Particle Filtered MCMC-MLE with Connections to Contrastive Divergence

Arthur U. Asuncion ASUNCION@ICS.UCI.EDU
Qiang Liu QLIU1@ICS.UCI.EDU
Alexander T. Ihler IHLER@ICS.UCI.EDU
Padhraic Smyth SMYTH@ICS.UCI.EDU
Department of Computer Science, University of California, Irvine, CA, 92697, USA

Abstract

Learning undirected graphical models such as Markov random fields is an important machine learning task with applications in many domains. Since it is usually intractable to learn these models exactly, various approximate learning techniques have been developed, such as contrastive divergence (CD) and Markov chain Monte Carlo maximum likelihood estimation (MCMC-MLE). In this paper, we introduce particle filtered MCMC-MLE, which is a sampling-importance-resampling version of MCMC-MLE with additional MCMC rejuvenation steps. We also describe a unified view of (1) MCMC-MLE, (2) our particle filtering approach, and (3) a stochastic approximation procedure known as persistent contrastive divergence. We show how these approaches are related to each other and discuss the relative merits of each approach. Empirical results on various undirected models demonstrate that the particle filtering technique we propose in this paper can significantly outperform MCMC-MLE. Furthermore, in certain cases, the proposed technique is faster than persistent CD.

1. Introduction

Undirected models such as Boltzmann machines, conditional random fields, and exponential random graph models are useful in many settings, including computer vision (Li, 1994), linguistics (Lafferty et al., 2001), and social network analysis (Robins et al., 2007). These models are often hard to learn exactly, and thus various approximation techniques have been proposed, including pseudolikelihood methods (Besag, 1974), variational methods (Wainwright & Jordan, 2008), and sampling methods (Geyer & Thompson, 1992).

Appearing in *Proceedings of the 27^{th} International Conference on Machine Learning*, Haifa, Israel, 2010.

One such technique is to use both MCMC and importance sampling to perform approximate maximum likelihood estimation. This technique, known as Markov chain Monte Carlo maximum likelihood estimation (MCMC-MLE), has been shown to be more accurate than pseudolikelihood methods on Ising models (Geyer & Thompson, 1992) and is widely used for estimating exponential random graph models (Snijders, 2002; Handcock et al., 2008). A benefit of MCMC-MLE is that the likelihood approximation becomes exact as the number of MCMC samples goes to infinity. However, since this technique is based on importance sampling, the quality of the approximation depends on the distance between the initial and the target distributions. If they are far apart, the samples from the initial distribution may not be able to adequately cover the target space, leading to unreliable estimates (Bartz et al., 2008).

To combat this potential impoverishment of the system's samples, we propose particle filtered MCMC-MLE (or "PF"). Our PF approach applies the sampling-importance-resampling scheme (Smith & Gelfand, 1992) to MCMC-MLE and performs rejuvenating MCMC steps to maintain diversity within the set of samples (Gilks & Berzuini, 2001). We find that PF is able to outperform MCMC-MLE due to these additional steps taken to mitigate degeneracy.

Another well-known technique is contrastive divergence (CD) (Hinton, 2002), which has been successfully applied to many problems, including the learning of Boltzmann machines and deep architectures (Salakhutdinov & Hinton, 2009). One benefit of CD is computational efficiency. Unlike MCMC-MLE, CD does not wait for the Markov chains to reach equilibrium but takes only a few sampling steps to obtain an approximate gradient. When there are no hidden variables in the model, CD with a single-variable sampling step corresponds to maximum pseudolikelihood estimation (MPLE) (Hyvärinen, 2006), and CD with blocked sampling corresponds to maximum composite likelihood estimation (MCLE) (Asuncion et al., 2010). A stochastic approximation variant of CD, also known as persistent contrastive divergence (PCD), has also been developed and has

been shown to improve upon CD (Younes, 1988; Tieleman, 2008). Interestingly, PCD can be recast within the sequential Monte Carlo scheme, and we show later that PCD is a version of PF with importance weights fixed to 1. These connections suggest a unified view, where slight algorithmic choices lead to these different methods.

In the next section, we review MCMC-MLE. Then we introduce PF and highlight the link to PCD. We present experimental results showing that PF improves upon MCMC-MLE and in certain cases also outperforms PCD.

2. A Review of MCMC-MLE

Before delving into our particle filtering technique, we review MCMC-MLE (Geyer & Thompson, 1992). Let us consider models in exponential family form[1]

$$p(x|\theta) = \exp(\theta' g(x))/Z(\theta), \tag{1}$$

where $g(x)$ is a vector of sufficient statistics, the apostrophe denotes transposition, and $Z(\theta)$ is the partition function.

Assume we have N independent observations from the model, $X = \{x^1, x^2, \ldots, x^N\}$, and the task is to find the parameters θ that generated the observed data. The standard approach is maximum likelihood estimation (MLE), which maximizes the log-likelihood,

$$\mathcal{L}(\theta|X) \propto \frac{1}{N}\sum_{i=1}^{N} \theta' g(x^i) - \log Z(\theta). \tag{2}$$

While the ML estimator, $\hat{\theta}^{\text{ml}} \equiv \arg\max_\theta \mathcal{L}(\theta|X)$, has appealing theoretical properties such as asymptotic consistency and statistical efficiency (Lehmann & Casella, 1998), it is typically hard to compute for models with complex dependencies, due to the intractability of $Z(\theta)$.

We can rewrite $Z(\theta)$ in terms of an alternate distribution $p(x|\theta_0)$ (Geyer & Thompson, 1992):

$$Z(\theta) = Z(\theta_0)\sum_x \exp\{(\theta-\theta_0)' g(x)\} p(x|\theta_0). \tag{3}$$

If we are able to obtain S samples from the alternate distribution ($x_0^s \sim p(x|\theta_0)$), a Monte Carlo approximation of $Z(\theta)$ yields the approximate likelihood (note that $Z(\theta_0)$ does not depend on θ and can be ignored):

$$\tilde{\mathcal{L}}(\theta|X) \propto \frac{1}{N}\sum_{i=1}^{N} \theta' g(x^i) - \log \frac{1}{S}\sum_{s=1}^{S} \exp\{(\theta-\theta_0)' g(x_0^s)\}. \tag{4}$$

This likelihood becomes exact when $S \to \infty$.

[1] It is also straightforward to apply the methods in this paper to models with hidden variables, such as RBMs.

Algorithm 1 MCMC-MLE

Initialize θ_0
Sample $\{x^s\} \sim p(x|\theta_0)$
$\theta_1 \leftarrow \theta_0$
for $i = 1$ to max-iterations (or convergence) **do**
  Calculate $\{w^s\}$ via eq. 6, using $\theta_i, \theta_0, \{x^s\}$
  Calculate $\nabla\tilde{L}$ via eq. 5, using $\{w^s\}, \{x^s\}$
  $\theta_{i+1} \leftarrow \theta_i + \eta\nabla\tilde{L}$
end for

MCMC-MLE constructs the approximate likelihood in eq. 4 using MCMC samples from the alternate distribution and then finds the maximizer to that likelihood to obtain the approximate ML estimate $\hat{\theta}^{\text{approx}}$. Since the initial θ_0 may be far from the true parameter θ^* (leading to unreliable estimates), in practice this procedure is iterated a number of times by running MCMC-MLE again with $\theta_0 = \hat{\theta}^{\text{approx}}$ found in the previous iteration. For instance, in the widely used statnet software which estimates exponential random graph models, the default number of MCMC-MLE rounds is 3 (Handcock et al., 2008). Another approach is to initialize parameters at the MPLE ($\theta_0 = \theta^{\text{mple}}$) (Snijders, 2002).

To elucidate the role of importance sampling within MCMC-MLE, we derive the gradient of the likelihood in eq. 4 with respect to θ,

$$\frac{d\tilde{\mathcal{L}}(\theta|X)}{d\theta} = \frac{1}{N}\sum_{i=1}^{N} g(x^i) - \frac{1}{S}\sum_{s=1}^{S} w^s g(x_0^s), \tag{5}$$

where

$$w^s = \frac{\exp\{(\theta-\theta_0)' g(x_0^s)\}}{\frac{1}{S}\sum_s \exp\{(\theta-\theta_0)' g(x_0^s)\}}. \tag{6}$$

These weights $\{w^s\}$ are the Monte Carlo versions of the standard importance weights $\{w_{\text{imp}}^s\}$,

$$w_{\text{imp}}^s = \frac{p(x_0^s|\theta)}{p(x_0^s|\theta_0)} = \frac{\exp\{(\theta-\theta_0)' g(x_0^s)\}}{Z(\theta)/Z(\theta_0)} \tag{7}$$

$$= \frac{\exp\{(\theta-\theta_0)' g(x_0^s)\}}{\sum_x \exp\{(\theta-\theta_0)' g(x)\} p(x|\theta_0)} \tag{8}$$

$$\approx \frac{\exp\{(\theta-\theta_0)' g(x_0^s)\}}{\frac{1}{S}\sum_s \exp\{(\theta-\theta_0)' g(x_0^s)\}} \equiv w^s, \tag{9}$$

where eq. 8 is obtained by applying eq. 3, and eq. 9 is the Monte Carlo approximation of eq. 8. Thus, the second term of the gradient in eq. 5 is estimated via importance sampling. Using the gradient, one can run optimization methods such as gradient ascent to find $\hat{\theta}^{\text{approx}}$. Algorithm 1 shows pseudocode for one round of MCMC-MLE.

3. Our Particle Filtering Approach

We now introduce particle filtered MCMC-MLE and show its close connection to contrastive divergence.

3.1. Particle Filtered MCMC-MLE

The realization that MCMC-MLE is performing importance sampling naturally leads to ways to improve MCMC-MLE. We propose particle filtered MCMC-MLE, which casts the algorithm within a sequential Monte Carlo framework (Doucet et al., 2001). Specifically, our PF approach adapts the sampling-importance-resampling (SIR) technique (Smith & Gelfand, 1992) to the task of estimating the second term of the gradient in eq. 5. At a high level, SIR is a way to perform importance sampling recursively using intermediate distributions, in such a way that the degeneracy of the importance weights is controlled.

In the SIR framework, each particle is initially sampled from the initial distribution $p(x|\theta_0)$ and the weights of all particles are set to 1. Then importance sampling is performed recursively on a sequence of intermediate distributions $\{p(x|\theta_i)\}$. In our task of finding the MLE, we specify the sequence of distributions to be precisely the algorithmic trail $\{\theta_0, \theta_1, \ldots\}$ taken by our optimization algorithm. This specification of intermediate distributions bears resemblance to the technique of Carbonetto et al. (2009).

At each iteration i, each particle weight w^s is calculated recursively in the standard SIR scheme:

$$w^s \leftarrow w^s \frac{p(x^s|\theta_i)}{p(x^s|\theta_{i-1})}. \tag{10}$$

These weights can be normalized such that $\sum_s w^s = 1$. This procedure is equivalent to calculating the weight non-recursively (via eq. 6), based on the current θ and the θ_{prev} when the particles were last resampled. The Monte Carlo approximation of the weights is used here as well.

An important aspect of this approach is monitoring the particle set's quality. The effective sample size (ESS) reveals the number of random samples needed to match the Monte Carlo variation of the particle set. ESS can be approximately computed using the weights (Kong et al., 1994),

$$\text{ESS}(\{w^s\}) = \frac{(\sum_s w^s)^2}{\sum_s (w^s)^2}. \tag{11}$$

An ESS that is low (i.e. below a predetermined threshold) suggests that the importance weights have become degenerate and that resampling and rejuvenation are necessary to replenish the diversity of the particles.

Resampling is performed by sampling S particles from the set $\{x^s\}$ with replacement, with probabilities proportional to $\{w^s\}$. After resampling, the weights are reset to 1. More advanced resampling techniques, such as stratified resampling, are also possible (Kitagawa, 1996). Particles with high relative weights (indicating closeness to the target) are sampled more frequently, while low-weight particles are often eliminated from the set. In the degenerate case where only a few particles have dominating weights, only those particles would be replicated and the diversity of the set would diminish greatly.

Algorithm 2 Particle Filtered MCMC-MLE

```
Initialize θ_0
Sample {x^s} ~ p(x|θ_0)
θ_1 ← θ_0
for i = 1 to max-iterations (or convergence) do
  Calculate {w^s} via eq. 10, using θ_i, θ_{i-1}, {x^s}
  if ESS({w^s}) < threshold then
    Resample {x^s} in proportion to {w^s}
    {w^s} ← 1
    Rejuvenate {x^s} for n MCMC steps, using θ_i
  end if
  Calculate ∇L̃ via eq. 5, using {w^s}, {x^s}
  θ_{i+1} ← θ_i + η∇L̃
end for
```

Rejuvenation combats the problem of degeneracy by applying MCMC move-steps to the particles after the resampling step (Gilks & Berzuini, 2001; Ridgeway & Madigan, 2003). Specifically, for each particle s, an MCMC chain governed by $p(x|\theta_i)$ is initialized at x^s and is advanced for n steps, leading to a new configuration for that particle. Rejuvenation increases the diversity of samples but can be computationally expensive, depending on the number of steps and distributions. To mitigate this cost, we restrict ourselves to rejuvenation steps only after resampling.

Pseudocode for our PF method is in Algorithm 2. One needs to specify the initial parameters θ_0, number of particles S, ESS threshold, rejuvenation length n, learning rate η, and convergence criteria. While the optimization algorithm shown is gradient ascent, one could also use (quasi)Newton methods. Note that as the number of particles goes to infinity, the likelihood gradient becomes exact.

3.2. Connection to Contrastive Divergence

Contrastive divergence (CD) is a popular learning algorithm that has been applied to a variety of models (Hinton, 2002). While CD's objective function is technically a difference of two KL divergences, the practical CD algorithm can be obtained by considering the data log-likelihood. Instead of using eq. 3, we derive the gradient of the likelihood in eq. 2 directly and apply a Monte Carlo approximation:

$$\frac{d\mathcal{L}(\theta|X)}{d\theta} = \frac{1}{N}\sum_{i=1}^{N} g(x^i) - \sum_x g(x)p(x|\theta) \tag{12}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} g(x^i) - \frac{1}{S}\sum_{s=1}^{S} g(x^s). \tag{13}$$

The samples $\{x^s\}$ in eq. 13 are drawn from $p(x|\theta)$ via MCMC. However, to obtain an accurate sample from the distribution, the MCMC chain would need to reach equilibrium. Since the CD philosophy values computational efficiency, CD-n initializes the chains from the data distribution (to get close to the true modes) and then runs the MCMC chain for only n steps, where n is often 1 (i.e. a full sweep over the variables). The rationale behind CD is that only a rough estimate of the gradient is necessary to move the parameters in the right direction. Note that as the number of steps n and the number of samples S go to infinity, the gradient calculation becomes exact. It is interesting to compare this feature to PF, which only needs the number of particles S to go to infinity to obtain an exact gradient.

Persistent contrastive divergence (PCD-n) is a version of CD-n where MCMC chains are persisted across iterations (Tieleman, 2008). At iteration i, each x^s is advanced n steps using the MCMC sampler governed by $p(x|\theta_i)$ where θ_i is the set of parameters at that iteration. At iteration $i+1$, the MCMC chain for each particle x^s is initialized with the end sample of the previous iteration's chain for x^s, and then the chain is advanced n MCMC steps under the distribution $p(x|\theta_{i+1})$. At each step, these configurations $\{x^s\}$ are used to estimate the gradient in eq. 13. A decreasing schedule for η can be used to obtain a stochastic approximation guarantee of asymptotic convergence to the MLE (Younes, 1988; Salakhutdinov, 2009).

Since x^s is approximately sampled from $p(x|\theta_i)$ at the end of iteration i, and since the goal in iteration $i+1$ is to compute an expectation of the sufficient statistics under the distribution $p(x|\theta_{i+1})$ (i.e. eq. 13), a natural extension is to augment PCD by inserting a sampling-importance-resampling step between iterations. Surprisingly, this slightly modified version of PCD is equivalent to our PF algorithm. Within our PF framework, we obtain this modified version of PCD-n if we (1) initialize θ_0 in such a way that sampling from $p(x|\theta_0)$ is easy, (2) set the ESS threshold to be greater than S so that resampling and rejuvenation take place every iteration, and (3) set the length of rejuvenation to be n.

The standard version of PCD can be viewed as PF with all the importance weights fixed to 1. In Algorithm 3, we show the pseudocode of PCD-n, in particle filtering terminology. The resampling step is not needed since all weights are 1.

The n sampling steps within each iteration of PCD can be interpreted as simply being forced rejuvenation steps. PCD does rejuvenation every iteration by construction, but if we follow the PF approach of monitoring the ESS, it is not necessary to rejuvenate every iteration. In iterations when the ESS remains high, PF essentially performs "PCD-0" (since there are 0 MCMC rejuvenation steps). This feature can potentially make PF faster than PCD.

Algorithm 3 PCD-n (in particle filtering terminology)

Initialize θ_0
Sample $\{x^s\} \sim p(x|\theta_0)$ using n steps of MCMC
$\theta_1 \leftarrow \theta_0$
for $i = 1$ to max-iterations (or convergence) **do**
  Fix $\{w^s\} \leftarrow 1$
  if true **then**
    Rejuvenate $\{x^s\}$ for n MCMC steps, using θ_i
  end if
  Calculate $\nabla \tilde{L}$ via eq. 5, using $\{w^s\}, \{x^s\}$
  $\theta_{i+1} \leftarrow \theta_i + \eta \nabla \tilde{L}$
end for

An issue with PF is that it can still have weight degeneracy issues if the step size η moves the parameters too far at each iteration. PCD sidesteps this issue since all particles are fixed to have weights of 1 a priori. An alternative technique to obtain weights approaching 1 is to heuristically introduce a temperature T in the importance weights:

$$w^s = \frac{p^{\frac{1}{T}}(x^s|\theta)}{p^{\frac{1}{T}}(x^s|\theta_0)} = \frac{p(x^s|\theta)}{p^{\frac{1}{T}}(x^s|\theta_0)p^{1-\frac{1}{T}}(x^s|\theta)} \tag{14}$$

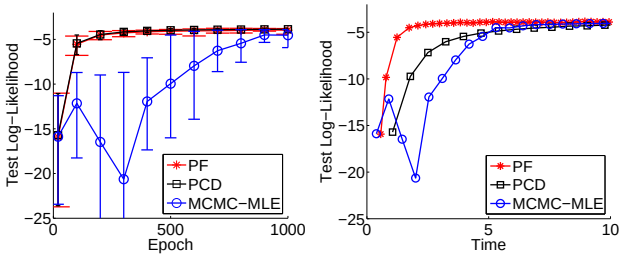
As $T \to \infty$, the weights move towards 1. A way to motivate this heuristic "smoothing" of the weights is that instead of estimating the current health of the particle x^s, we are trying to estimate the future health of the particle after n steps of rejuvenation under distribution $p(x^s|\theta)$, and thus we interpolate between $p(x^s|\theta_0)$ and $p(x^s|\theta)$ in the denominator of eq. 14. It may be possible to adaptively learn the optimal T, but as we will see in the experiment in Figure 4, even using a fixed $T > 1$ can help in cases when weight degeneracy occurs in PF. Nonetheless, this heuristic is not needed as long as η is small enough, and in most of our experiments, we leave $T = 1$.

4. Experimental Analysis

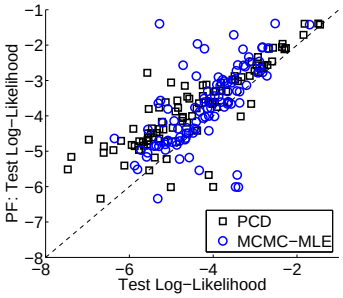
We empirically compare MCMC-MLE, PF, and PCD on undirected models such as visible Boltzmann machines, exponential random graph models, conditional random fields, and restricted Boltzmann machines. Our experimental results suggest that PF generally outperforms MCMC-MLE and in some cases is computationally faster than PCD.

4.1. Visible Boltzmann Machines

Our first model is a visible Boltzmann machine (VBM) of the form $p(x|\theta) = \frac{1}{Z(\theta)} \exp\{\sum_{i<j} \theta_{i,j} x_i x_j\}$ where x is a vector of 15 binary variables (-1/1). This small model allows us to exactly compute log-likelihoods for evaluation. We perform 100 VBM experiments in the following manner. First, the model parameters $\theta_{i,j}$ are randomly drawn from $\mathcal{N}(0, 1)$, and then 500 training cases and 100 test

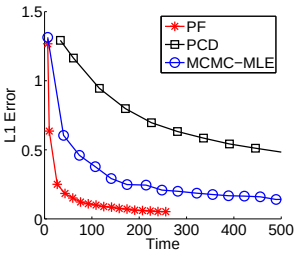


(a) Likelihood over epochs. (b) Likelihood over time.

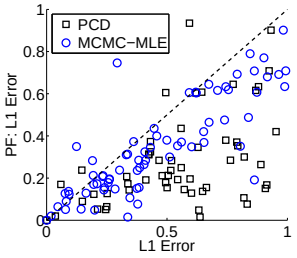


(c) Likelihoods achieved in the time it takes PF to run 2,000 epochs, for 100 different models.

Figure 1. Comparison of MCMC-MLE, PF, and PCD on visible Boltzmann machines with 15 variables.



(a) L1 error over time.



(b) L1 errors, for 100 models.

Figure 2. Comparison of MCMC-MLE, PF, and PCD on exponential random graph models.

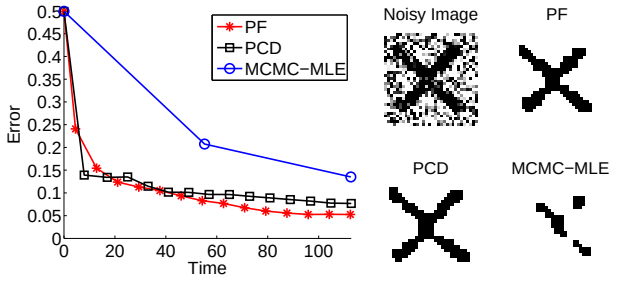
cases are drawn from $p(x|\theta)$. For 10 different initial settings ($\theta_0 \sim \mathcal{N}(0,1)$), we run MCMC-MLE, PF, and PCD on the training set for thousands of epochs, using $S = 50$ chains and an initial rate $\eta = 0.01$ that is gradually decreased. For PCD and PF, the number of rejuvenation steps is $n = 1$ (one pass through the variables). For MCMC-MLE and PF, we use 10 sampling steps to sample particles from the initial distribution $p(x|\theta_0)$. For PF, the ESS threshold is $0.9 \times S$; furthermore, we enable a rejuvenation every 100 epochs. We run MCMC-MLE iteratively as described in Section 2; the algorithm is reinitialized whenever there is convergence to the maximizer $\hat{\theta}^{\text{approx}}$ (using the criteria that the L1 norm of the gradient is less then a threshold) or if 100 iterations are reached. At each reinitialization of MCMC-MLE, 10 steps are used to sample x^s from the distribution at the previous maximizer $p(x|\hat{\theta}^{\text{approx}})$. The chains are persisted across MCMC-MLE rounds.

Figures 1(a) and 1(b) show the average test log-likelihood achieved by each algorithm as a function of the number of epochs and the amount of time, for one of the experiments above. The error bars show the variation between 10 different initial settings of θ_0. Figure 1(a) shows that PF and PCD converge at the same rate, while MCMC-MLE is significantly underperforming due to the lack of rejuvenation of the importance samples. In Figure 1(b), we observe that PF is computationally faster than PCD, since PF is able to skip rejuvenation steps when the ESS is high. MCMC-MLE is slower because it needs to wait for the MCMC chains to reach equilibrium (which is approximately done by using 10 sampling steps). Figure 1(c) compares the log-likelihood achieved by PF after 2,000 epochs (for each of the 100 experiments) to the corresponding log-likelihoods achieved by PCD and MCMC-MLE in the time it takes for PF to reach 2,000 epochs. This result suggests that PF is generally faster than PCD and MCMC-MLE at estimating the parameters for this model.

We also tried other learning rates ($\eta = 0.1$ to 0.001), various amounts of chains ($S = 20$ to 500), and various ESS thresholds, and the results were qualitatively similar. MCMC-MLE can be made almost as fast as PCD by reducing the number of sampling steps from 10 to 5; however, as this number is reduced, we stray further from the main philosophy of MCMC-MLE (which waits until the chains have reached equilibrium) and move closer to the PF approach. The variance of MCMC-MLE in Figure 1(a) suggests that MCMC-MLE is sensitive to parameter initialization.

4.2. Exponential Random Graph Models

We consider exponential random graph models (ERGMs), also known as p* models, which are widely used in social network analysis (Robins et al., 2007). ERGMs take the form $p(x|\theta) = \frac{1}{Z(\theta)} \exp\{\theta' g(x)\}$ where x is a vector of binary variables (0/1) denoting the presence or absence of edges in a social network, and $g(x)$ are graph-based statistics. We use a well-known triad model with three graph-based statistics: number of edges, number of two-stars, and number of triangles (Snijders, 2002); thus, this model has three θ parameters. As in the VBM case, we run 100 different experiments, on networks with 10 nodes and 45 edge variables. Our ground truth parameters θ are randomly drawn from $\mathcal{N}(0,1)$, and we check that θ is not in a degenerate region (where one configuration is dominant). 500 training cases are drawn from $p(x|\theta)$. We run MCMC-MLE, PF, and PCD on the training set for thousands of epochs, using $S = 100$ chains and an initial rate



(a) Error rate over time.

(b) Results of binary image denoising.

Figure 3. Comparison of MCMC-MLE, PF, and PCD on a conditional random field for image segmentation.

$\eta = 0.005$ that is gradually decreased. We initialize each algorithm at approximately the MPLE (found by CD with a single-variable sampling step). For PCD and PF, $n = 1$. For PF, the ESS threshold is $0.9 \times S$. We use 10 sampling steps to sample particles from the initial distribution θ_0. We run MCMC-MLE as described in the VBM case, with 10 sampling steps per reinitialization of the algorithm.

Figure 2(a) shows the L1 error between the ground truth parameters and the parameters estimated by each algorithm, as a function of time, for one of the experiments. As in the VBM case, PF is significantly faster than both PCD and MCMC-MLE. MCMC-MLE performs well relative to PCD, due to the fact that we started with a good initialization for θ_0. When initializing at $\theta_0 = [0, 0, 0]$, we found that MCMC-MLE typically performs much worse than PCD. Figure 2(b) shows the L1 error after 40,000 epochs of PF for each of the 100 experiments, compared to the L1 errors of PCD and MCMC-MLE at the time it takes for PF to reach 40,000 epochs. In virtually all cases, PF outperforms both PCD and MCMC-MLE for these ERGMs.

4.3. Conditional Random Fields

We also test our method on a conditional random field (CRF) for image segmentation (Kumar & Hebert, 2003; Vishwanathan et al., 2006). Let x be a binary image, where $x_j = \pm 1$ is the label of the j^{th} pixel. Let y be a noisy observation of x. In this CRF, the posterior distribution $p(x|y)$ is specified as an Ising model,

$$p(x|y,\theta) = \frac{1}{Z}\exp\Big(\sum_j w^T h_j(y) x_j + \sum_{i\sim j} v^T h_{ij}(y) x_i x_j\Big),$$

where $i \sim j$ indicates that pixel i and pixel j are adjacent in the image. Meanwhile, $h_j(y)$ and $h_{ij}(y)$ are the node features and edge features. In this setting, the features are defined as $h_i(y) = [1, y_i]$ and $h_{ij}(y) = [1, |y_i - y_j|]$.

In our experiment, we use an X-shaped image with 32×32 pixels, and we generate 15 noisy images by adding $\mathcal{N}(0, 1)$ noise on the original image. We use 10 noisy images (totaling 10,240 pixels) for training and 5 images (totaling 5,120 pixels) for testing. We then run MCMC-MLE, PF, and PCD to estimate the parameters $[w, v]$, with $\eta = 0.001$, $S = 100$, and the ESS threshold in PF equal to $0.95 \times S$. For each rejuvenation step, we run 100 steps of random-scan Gibbs sampling. Note that since there are 1,024 variables in our model, 100 steps of random-scan sampling is 1/10 of a normal systematic step. For MCMC-MLE, we use 3×1024 random-scan sampling steps for each reinitialization. We then use loopy BP to estimate the MAP image on test data and we calculate the average error rate of the MAP image with respect to the original image.

Figure 3(a) shows the percentage of errors achieved by each algorithm as a function of time. While PF significantly outperforms MCMC-MLE, it performs similarly to PCD due to a low ESS that frequently triggers the rejuvenation step, causing PF to lose its computational advantage over PCD in this case. Figure 3(b) shows the reconstructed version of the noisy image achieved by each algorithm at time 120, which suggests that MCMC-MLE is much slower at recovering the parameters than either PF or PCD.

4.4. Restricted Boltzmann Machines

Finally, we compare the performance of the algorithms on a restricted Boltzmann machine (RBM) (Hinton & Salakhutdinov, 2006) which takes the form,

$$P(x|w,b,c) = \frac{1}{Z}\sum_h \exp\{-E(x,h|w,b,c)\},$$

where x are observed binary variables (or "visible units"), h are hidden units, and $E(x,h|w,b,c)$ is the energy,

$$E(x,h|w,b,c) = -\sum_{i,j} x_i h_j w_{ij} - \sum_i x_i b_i - \sum_j h_j c_j,$$

where w are the interaction weights, b are the biases of the observed units, and c are the biases of the hidden units.

For this experiment, we use the standard MNIST data (LeCun & Cortes) which consists of handwritten digit images (0-9), each of size 28×28 pixels. The training set contains 60,000 images, while the test set contains 10,000 images. While each pixel has an intensity i from 0 to 255, we binarize each pixel by drawing from a Bernoulli with probability $\frac{i}{255}$; thus, each data case becomes a binary vector of length 784. Each training label is a binary vector of length 10, indicating the particular digit of the training case. We use the "classification" RBM described by Tieleman (2008), where visible units are comprised of the data augmented with the label (totaling 794 visible units in the case of MNIST); furthermore, we use 500 hidden units.

Since RBMs have hidden units, the log-likelihood gradient is slightly different from eq. 5. The first term of the gradient in eq. 5 involves a little more computation (see Tieleman (2008) for details), while the second term of the gradient is an expectation with respect to the joint distribution $p(x, h)$. Nonetheless, our importance sampling techniques can be directly applied to the estimation of this gradient.

We run our algorithms on this RBM for 100 epochs, using $S = 100$ chains and an initial rate of 0.05 which is gradually reduced. We also use a momentum term of 10^{-6} and a weight-decay term of 10^{-5}. We perform mini-batch learning where the training set is divided into 600 batches of 100 data cases, with a gradient ascent step applied after each batch. For PF and PCD, $n = 1$ (one pass over all units). For MCMC-MLE, 10 sampling passes are performed per reinitialization, which occurs at least once every 100 batches (or even more frequently if the convergence criteria is met). The ESS threshold for PF is $0.9 \times S$. Once the parameters are learned, the probability of each label unit given a test case can be computed, and the highest-probability label unit is used to predict the test case's digit.

Figure 4 shows the classification errors achieved by the algorithms as a function of the number of epochs. We observe that MCMC-MLE is performing significantly worse than the other algorithms in this high-dimensional problem since the importance samples are quickly becoming impoverished. We also plot another run of MCMC-MLE where the rate is decreased to $\eta = 0.001$, allowing it to perform better. We also show the results of CD-1. While PF performs better than both MCMC-MLE and CD-1, it is less accurate than PCD in this case. At each iteration, the PF weights become degenerate since the parameters are shifting quickly, and the ESS becomes low. Resampling based on these degenerate weights depletes the particle set, leading to less accurate gradient estimates. However, Figure 4 suggests that PF can nearly match the accuracy of PCD when the temperature in eq. 14 is raised to $T = 10$.

We note that the MNIST data has been widely studied, with advanced classification techniques reaching as low as 0.4% error (LeCun & Cortes); however, our main goal is to simply compare MCMC-MLE, PF, and PCD on this problem.

4.5. General Insights

Several insights can be drawn from these empirical results. While MCMC-MLE can reach accurate estimates relatively quickly if θ_0 is initialized close to the target (as in the ERGM case), we observe inaccurate results and slow convergence if the initialization is arbitrary or if the problem is high-dimensional. In the models we have analyzed, PF significantly outperforms MCMC-MLE in both computational efficiency and the ability to handle the impoverishment of the samples. For models such as VBMs and ERGMs, we find that PF is also computationally faster than PCD since it can skip rejuvenation steps. However, for certain problems (like learning RBMs on MNIST), it seems necessary to perform rejuvenation at each iteration to cope with the fast movement of the parameters, and in these cases, PF loses its computational advantage over PCD. In these cases, a weight temperature may also be useful in preventing the resampling step from depleting the particle set.

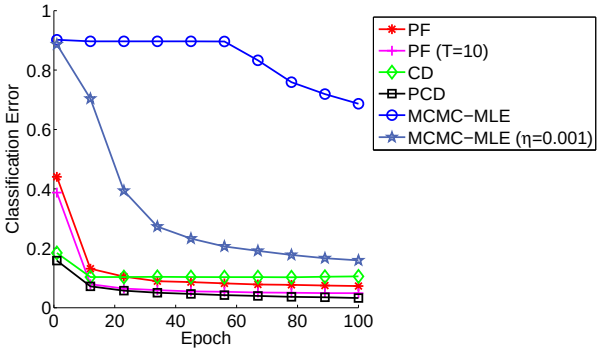


Figure 4. Classification error achieved by MCMC-MLE, PF, and PCD on MNIST digit data, using a restricted Boltzmann machine. We also show CD-1, PF with $T = 10$, and a run of MCMC-MLE with a smaller learning rate $\eta = 0.001$.

5. Related Work

The work of Carbonetto et al. (2009) is the technique most closely related to our own. Their technique uses stochastic approximation and sequential Monte Carlo methods to perform Bayesian inference on graphical models, using the variational framework of minimizing KL divergence. Like our PF method, the intermediate distributions used within their scheme are dictated by the algorithmic path taken when optimizing the variational parameters. The main difference to our work is that we focus on the task of maximum likelihood estimation for undirected graphical models. Another difference is that we incorporate MCMC rejuvenation steps after resampling, which assists in combating degeneracy and allows us to highlight a close connection between our PF approach and the CD framework.

Another related method is annealed importance sampling (AIS) (Neal, 2001) which estimates ratios of partition functions and has been used to evaluate models such as deep belief networks (Salakhutdinov & Murray, 2008). AIS specifies a sequence of intermediate annealed distributions between initial and target distributions. AIS recursively updates importance weights using this sequence, by creating samples which are rejuvenated by each intermediate distribution. Our PF method is similar, with the main difference being the specification of the intermediate distributions; also, we use these weights for parameter estimation.

There also exist other ways to improve MCMC-MLE, including the use of a mixture umbrella distribution that takes advantage of samples drawn from previous rounds of MCMC-MLE (Bartz et al., 2008). In addition, there are techniques for improving persistent contrastive divergence, including parallel tempering (Desjardins et al., 2010), tempered transitions (Salakhutdinov, 2009), and fast weights (Tieleman & Hinton, 2009). Younes (1998) explores the use of Newton's method in PCD. We believe that these advanced techniques could be straightforwardly integrated into our general PF framework.

6. Conclusion

We have introduced a particle filtered version of MCMC-MLE and have shown close connections to the contrastive divergence framework. Empirical results show that our proposed PF approach generally outperforms MCMC-MLE and in some cases provides computational gains over persistent contrastive divergence. We anticipate that the unified view between these different techniques will allow for the design of more efficient and accurate algorithms that utilize advanced sequential Monte Carlo methods.

Acknowledgments

We thank Max Welling for helpful suggestions. This work was supported in part by NSF under grant number IIS-0083489 (PS, AA) and an NSF Graduate Fellowship (AA), by ONR/MURI under grant number N00014-08-1-1015 (PS, AA, QL), by NIH-NIAMS under grant AR-44882 BIRT revision (AI, QL), and by Google (PS).

References

Asuncion, A., Liu, Q., Ihler, A., and Smyth, P. Learning with blocks: Composite likelihood and contrastive divergence. *JMLR W&CP*, 9:33–40, 2010.

Bartz, K., Blitzsten, J., and Liu, J. Monte Carlo maximum likelihood for exponential random graph models: From snowballs to umbrella densities. Technical report, Harvard University, 2008.

Besag, J. Spatial interaction and the statistical analysis of lattice systems. *J. of Royal Stats Soc, Series B*, 36(2):192–236, 1974.

Carbonetto, P., King, M., and Hamze, F. A stochastic approximation method for inference in probabilistic graphical models. In *NIPS 22*, pp. 216–224. 2009.

Desjardins, G., Courville, A., Bengio, Y., Vincent, P., and Delalleau, O. Tempered Markov chain Monte Carlo for training of restricted Boltzmann machines. *JMLR W&CP*, 9:145–152, 2010.

Doucet, A., De Freitas, N., and Gordon, N. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

Geyer, C.J. and Thompson, E.A. Constrained Monte Carlo maximum likelihood for dependent data. *J. of Royal Stats Society. Series B*, 54(3):657–699, 1992.

Gilks, W.R. and Berzuini, C. Following a moving target-Monte Carlo inference for dynamic Bayesian models. *J. of Royal Stats. Society. Series B*, 63(1):127–146, 2001.

Handcock, M.S., Hunter, D.R., Butts, C.T., Goodreau, S.M., and Morris, M. statnet: Software tools for the representation, visualization, analysis and simulation of network data. *Journal of Statistical Software*, 24(1):1548, 2008.

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

Hinton, G. and Salakhutdinov, R. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504, 2006.

Hyvärinen, A. Consistency of pseudolikelihood estimation of fully visible Boltzmann machines. *Neural Computation*, 18 (10):2283–2292, 2006.

Kitagawa, G. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *JCGS*, 5(1):1–25, 1996.

Kong, A., Liu, J.S., and Wong, W.H. Sequential imputations and Bayesian missing data problems. *JASA*, 89:278–288, 1994.

Kumar, S. and Hebert, M. Discriminative fields for modeling spatial dependencies in natural images. In *NIPS 16*, pp. 1531–1538, 2003.

Lafferty, J.D., McCallum, A., and Pereira, F.C.N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pp. 282–289, 2001.

LeCun, Y. and Cortes, C. The MNIST database. URL `http://yann.lecun.com/exdb/mnist/index.html`.

Lehmann, E. L. and Casella, G. *Theory of Point Estimation*. Springer Verlag, 1998.

Li, S.Z. Markov random field models in computer vision. *Lecture Notes in Computer Science*, 801:361–370, 1994.

Neal, R.M. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

Ridgeway, G. and Madigan, D. A sequential Monte Carlo method for Bayesian analysis of massive datasets. *Data Mining and Knowledge Discovery*, 7(3):301–319, 2003.

Robins, G., Snijders, T., Wang, P., Handcock, M., and Pattison, P. Recent developments in exponential random graph (p*) models for social networks. *Social Networks*, 29(2):192–215, 2007.

Salakhutdinov, R. Learning in Markov random fields using tempered transitions. In *NIPS 22*, pp. 1598–1606. 2009.

Salakhutdinov, R. and Hinton, G. Deep Boltzmann machines. *JMLR W&CP*, 5:448–455, 2009.

Salakhutdinov, R. and Murray, I. On the quantitative analysis of deep belief networks. In *ICML*, pp. 872–879, 2008.

Smith, A.F.M. and Gelfand, A.E. Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46(2):84–88, 1992.

Snijders, T.A.B. Markov chain Monte Carlo estimation of exponential random graph models. *J. of Soc. Str.*, 3(2):1–40, 2002.

Tieleman, T. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *ICML*, pp. 1064–1071, 2008.

Tieleman, T. and Hinton, G. Using fast weights to improve persistent contrastive divergence. In *ICML*, pp. 1033–1040, 2009.

Vishwanathan, S., Schraudolph, N., Schmidt, M., and Murphy, P. Accelerated training of conditional random fields with stochastic gradient methods. In *ICML*, pp. 969–976, 2006.

Wainwright, M.J. and Jordan, M.I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

Younes, L. Estimation and annealing for Gibbsian fields. *Ann. Inst. Henri Poincar e-Prob. es et Stats.*, 24(2):269–294, 1988.

Younes, L. Stochastic gradient estimation strategies for Markov random fields. In *SPIE*, volume 3459, pp. 315–325, 1998.