \rangle_{\theta;\boldsymbol{y}} \approx \tilde{x}_i \tilde{x}_j$. Clearly, in the binary case, maximum marginals are just the thresholded sitewise marginals. Thus, MMA can be interpreted as a discrete approximation of PMA. We experimented with both MMA and SPA in order to gain a better understanding of the consequences of discretization (see Section 5 and 7.1).

## 5 Experimental observations: parameter learning

To analyze the convergence behavior of various parameter learning procedures described in the previous section, we learned a DRF model for a binary image denoising application. The aim was to obtain true labels from corrupted binary images. A binary image (leftmost image in the top row of Figure 2) of size $64 \times 64$ pixels was corrupted by two types of noise: Gaussian noise and Bimodal (mixture

[2] Seen from the Boltzmann distribution point of view, for the distribution $P(\boldsymbol{x}|\boldsymbol{y}, \theta)$, this will happen at the zero-temperature limit.

of two Gaussians) noise. For each noise model, 10 noisy images were used as the training set for learning the parameters. The unary and pairwise features were defined as: $\boldsymbol{h}_i(\boldsymbol{y}) = [1, I_i]^T$ and $\boldsymbol{\mu}_{ij}(\boldsymbol{y}) = [1, |I_i - I_j|]^T$ respectively, where $I_i$ and $I_j$ are the pixel intensities at site $i$ and site $j$. The details of the noise parameters for this dataset are given in [1]. Here, the parameter vectors $\boldsymbol{w}$ and $\boldsymbol{v}$ were both two-element vectors, i.e., $\boldsymbol{w} = [w_0 \;\; w_1]^T$, and $\boldsymbol{v} = [v_0 \;\; v_1]^T$.

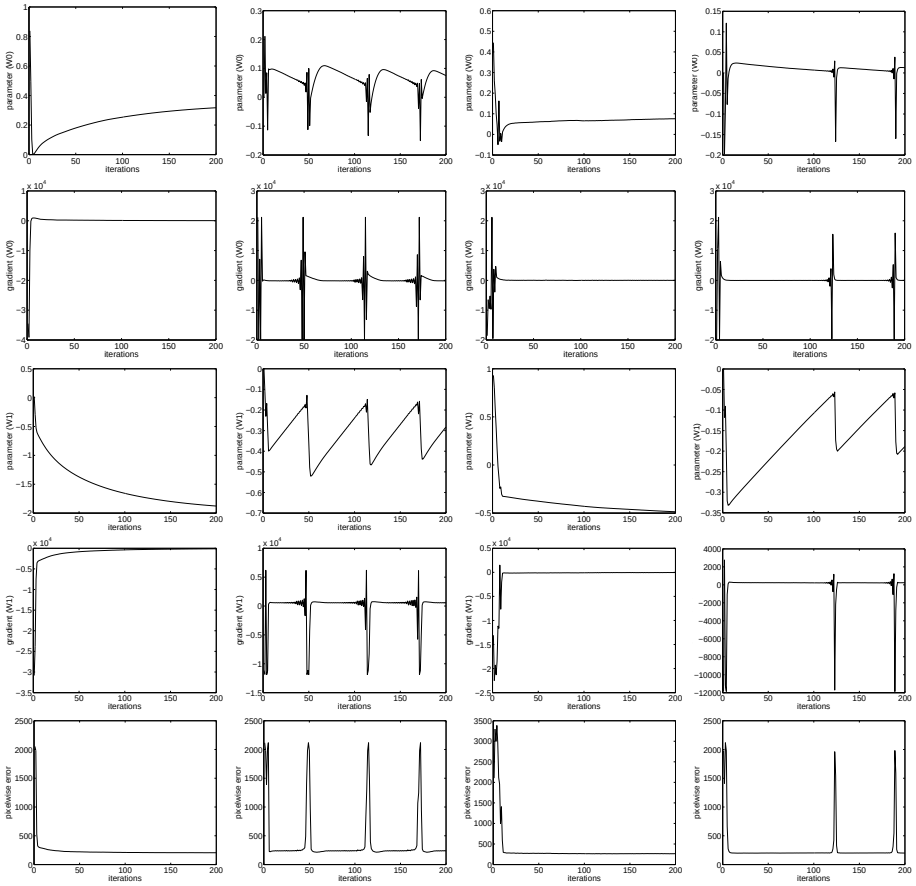
In all the experiments, parameters were initialized from random values and updates were based on gradient ascent. The step size $\eta$ was fixed to a small value ($10^{-5}$). Fig. 1 shows, for each approximation, plots of the approximated gradients and the parameters at each iteration for a typical run with bimodal noise. For brevity we show plots only for parameters $w_0$ and $w_1$. The other parameters behaved similarly. The last row in Figure 1 shows the number of training errors ($N_E^\theta$) made at the current estimate of the parameters using the same inference technique on which a particular gradient approximation is based.

Since the log likelihood in (2) is a convex function of parameters, the final parameter values at convergence will be independent of their initialization in the true gradient ascent. For the PMA based learning, this desirable behavior was seen (Fig. 1 (a)).

For SPA and MMA based learning, an interesting behavior emerges since both of them make discrete approximations of the true expectations. It was found that both SPA and MMA show two different stereotypical patterns of limit cycle convergence depending on the parameter initialization (see Section 7.1). For SPA, in the first case (Figure 1 (b)), the approximated gradients for all the parameters show oscillatory behavior. Initially there are large oscillations in gradients which later settle down to a low gradient zone. The gradients remain in this zone for a relatively long duration before showing large oscillations with changing sign again. Note that this will not occur for the gradient ascent with true gradients if suitably small $\eta$ is chosen. One possibility of damping the oscillations is by annealing $\eta$ according to a decrementing schedule for $\eta$. However such ad-hoc procedures of forcing convergence lead to bias in the final parameters. In the oscillatory case, one can choose any of the parameter selection heuristics commonly used in perceptron learning where convergence is also not guaranteed, e.g., the voted perceptron [14] [13]. In this work we simply used majority vote parameter setting, i.e., the parameters for which the training error was minimum.

The second kind of SPA behavior is seen in Figure 1 (c), where after initial oscillations, the gradients do not show 'periodic' large oscillations again but maintain microscopic oscillations within low gradient zones (not visible in the figure due to the scale of the plots). MMA-based learning behaved similar to the SPA-based learning indicating that these behaviors are related to the discrete, piecewise constant approximation of the actual expectations. An oscillating gradients case for MMA is shown in Figure 1 (d). In Section 7.1 we will discuss these limit cycle behaviors of SPA- and MMA-based learning procedures.

Finally, note that the number of errors for all approximations is small whenever gradient magnitudes are small, which indicates that all the three techniques




(a) PMA (b) SPA-1 (c) SPA-2 (d) MMA

**Fig. 1.** Plots of DRF parameter ($w_0$) updates (top row), and the approximate gradient (second row) for different approximations. PMA shows a converging behavior while SPA shows oscillations which may be large-scale (SPA-1) or small-scale (SPA-2). MMA shows similar behavior as SPA. Rows 3 and 4 show the analogous plots for parameter $w_1$. The last row shows number of errors at each parameter update. The errors are low when the gradient magnitudes are small.

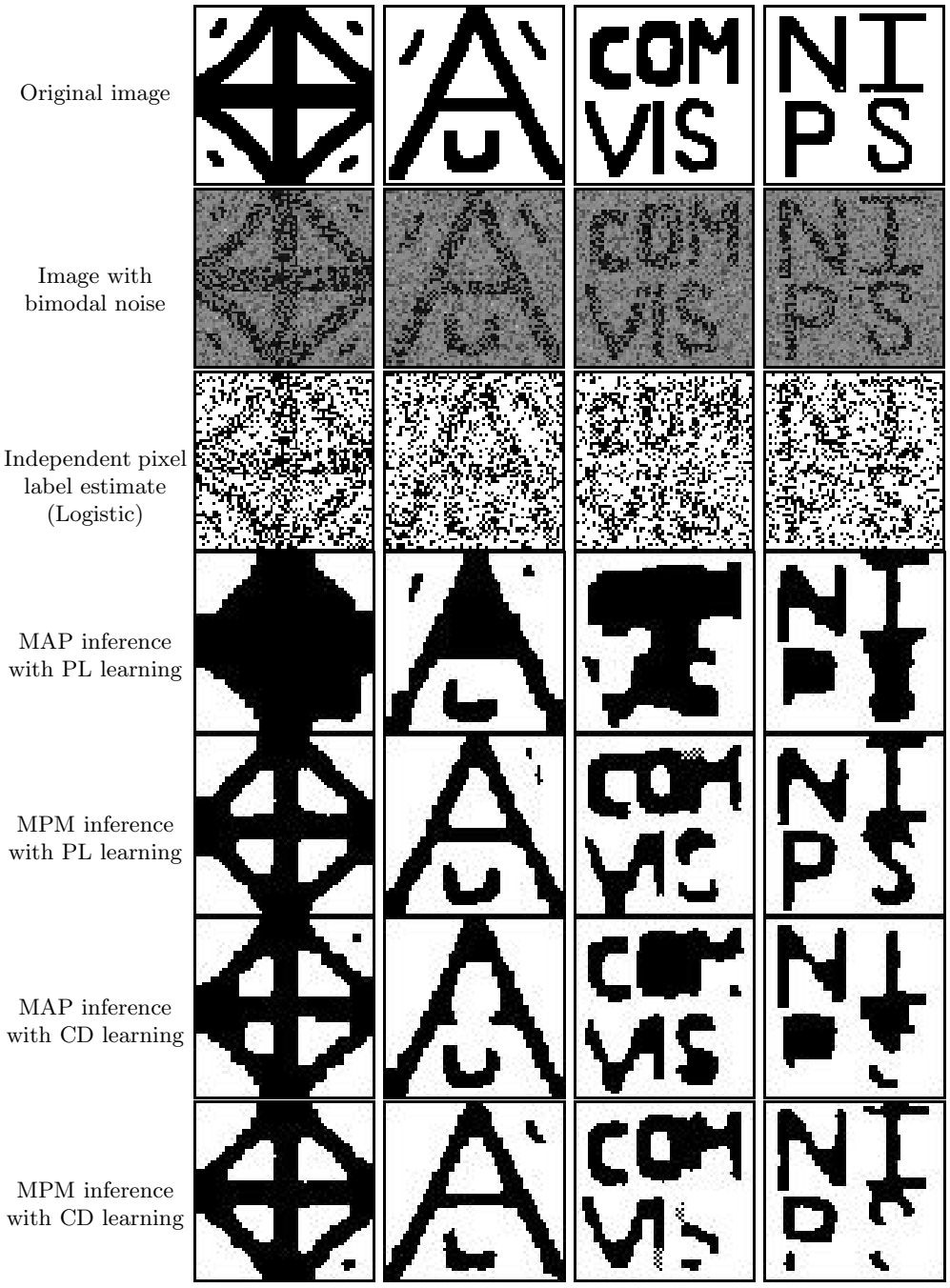



**Fig. 2.** Image denoising results on synthetic images with existing parameter learning methods (MAP: Maximum A Posteriori, MPM: Maximum Posterior Marginal, PL: Pseudo-Likelihood, CD: Contrastive Divergence). Both PL and CD yield poor estimates of the parameters.

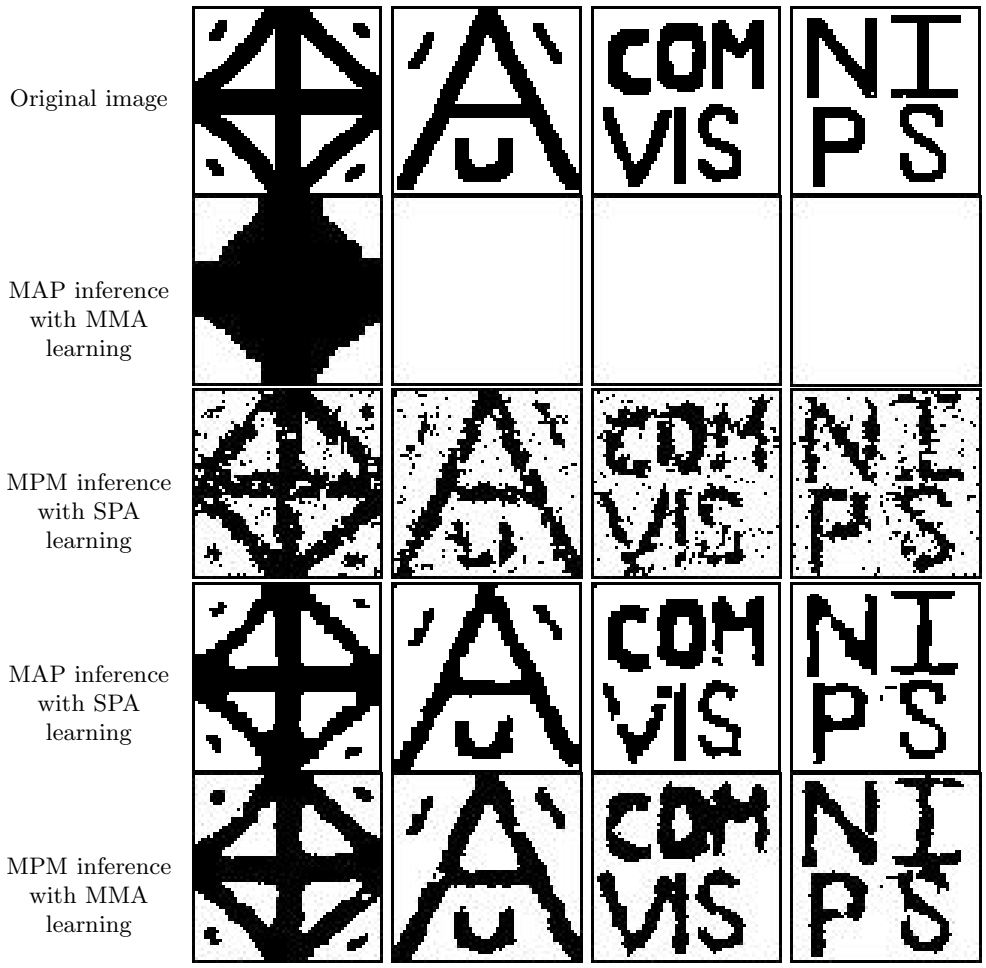



**Fig. 3.** Image denoising results on the noisy images shown in Figure 2 (MAP: Maximum A Posteriori, MPM: Maximum Posterior Marginal, SPA: Saddle Point Approximation, MMA: Maximum Marginal Approximation.) When inference algorithm is mismatched to the parameter learning method, the results are poor (rows 2 and 3). For example, oversmoothing is observed for MAP inference with MMA learning. MPM inference yields undersmoothed results with SPA learning. The results are good whenever the parameter learning is matched with the inference procedure (rows 4 and 5), i.e., MAP inference with SPA learning (both use min-cut) or MPM inference with MMA learning (both use BP).

tend to achieve parameter values that minimize the errors for that particular inference. This is especially interesting in the case of SPA and MMA because of the nature of the approximations. We will compare the performance of the parameter learning procedures with different inference techniques on a separate test set in Section 6.

## 6 Experimental observations: inference

The aim of these experiments was to compare the performance of different parameter learning procedures for a *fixed inference procedure*. For each noise model introduced in Section 5, a test set of 200 noisy images was generated using 50 noisy images each from four ground truth images shown in top row of Figure 2. For comparison, we also obtain the local MAP solution using Iterated Conditional Modes (ICM) [15] which has been shown to be robust to incorrect parameter settings. In addition, we also compare results with parameters learned through pseudo-Likelihood (PL), which uses a factored approximation of the partition function, $Z$, for tractability [1]. All results were computed on a 2.8 GHz CPU with code written in Matlab and C.

Figure 2 shows the denoising performance on four typical test images corrupted by the 'bimodal' noise. The parameters were first learned using existing techniques, i.e., pseudo-likelihood and contrastive divergence. It is clear from the figure that both the techniques give poor results with MAP or MPM inference. The MAP inference with the matched learning technique, i.e., SPA, yields good results as shown in Figure 3. The same is true for MPM inference with MMA learning.

The overall pixelwise errors on the test set are given in Table 1. There are three key observations. Firstly, MAP inference works best with SPA parameters (both use min-cut [16]), and MPM works best with PMA and MMA parameters (all use BP), empirically verifying the claim of *learning/inference coupling*. Secondly, for MAP inference, SPA based learning is also the most efficient approach. The SPA learning is more than 14 times faster than the next most accurate method, PMA. Lastly, MMA is able to learn reasonable parameters for MPM inference (both use BP), at almost half the training time for PMA at the cost of slight decrease in performance from PMA. Note that both PMA and MMA use BP at the learning stage and slightly better results of PMA may be because PMA returns a single converged estimate of the parameters while in MMA one has to heuristically pick the best set of parameters. Better performance may be expected if a better heuristic is used instead of picking the majority voted parameters.

Three main observations help understand the differences between PMA and MMA. Firstly, since MMA is simply a discretized version of PMA, MMA will remain exact even if the pseudo-marginals converge to erroneous values, provided that the ranking of the labels implied by the pseudo-marginals is the same as that implied by the true marginals. This makes MMA more robust to errors in the estimate of marginals when pseudo-marginals tend to give poor estimates

**Table 1.** Pixelwise classification errors (%) on 200 test images ($64 \times 64$ pixels each). The rows show different parameter learning procedures and the columns show different inference techniques used for two different noise models. See text for more.

| Inference methods | | Gaussian noise MAP | MPM | ICM | Bimodal noise MAP | MPM | ICM | Learning time (Sec) |
|---|---|---|---|---|---|---|---|---|
| Parameter Learning Methods | PMA | 2.73 | **2.51** | 3.91 | 6.45 | **5.48** | 17.39 | 1183.13 |
| | SPA | **2.49** | 7.64 | 3.98 | **5.82** | 19.19 | 14.88 | 81.52 |
| | MMA | 34.34 | 2.96 | 4.11 | 26.53 | 5.70 | 16.00 | 635.78 |
| | PL | 3.82 | 3.10 | **3.89** | 17.69 | 7.31 | 22.22 | 299.75 |
| | CD | 3.78 | 2.82 | 4.09 | 8.88 | 6.29 | **8.92** | 206.93 |
| Inference time (Sec) | | 5.52 | 90.04 | 5.20 | 5.96 | 113.84 | 5.20 | |

of the true marginals, e.g., in the presence of strong attractions or repulsion between nodes [17].

Secondly, this discretization accelerates parameter learning since we only need to run BP for enough iterations to stabilize the ranking of the labels, not the exact evaluations of the pseudo-marginals. The former is a coarse (low-resolution) computation, while the latter is a fine (high-resolution) computation. Empirically we noticed that most of the changes in the relative ranking of marginals generally occur in the first few iterations. This partly explains faster learning through MMA in comparison to PMA as shown in Table 1.

Thirdly, while learning the parameters using gradient ascent, MMA gives rise to oscillatory non-convergent behavior. Similar to SPA, this usually requires far fewer iterations of gradient ascent, as typically the *limit-cyclic* behavior in MMA implies that we can stop the gradient ascent iterations after one or two such 'cycles' to obtain sufficiently accurate estimate of the parameters.

An interesting observation is that the MAP inference is very poor with MMA parameters and the same is true for MPM inference with SPA parameters. This further enforces the idea that learning/inference coupling is rooted in minimizing the classification error for a learning/inference pair, rather than maximizing the true likelihood.

As a by-product of this comparison, we find that MPM inference is more robust to the parameters returned by other techniques than MAP which gives significantly worse results with parameters other than SPA and PMA. In addition, the PL and CD parameters generally give bad estimates while ICM does poor inference due to the problem of label initialization.

## 7 Discussion

### 7.1 Dynamics of SPA- and MMA-based learning

What is the origin of the complex dynamics of our proposed parameter learning methods (Figure 1)? In SPA and MMA we replace the expectations $\langle x_i \rangle_{\theta;\boldsymbol{y}}$ and $\langle x_i x_j \rangle_{\theta;\boldsymbol{y}}$ in the true likelihood gradient with approximations $f_i(\theta;\boldsymbol{y})$ and $g_{ij}(\theta;\boldsymbol{y}) = f_i(\theta;\boldsymbol{y}) f_j(\theta;\boldsymbol{y})$ obtained from MAP and MPM label estimates. These

estimates are necessarily discrete values in the set $\{-1, +1\}$, and therefore $f_i(\theta; \boldsymbol{y})$ and $g_{ij}(\theta; \boldsymbol{y})$ are piecewise constant functions of the parameter $\theta \in \Theta$. In other words, the discrete label estimates induce a partition $\{\Theta_k\}$ of parameter space $\Theta$ into a disjoint union $\cup_k \Theta_k$ where $f_i(\theta; \boldsymbol{y})$ and $g_{ij}(\theta; \boldsymbol{y})$ are constant within each cell $\Theta_k$. By substitution, the approximate gradient $\boldsymbol{J}(\theta)$ is also piecewise constant for the same partition $\{\Theta_k\}$ of $\Theta$.

As a consequence, integral curves through vector field $\boldsymbol{J}(\theta)$ will be piecewise linear, with "kinks" at the boundaries between cells, say between $\Theta_k$ and $\Theta_{k'}$. Our approximate gradient ascent with its finite step size will therefore result in a sequence of parameters along piecewise linear trajectories.

One cannot generally expect these trajectories to terminate, as that would require $\boldsymbol{J}(\theta)$ to be identically zero for all $\theta$ in some cell $\Theta_k$. To understand why, consider the double sum in (5) as a product $\frac{1}{2}H(\boldsymbol{x} - \boldsymbol{f})$ of the matrix $H = [\boldsymbol{h}_i(\boldsymbol{y}^m)]$ with vector $\boldsymbol{x} - \boldsymbol{f}$, where $\boldsymbol{x} = [x_i^m]$ and $\boldsymbol{f} = [f_i(\theta; \boldsymbol{y}^m)]$. Now, $\boldsymbol{J}(\theta) = 0$ requires that $\boldsymbol{x} - \boldsymbol{f}$ be in the nullspace of $H$. Because both training labels and the label estimates are discrete, the components $x_i^m - f_i(\theta; \boldsymbol{y}^m)$ of $\boldsymbol{x} - \boldsymbol{f}$ will be one of the integers -2,0, or +2. But the subset of real matrices $H$ that have an integer vector in their nullspace has measure zero, and therefore the possibility that $(\boldsymbol{x} - \boldsymbol{f}) \in \text{nullspace}\, H$ is both unlikely and unstable. Generally, therefore, the approximate gradient ascent using SPA or MMA will not stop.

In the simpler case of true gradient ascent, for a sufficiently small step size $\eta$, the parameter updates converge (without stopping) in a neighborhood of a stationary point of the gradient vector field where the gradient is zero. Why does this ascent converge? Because this gradient vector field is smooth and thus the gradients along the ascent become arbitrarily small near the stationary point, automatically slowing the ascent.

Although our approximate gradients $\boldsymbol{J}(\theta)$ may become small in the vicinity of the true maximum likelihood solution, they cannot become *arbitrarily* small because they are quantized, and therefore the trajectories never slow down beyond some nonzero lower bound. Indeed, our empirical results show a quasi-cyclical behavior of the parameter trajectories. Similar behavior, called *limit cycles*, is common in digital control systems and signal processing, and arises from quantizing states and coefficients in continuous dynamical systems. Such limit cycles have been observed with small oscillations after a single initial transient or with quasi-periodic transients followed by small oscillations. The small oscillation case corresponds to a parameter trajectory passing in a tight loop through nearby portions of abutting cells, say $\Theta_k, \Theta_{k'}$, and $\Theta_{k''}$, which all have small approximate gradients. But there is no guarantee that cells with small and large approximate gradients will not be adjacent. Thus the observed "wild" transient behavior in Figure 1 can arise from several adjacent cells with small approximate gradient linked by cells with large approximate gradient: most of the time is spent in the cells with small approximate gradient, but rapid change occurs in cells with large gradient. To summarize, discretization can account for these limit cycle dynamics.

### 7.2 The role of classification errors in parameter learning

Given these limit cycle dynamics, how may one choose the best parameter along the trajectory? Approximate gradients alone may be misleading, as there may be large approximate gradients nearer to the optimal solution than some small approximate gradients. In true gradient ascent, one may use the likelihood itself as "yard stick" for choosing the best parameter, e.g., at the maximal likelihood observed on the trajectory. The likelihood is also useful in diagnosing pathological dynamics from too large a step size, e.g., if the likelihood decreased significantly. From a dynamical systems perspective, the likelihood exists because the gradient is, by construction, *integrable*.

Instead we have only approximate gradients, which may not be integrable: they may not be the actual gradient of any function. In other words, there may be no approximate likelihood for our approximate gradient!

To overcome this lack of an approximate likelihood, we guide our choice of parameter using the number of classification errors, a widely-employed performance criterion in parameter learning.[3] But what inference algorithm should one use to measure these classification errors? In keeping with the coupling of parameter learning and inference first discussed in Section 3.2, we compute the number of errors $N_E^\theta$ at parameter estimate $\theta$ using the inference method used in the gradient approximation (7), i.e., $N_E^\theta = (1/2)\sum_m \sum_{i\in S} |x_i - f_i(\theta)| = (1/2)\|\boldsymbol{x} - \boldsymbol{f}\|$, where $\|\cdot\|$ is the $L_1$ norm. Formally, this choice is motivated by the following simple bound.

**Lemma 1.** $\|\boldsymbol{J}(\theta)\| \leq cN_E^\theta$, *for some* $c > 0$.

In other words, the number of errors provides an upper bound on the approximate gradient. Note that matching the inference method used in both the number of errors and the approximate gradient is required in the following proof of the lemma.

*Proof.* Recall that $\boldsymbol{J}(\theta) = (\boldsymbol{J_1}(\theta), \boldsymbol{J_2}(\theta))$. Using the form of $\boldsymbol{J_1}(\theta)$ in (5), $\|\boldsymbol{J_1}(\theta)\| \leq R\, N_E^\theta$, where $R = \max_{i,m} \|\boldsymbol{h}_i(\boldsymbol{y}^m)\|$. Now, define the pairwise error $N_P^\theta := (1/2)\sum_m \sum_{i\in S} \sum_{j\in\mathcal{N}_i} |x_i x_j - f_i(\theta;\boldsymbol{y}^m) f_j(\theta;\boldsymbol{y}^m)|$. Using the form of $\boldsymbol{J_2}(\theta)$ in (6) with $g_{ij}(\theta;\boldsymbol{y}^m) = f_i(\theta;\boldsymbol{y}^m) f_j(\theta;\boldsymbol{y}^m)$, it is easy to see that $\|\boldsymbol{J_2}(\theta)\| \leq 2Q\, N_P^\theta$, where $Q = \max_{ijm} \|\boldsymbol{\mu}_{ij}(\boldsymbol{y}^m)\|$. This implies that $\|\boldsymbol{J_2}(\theta)\| \leq 2QdN_E^\theta$, since $N_P^\theta \leq dN_E^\theta$, where $d$ is the maximum degree of the graph, i.e., $d = \max_i |\mathcal{N}_i|$. Combining these results, we have $\|\boldsymbol{J}(\theta)\| = \|\boldsymbol{J_1}(\theta)\| + \|\boldsymbol{J_2}(\theta)\| \leq (R + 2Qd)N_E^\theta$, as required. QED.

This bound is useful in two ways. First, if $\|\boldsymbol{J}(\theta)\|$ is large, then $N_E^\theta$ is also large as verified in the plots in Figure 1. Second, if at some $\theta$, $N_E^\theta$ is small, $\|\boldsymbol{J}(\theta)\|$ will also be small. Thus, for a suitably small step size $\eta$, the parameter change

[3] Ideally, one would like to minimize the generalization error, i.e., expected error on the test set. This is a combination of the training error and the complexity of the learned classifier.

will also be small. This would mean that one will stay in a low error zone for a long period as seen in Figure 1.

Indeed, given the importance we put in the number of classification errors, one might ask whether minimizing $N_E^\theta$ itself should be used as a starting point for *deriving* parameter learning algorithms. Unfortunately, since the number of errors is piecewise constant in the parameters, its gradient is zero except on a set of measure zero. The number of errors is therefore useless to derive a gradient-based learning algorithm as known from the perceptron learning literature [18].

### 7.3 Related Work

The problem of learning the parameters of loopy discriminative graphs has been addressed before under different paradigms. In a non-probabilistic setting, Taskar et al. [19] learn the model parameters by maximizing the margin. Lecun and Huang [20] have described the sufficient conditions for the training of energy-based (unnormalized) graphical models. In our previous work [1], we proposed the use of penalized pseudo-likelihood that gives reasonable estimates of the parameters. However, this needs hand-tuning of the regularizing constant. Finally, taking the Bayesian view, Qi et al.[21] have argued for integrating the parameters while predicting the labels on a test input instead of using a point estimate of the parameters using maximum likelihood. Integrating the parameters, however, is generally a difficult task.

## 8 Conclusion and future work

We have presented an approach for learning the parameters of discriminative field models that uses inference to approximate the gradients used in maximum likelihood learning. We showed that the proposed approximations lead to a limit cycle convergence behavior of the learning procedures. Further, the learned parameters lead to good classification performance so long as the method used for approximating the gradient is consistent with the inference mechanism. We also provided an experimental comparison of commonly used learning and inference techniques for discriminative fields. For MAP inference, SPA based learning was found to be most accurate as well as efficient. Similarly, for MPM inference, PMA and MMA performed best. Although we restricted ourselves to binary fields in this paper, we have already used maximum marginal approximation to successfully learn more than 3000 parameters for multiclass DRFs applied to object detection [4]. We are currently evaluating the performance of the proposed approximate parameter learning procedures with conventional MRFs.

**Acknowledgments** Our thanks to T. Minka for very helpful discussions on SPA based learning. Thanks to V. Kolmogorov, J. Lafferty and R. Mugizi for providing the min-cut code.

## References


1. S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. *in adv. in Neural Information Processing Systems (NIPS)*, 2004.
2. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *In Proc. Int. Conf. on Machine Learning*, 2001.
3. F. Sha and F. Pereira. Shallow parsing with conditional random fields. *In Proc. Human Language Technology-NAACL*, 2003.
4. S. Kumar and M. Hebert. Multiclass discriminative fields for parts-based object detection. *Snowbird Learning Workshop, Utah*, 2004.
5. G. E. Hinton. Training product of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.
6. Z.W. Tu and S.C. Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 24(5):657–673, 2002.
7. C. K. I. Williams and F. V. Agakov. *An Analysis of Contrastive Divergence Learning in Gaussian Boltzmann Machines*. EDI-INF-RR-0120, Informatics Research Report, May 2002.
8. M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree-reweighted belief propagation and approximate ml estimation by pseudo-moment matching. *9th Workshop on AI Stat*, 2003.
9. A. McCallum, K. Rohanimanesh, and C. Sutton. Dynamic conditional random fields for jointly labeling multiple sequences. *NIPS'03 workshop on Syntax, Semantics and Statistic*, 2003.
10. J. S. Yedidia, W. T. Freeman, and Yair Weiss. Generalized belief propagation. *In Advances Neural Information Processing Systems*, 13:689–695, 2001.
11. D. Geiger and F. Girosi. Parallel and deterministic algorithms from mrf's: Surface reconstruction. *IEEE Trans PAMI*, 5(5):401–412, May 1991.
12. Y. LeCun, L. Bottou, Y. Bengio, and P Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, 1998.
13. M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. *In Proc. EMNLP*, 2002.
14. Y. Freund and R. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.
15. J. Besag. On the statistical analysis of dirty pictures. *Journal of Royal Statistical Soc.*, B-48:259–302, 1986.
16. D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of Royal Statis. Soc.*, 51(2):271–279, 1989.
17. Kevin Murphy, Antonio Torralba, and William T. Freeman. Using the forest to see the trees:a graphical model relating features, objects and scenes. *in Advances in Neural Information Processing Systems (NIPS 03)*, 2003.
18. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley, New York, 2001.
19. B. Taskar, C. Guestrin, and D. Koller. Max-margin markov network. *Neural Information Processing Systems Conference (NIPS03)*, 2003.
20. Y. LeCun and F. J. Huang. Loss functions for discriminative training of energy-based models. *AI-Stats*, 2005.
21. Yuan Qi, Martin Szummer, and Thomas P. Minka. Bayesian conditional random fields. *AI & Statistics*, 2005.