Large Margin Boltzmann Machines*

Xu Miao and **Rajesh P. N. Rao**
Dept. of Computer Science & Engineering
University of Washington, Seattle, USA
{xm,rao}@cs.washington.edu

Abstract

Boltzmann Machines are a powerful class of undirected graphical models. Originally proposed as artificial neural networks, they can be regarded as a type of Markov Random Field in which the connection weights between nodes are symmetric and learned from data. They are also closely related to recent models such as Markov logic networks and Conditional Random Fields. A major challenge for Boltzmann machines (as well as other graphical models) is speeding up learning for large-scale problems. The heart of the problem lies in efficiently and effectively approximating the partition function. In this paper, we propose a new efficient learning algorithm for Boltzmann machines that allows them to be applied to problems with large numbers of random variables. We introduce a new large-margin variational approximation to the partition function that allows Boltzmann machines to be trained using a support vector machine (SVM) style learning algorithm. For discriminative learning tasks, these large margin Boltzmann machines provide an alternative approach to structural SVMs. We show that these machines have low sample complexity and derive a generalization bound. Our results demonstrate that on multilabel classification problems, large margin Boltzmann machines achieve orders of magnitude faster performance than structural SVMs and also outperform structural SVMs on problems with large numbers of labels.

1 Introduction

Boltzmann machines [Hinton and Sejnowski, 1983] have played an important role in the history of machine learning. They were one of the first learning algorithms for undirected graphical models.Originally inspired by biological neural networks and Ising models in statistical physics, they can be viewed as a type of Markov random field (MRF) with weights that are learned from data.

*This work was supported by NGA grant no. HM1582-05-C-0004 , NSF grant no. 0622252, and the Packard Foundation.

A Boltzmann machine is usually trained by *maximum likelihood estimation*. However, the likelihood function involves the normalization or partition function Z (see Section 2), whose exact computation for a general graph is a well-known #P-complete problem. Much research has been devoted to finding an efficient and accurate approximation to the partition function using, for example, *variational inference* or *Markov chain Monte Carlo* (MCMC) methods. Stochastic MCMC algorithms such as Gibbs sampling effectively approximate the likelihood [Neal, 2003; Geman and Geman, 1984] but are typically slow to converge, especially when the number of variables is large. Variational approximations turn the integration problem into an optimization problem and include methods such as *mean field* approximations, *belief propagation* [Yedidia *et al.*, 2005] and *convex relaxation* [Wainwright *et al.*, 2005b]. *Mean field* methods find lower bounds to the partition function, but the bounds are typically not very tight and the methods can get trapped in local maxima. *Loopy belief propagation* has been shown to provide a lower bound on the partition function for *binary attractive MRFs* that is tighter than the *mean field* method [Sudderth *et al.*, 2007]. However, in general graphs, the bounds from belief propagation-based approximation are not yet clear. Wainwright *et al.* have generalized belief propagation to convex relaxation based methods. Their *tree-reweighted message passing* algorithm upper bounds the partition function tightly [Wainwright *et al.*, 2005b] and is guaranteed to converge under certain conditions, but its scalability to large-scale problems remains to be demonstrated.

In this paper, we propose a new learning algorithm for Boltzmann machines that allows fast learning on large numbers of random variables. The contributions of the paper are as follows: (1) Our algorithm is based on a new variational method that upper bounds the partition function of BMs using a hinge functional approximation. The approximation avoids the need to compute the partition function explicitly. (2) In the case of a single output variable, the hinge loss function of *support vector machines* (SVMs) is shown to be an upper bound on the *negative log likelihood* (NLL) loss function, motivating an SVM-style learning algorithm for Boltzmann machines. (3) When conditional likelihood $\Pr(\mathbf{Y}|\mathbf{X})$ is used for learning instead of data likelihood $\Pr(\mathbf{X})$, the resulting *large margin Boltzmann machine* (LMBM) leads to a new and potentially powerful alternative to existing *structural sup-*

port vector machines (SSVMs) [Tsochantaridis *et al.*, 2004; Finley and Joachims, 2008; Joachims *et al.*, to appear]. (4) We analyze the conditions under which LMBMs have low sample complexity and provide a generalization bound. (5) We present results on multi-label classification problems demonstrating the effectiveness and efficiency of the proposed learning algorithm in tackling large-scale problems.

2 Boltzmann Machines

Many important machine learning problems, e.g., language parsing, sequence alignment, image/text segmentation, require modeling the dependencies between random variables. Probabilistic graphical models are well-suited for this purpose. Popular graphical models include *Markov random fields* (MRFs)and related models such as *conditional random fields* (CRF) [Lafferty *et al.*, 2001] and *Markov logic networks* (MLN) [Richardson and Domingos, 2006]. The *Boltzmann machine* can be regarded as an early type of MRF that was inspired by biological neural networks: each binary random variable corresponds to a stochastic neuron-like unit that fires according to some probability conditioned on its neighbors. Neighborhoods correspond to cliques in the graphical model. Thus, a more general definition is as follows:

Definition 2.1. A *Boltzmann machine* (BM) is a graph $G = (\mathbf{V}, \mathbf{E})$,where $\mathbf{V}$ is a set of vertices, representing a set of binary random variables on the domain $\{-1, 1\}^K$, and $\mathcal{C} = \{C_j | C_j \subseteq \mathbf{V}\}$ is a collection of cliques in graph G, each associated with a weight w_j and a feature function $f_j = \prod_{i:V_i \in C_j} v_i$. This likelihood is defined in Equation 1:

$$\Pr(\mathbf{V}|\mathbf{w}) = \frac{1}{Z(\mathbf{w})} e^{\sum_j w_j f_j} \tag{1}$$

where $Z(\mathbf{w}) = \sum_{\mathbf{V}} e^{\sum_j w_j f_j}$ is the partition function.

2.1 Relationship to Other Models

It is possible to show that some recently proposed models can be transformed into a BM. MLNs [Richardson and Domingos, 2006] provide a general framework for combining logic with probability. Weights are attached to first-order logic formulas and weighted formulas are used to construct MRFs. The probability distribution over the variables of a grounded MLN is usually expressed as $\frac{1}{Z} e^{\sum_j w_j n_j}$, where n_j is the number of cliques that make the j^{th} formula satisfied. This distribution has a form similar to a BM (Equation 1) except that $f_j = \vee_{i:V_i \in C_j} v_i$ where $\vee$ denotes the OR function and v_i can take on the values 0 or 1. The equivalence with the BM equation is established by noting that (1) each formula can be instantiated with many variables, resulting in potentially many cliques which have the same weight, and (2) all f_j evaluate to zero or one. It is then easy to show the following:

Remark 2.2. A grounded MLN with weights w_j and realizations $v_i \in \{0, 1\}$ can be transformed into a BM with weights w'_j and realizations $v'_i \in \{-1, 1\}$.

Proof.

$$\begin{aligned}
& \sum_j w_j \vee_{i:V_i \in C_j} v_i \\
= & \sum_j w_j (1 - \prod_{i:V_i \in C_j} (1 - v_i)) \\
= & \sum_j w_j (1 - \prod_{i:V_i \in C_j} (1 - \frac{v'_i + 1}{2})) \\
= & \sum_j w_j \left(1 - \sum_{k:C_k \subseteq C_j} \frac{(-1)^{|C_k|}}{2^{|C_j|}} \prod_{i:V_i \in C_k} v'_i\right) \\
= & \sum_j \left(\sum_{k:C_j \subseteq C_k} \frac{(-1)^{|C_j|+1}}{2^{|C_k|}} w_k\right) \prod_{i:V_i \in C_j} v'_i + \sum_j w_j \\
= & \sum_j w'_j \prod_{i:V_i \in C_j} v'_i + const. \quad \square
\end{aligned}$$

If $\mathbf{V} = \{\mathbf{X}, \mathbf{Y}\}$, where $\mathbf{X}$ are treated as input variables and $\mathbf{Y}$ as output variables, and $\Pr(\mathbf{Y}|\mathbf{X})$ is modeled instead of $\Pr(\mathbf{X}, \mathbf{Y})$, the resulting conditional BM has a form identical to a CRF.

3 Large Margin Boltzmann Machines

The starting point for our efficient learning algorithm for BMs is the following lemma:

Lemma 3.1.

$$\begin{aligned}
\log(e^x + e^{-x}) \quad \leq \quad & \min_{\xi} (x + 2\xi + b) \\
\textit{subject to} \quad & \xi \geq 0 \\
& x \geq \gamma - \xi \\
& \gamma > 0 \\
& b = \log(e^\gamma + e^{-\gamma}) - \gamma
\end{aligned}$$

Proof. As seen in Figure 1, when $x > \gamma$, the log-sum-exp function $\log(e^x + e^{-x})$ is upper bounded by the linear function $x + b$, where b is defined as above. On the other hand, as $x \to -\infty$, the two functions could deviate at most by 2ξ, where ξ is the distance between x and γ. Hence, the upper bound is $x + 2\xi + b$. Combining both bounds yields the desired result. The bound works for any γ. $\square$

Lemma 3.1 allows us to lower bound the log-likelihood function as follows.

Theorem 3.2.

$$\begin{aligned}
\log \Pr(\mathbf{V} = \mathbf{v}|\mathbf{w}) \quad \geq \quad & -\min_{\xi} (\sum_i 2\xi_i + Kb + R(\mathbf{w})) \\
\textit{subject to} \quad & \xi_i \geq 0 \\
& \sum_{j:V_i \in C_j} w_j f_j \geq \gamma - \xi_i \\
& R(\mathbf{w}) < K \log 2 + 4\|\mathbf{w}\|_1
\end{aligned}$$

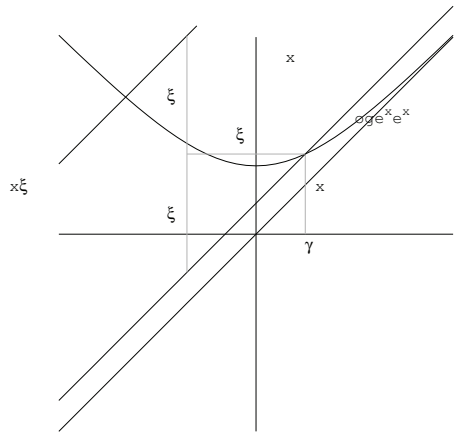

Figure 1: Illustration of upper bounds on $\log(e^x + e^{-x})$

Proof. We upper bound the partition function as follows:

$$
\begin{aligned}
&\sum_{\mathbf{V}} e^{\sum_j w_j f_j} \\
=& \sum_{\mathbf{V}\setminus V_1} \Big(\sum_{V_1=\{-v_1,v_1\}} e^{\sum_{j:V_1\in C_j} w_j f_j} \Big) e^{\sum_{j:V_1\notin C_j} w_j f_j} \\
=& \sum_{\mathbf{V}\setminus V_1} \big(e^{\sum_{j:V_1\in C_j} w_j f_j} + e^{-\sum_{j:V_1\in C_j} w_j f_j}\big) e^{\sum_{j:V_1\notin C_j} w_j f_j} \\
\le& \sum_{\mathbf{V}\setminus V_1} e^{\sum_{j:V_1\in C_j} w_j f_j + 2\xi_1 + b} e^{\sum_{j:V_1\notin C_j} w_j f_j} \\
=& \sum_{\mathbf{V}\setminus V_1} e^{\sum_j w_j f_j + 2\xi_1 + b} \\
=& \sum_{\mathbf{V}\setminus\{V_1,V_2\}} \Big(\sum_{V_2} e^{\sum_{j:V_2\in C_j} w_j f_j + 2\xi_1 + b}\Big) e^{\sum_{j:V_2\notin C_j} w_j f_j} \\
\le& \sum_{\mathbf{V}\setminus\{V_1,V_2\}} \Big(\sum_{V_2} e^{\sum_{j:V_2\in C_j} w_j f_j}\Big)\Big(\sum_{V_2} e^{2\xi_1+b}\Big) e^{\sum_{j:V_2\notin C_j} w_j f_j} \\
\le& \sum_{\mathbf{V}\setminus\{V_1,V_2\}} e^{\sum_j w_j f_j + 2\xi_2 + b} \Big(\sum_{V_2} e^{2\xi_1+b}\Big)
\end{aligned}
$$

In order to proceed further, we need to bound $\sum_{V_2} e^{2\xi_1+b}$ since ξ_1 depends on V_2. Let $[\,\cdot\,] = \max(0,\cdot)$ be the rectification function.

$$
\begin{aligned}
&\sum_{V_2=\{-v_2,v_2\}} e^{2\xi_1+b} \le \\
&e^b \big(e^{2[\gamma - \sum_{j:V_1,V_2\in C_j} w_j f_j - \sum_{j:V_1\in C_j} w_j f_j]_+} + \\
&e^{2[\gamma - \sum_{j:V_1,V_2\in C_j} w_j f_j - \sum_{j:V_1\in C_j} w_j f_j + 2\sum_{j:V_1,V_2\in C_j} w_j f_j]_+}\big)
\end{aligned}
$$

Since $[\,\cdot\,]_+$ is 1-Lipschitz, and $|f_j| \le 1$

$$
\sum_{V_2=\{-v_2,v_2\}} e^{2\xi_1+b} \le e^{2\xi_1+b} 2e^{4g\sum_{j:V_1,V_2\in C_j}|w_j|}
$$

We substitute the above result back, and continue the same approximation over all the other V_i to get the desired bound. □

Adding an L_2 regularization term on $\mathbf{w}$, with a given set of samples of size N, we can train the BM by solving the following optimization problem:

Optimization Problem 3.3.

$$
\begin{aligned}
\min_{\mathbf{w},\xi} \quad & \frac{1}{N}\sum_{i,l}\xi_{il} + \lambda(\eta_0 R(\mathbf{w}) + \frac{1}{2}\|\mathbf{w}\|^2) \\
\textit{subject to} \quad & \sum_{j:V_i\in C_j} w_j f_{jl} \ge \gamma - \xi_{il} \\
& \xi_{il} \ge 0
\end{aligned}
$$

Here $R(\mathbf{w})$ can be considered as an extra regularization of $\mathbf{w}$, and its weight η_0 will be selected according to cross validation. We will justify its importance to generalization performance in Section 4. After training, we predict values of the random variables $\mathbf{V}$ by solving:

Optimization Problem 3.4.

$$
\begin{aligned}
\hat{\mathbf{v}} = \arg\min_{\mathbf{v},\xi} \quad & \sum_i \xi_i \\
\textit{subject to} \quad & \sum_{j:V_i\in C_j} w_j f_j \ge \gamma - \xi_i \\
& \xi_i \ge 0
\end{aligned}
$$

In the case of conditional BMs where $\mathbf{V} = (\mathbf{X}, \mathbf{Y})$, the objective functions above only involve hinge loss terms for variables $\mathbf{Y}$ and ignore the loss terms for variables $\mathbf{X}$. The extra penalty term $R(\mathbf{w})$ only contains the weights of those cliques that involve at least two Y vertices. In addition, the partition function of the conditional likelihood does not marginalize over $\mathbf{X}$, and the domain of $\mathbf{X}$ can be relaxed to real values. In the rest of this paper, we discuss the conditional BMs for classification tasks, although the algorithms apply to the original BMs as well.

3.1 Relation to other large margin based structural learning methods

Large margin based approaches have been explored extensively in the past few years [Crammer and Singer, 2001; Collins, 2002; Taskar *et al.*, 2004; Tsochantaridis *et al.*, 2004]. The basic idea is that the objective function $\sum_j w_j f_j$ with true assignments of $\mathbf{Y}$ must be greater than ones with other assignments by a margin γ. The *Structural Support Vector Machine* (SSVM) is one approach to this problem that can be solved efficiently [Tsochantaridis *et al.*, 2004; Joachims *et al.*, to appear] by considering only a polynomial number of constraints to achieve optimality. For each added constraint, one has to perform an inference (the "separation oracle"). When exact inference is not available, either performance or correctness is not guaranteed [Finley and Joachims, 2008; Kulesza and Pereira, 2007].

In contrast, the LMBM does not need any inference during training and leads to a fast learning algorithm. Furthermore, we show that the LMBM has a generalization bound with a sample complexity as low as the PAC-Bayes bound of SSVMs.

4 Empirical Risk Minimization and Generalization Bound

The training of SSVMs minimizes the prediction errors empirically, which is considered an advantage over *maximum likelihood estimation*. Similarly, the objective function of the LMBM also upper bounds the 0-1 error as shown in Theorem 4.1 below. Therefore, the performance of the LMBM can be justified by *empirical risk minimization* as well.

Theorem 4.1. *Consider the decision function defined in O.P 3.4, and let* $L(\mathbf{x},\mathbf{y},\mathbf{w}) = \sum_i \xi_i(\mathbf{x},\mathbf{y},\mathbf{w})$. *If* $\forall \mathbf{y} \neq \hat{\mathbf{y}}, \mathbf{x},\mathbf{y} \sim \mathcal{D}$, $\exists T > 0$, $L(\mathbf{x},\mathbf{y},\mathbf{w}) > T$, *then*

$$\Pr(\hat{\mathbf{y}} \neq \mathbf{y}|\mathbf{w}) \leq \frac{1}{T}\mathbb{E}_{\mathcal{D}}[L(\mathbf{x},\mathbf{y},\mathbf{w})] \tag{2}$$

Proof.

$$\begin{aligned}\Pr(\hat{\mathbf{y}} \neq \mathbf{y}|\mathbf{w}) &= \mathbb{E}_{\mathcal{D}}[\mathbf{1}(\hat{\mathbf{y}} \neq \mathbf{y})] \\ &\leq \mathbb{E}_{\mathcal{D}}[\mathcal{H}(L(\mathbf{x},\mathbf{y},\mathbf{w}) - T)] \\ &\leq \frac{1}{T}\mathbb{E}_{\mathcal{D}}[L(\mathbf{x},\mathbf{y},\mathbf{w})]\end{aligned}$$

The function $\mathbf{1}(z)$ is the indicator function that is 1 when z is true and 0 for false. $\mathcal{H}(z)$ is the Heaviside function that is 1 for $z \geq 0$ and 0 otherwise. The last inequality comes from the fact that $\mathcal{H}(v - T) \leq \frac{v}{T}$. □

In above theorem, the threshold T is a crucial factor when the expected loss $\mathbb{E}_{\mathcal{D}}[L(\mathbf{x},\mathbf{y},\mathbf{w})]$ is greater than 0 due to the existence of noise. The larger the T, the better the generalization performance. The threshold T for the LMBM is given below:

Remark 4.2. Let $S_i = \{j|\exists k \neq i, Y_i, Y_k \in C_j\}$, $g_j = \sup_{\mathbf{x},\mathbf{y}} |f_j(\mathbf{x},\mathbf{y})|$, we have $T = \min_i[\gamma - \sum_{j\in S_i} |w_j|\, g_j]_+$

Proof. Let $f_j = f_j(\mathbf{x},\mathbf{y}), \hat{f}_j = f_j(\mathbf{x},\hat{\mathbf{y}})$. For any $y_i \neq \hat{y_i}$, we have $\xi_i = [\gamma - A_0 - A_1]_+$,$\hat{\xi}_i = [\gamma - A_0 - A_2]_+$ where $A_0 = \sum_{j:Y_i\in C_j \wedge f_j = \hat{f}_j} w_j f_j, A_1 = \sum_{j:Y_i\in C_j \wedge f_j \neq \hat{f}_j} w_j f_j, A_2 = \sum_{j:Y_i\in C_j \wedge f_j \neq \hat{f}_j} w_j \hat{f}_j$. Since $\mathbf{x}$ is the same, and only $\mathbf{y} \in \{-1,1\}^K$ changed, it is easy to verify $f_j = -\hat{f}_j$, if $f_j \neq \hat{f}_j$. So $\hat{\xi}_i = [\gamma - A_0 + A_1]_+$.
If $A_1 < 0$, we have $L > \xi_i > [\gamma - A_0]_+$. Otherwise, $L > \hat{L} > \hat{\xi}_i > [\gamma - A_0]_+$. So $L > [\gamma - \sum_{j:Y_i\in C_j \wedge f_j = \hat{f}_j} w_j f_j]_+$. To make the threshold independent of $\mathbf{x},\mathbf{y}$, we further loosen it to $L > \min_i[\gamma - \sum_{j\in S_i} |w_j|\, g_j]_+$. □

If the weights of those cliques that involve at least two Y variables are too large, the threshold is possibly very small, and generalization performance will not be guaranteed. In fact, the extra regularization term $R(\mathbf{w})$ in O.P. 3.3 can keep those weights small to ensure a large threshold. In the experiments, we show that the LMBM with this extra regularization consistently outperforms the one without it. Furthermore, although $R(\mathbf{w})$ contains a L_1 term, for the algorithmic simplicity, we replace it with a L_2 term.

The convergence rate of a stochastic objective function similar to LMBM's has been studied by Shalev-Shwartz et al. [Shalev-Shwartz *et al.*, 2008], where only one output variable is considered. With Lemma 4.3, Corollary 4 in [Shalev-Shwartz *et al.*, 2008] can be applied to the objective function of LMBM.

Lemma 4.3. *Let* $\mathcal{F} = \{\mathbf{x},\mathbf{y} \mapsto L(\mathbf{x},\mathbf{y},\mathbf{w})\}$, $\mathcal{F}_i = \{\mathbf{x},\mathbf{y} \mapsto \sum_{j:V_i\in C_j} w_j f_j(\mathbf{x},\mathbf{y})\}$, $\phi(z) = [\gamma - z]_+$. *We have*

$$\mathbb{E}\left[\sup_{h\in\mathcal{F}}\left(\mathbb{E}h - \hat{\mathbb{E}}_N h\right)\right] \leq \sum_i \mathcal{R}_N(\phi \circ \mathcal{F}_i)$$

Proof.

$$\begin{aligned}&\mathbb{E}\left[\sup_{h\in\mathcal{F}}\left(\mathbb{E}h - \hat{\mathbb{E}}_N h\right)\right] \\ \leq\ &\mathbb{E}\left[\sup_{h\in\mathcal{F}}\frac{1}{N}\sum_l \left(h(\mathbf{x}'_l,\mathbf{y}'_l) - h(\mathbf{x}_l,\mathbf{y}_l)\right)\right] \\ \leq\ &\mathbb{E}\left[\sum_i \sup_{h'_i\in\phi\circ\mathcal{F}_i}\frac{1}{N}\sum_l \left(h'_i(\mathbf{x}'_l,\mathbf{y}'_l) - h'_i(\mathbf{x}_l,\mathbf{y}_l)\right)\right] \\ =\ &\sum_i \mathcal{R}_N(\phi\circ\mathcal{F}_i)\end{aligned}$$

Here $\mathcal{R}_N$ is the *Rademacher complexity* [Bartlett and Mendelson, 2002] of sample size N. See [Bartlett and Mendelson, 2002] for details on the notation. □

We can now derive a generalization bound as in Theorem 4.4 below by first defining:

$$\begin{aligned}\mathcal{L}(\mathbf{w}) &= \mathbb{E}_{\mathcal{D}}[L(\mathbf{x},\mathbf{y},\mathbf{w})] \\ \hat{\mathbf{w}} &= \arg\min_{\mathbf{w}}\left(\frac{1}{N}\sum_l L(\mathbf{x}_l,\mathbf{y}_l,\mathbf{w}) + \lambda\|\mathbf{w}\|_2^2\right) \\ \mathbf{w}_o &= \arg\min_{\mathbf{w}}\mathcal{L}(\mathbf{w})\end{aligned}$$

Theorem 4.4. *Assuming* $\sum_j \|f_j\|_2^2 < B^2$, d *is the maximum clique size, for any* $\delta > 0$, *with probability* $1-\delta$ *over the sample size* N, *if* $\lambda = c\frac{B\sqrt{d/\delta}}{\|\mathbf{w}_o\|\sqrt{N}}$, *where* c *is a constant,we have*

$$\begin{aligned}&\Pr(\hat{\mathbf{y}} \neq \mathbf{y}|\hat{\mathbf{w}}) \leq \frac{1}{T}\mathcal{L}(\hat{\mathbf{w}}) \\ \leq\ &\frac{1}{T}\left(\mathcal{L}(\mathbf{w}_o) + O\left(\sqrt{\frac{B^2 d\,\|\mathbf{w}_o\|^2\log(1/\delta)}{N}}\right)\right)\end{aligned}$$

This generalization bound is better than the PAC-Bayes bound for SSVMs by $\sqrt{\log N}$. When the exact inference is intractable, the generalization bound for SSVMs is further worsened due to the extra degrees of freedom introduced by relaxation [Kulesza and Pereira, 2007].

5 Dual Coordinate Descent Method for Large Margin Boltzmann Machines

In this section, we propose a dual coordinate descent method for large margin BMs that shares similarities with the dual

coordinate method used for linear SVMs [Hsieh *et al.*, 2008]. The method contains a fast update for each iteration, with a relatively small number of iterations. Unlike nonlinear SVMs, one does not need to re-compute the whole gradient in each iteration, which saves a factor of $O(N)$ time where N is the number of samples.

Consider the optimization problem

$$\begin{aligned} \min_{\mathbf{w},\xi} \quad & \frac{1}{2}\sum_j \eta_j \mathbf{w}_j^2 + U\sum_{il}\xi_{il} \\ \text{subject to} \quad & \sum_{j:V_i\in C_j} w_j f_{jl} \geq \gamma - \xi_{il} \\ & \xi_{il} \geq 0 \end{aligned}$$

where $U = 1/(\lambda N)$ and η are the penalty constants set by cross-validation. If the j^{th} clique contains only one Y variable, $\eta_j = 1$, otherwise $\eta_j = \eta_0$. Let α_{il} and β_{il} be Lagrange multipliers. Then, we have the Lagrangian:

$$\begin{aligned} \mathcal{L}(\mathbf{w},\xi,\alpha,\beta) &= \frac{1}{2}\sum_j \eta_j \mathbf{w}_j^2 + U\sum_{il}\xi_{il} - \sum_{il}\beta_{il}\xi_{il} - \\ & \sum_{il}\alpha_{il}(\sum_{j:V_i\in C_j} w_j f_{jl} - \gamma + \xi_{il}) \end{aligned}$$

We optimize $\mathcal{L}$ with respect to $\mathbf{w}$ and ξ:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial w_j} &= \eta_j w_j - \sum_l \sum_{i:V_i\in C_j} \alpha_{il} f_{jl} = 0 \\ \frac{\partial \mathcal{L}}{\partial \xi_{il}} &= U - \alpha_{il} - \beta_{il} = 0 \end{aligned}$$

Substituting for $\mathbf{w}$ and ξ, we have the dual Lagrangian:

$$\mathcal{L}_\alpha = \frac{1}{2}\sum_{j,l,l'}\sum_{i:V_i\in C_j}\sum_{i':V_{i'}\in C_j} \alpha_{il}\alpha_{i'l'}\frac{f_{jl}f_{jl'}}{\eta_j^2} - \gamma\sum_{il}\alpha_{il}$$

The dual coordinate descent algorithm picks α_{il} one at a time and optimizes the dual Lagrangian with respect to this variable:

$$\begin{aligned} \min_d \quad & \mathcal{L}_\alpha(d) = \frac{1}{2}\sum_{j:V_i\in C_j} Q_{jll}d^2 - \nabla_{il}\mathcal{L}_\alpha d + const. \\ \text{subject to} \quad & 0 \leq \alpha_{il} + d \leq U \end{aligned}$$

where $Q_{jll} = \frac{f_{jl}f_{jl}}{\eta_j^2}$, $\nabla_{il}\mathcal{L}_\alpha = \sum_{j:V_i\in C_j} w_j f_{jl} - \gamma$. The objective has the almost the same form as linear SVMs. The Q matrix satisfies all properties required for convergence described in [Hsieh *et al.*, 2008].

6 Experimental Results

We applied LMBMs to the challenging problem of multi-label classification. We choose four popular benchmark datasets: **Scene** [Boutell *et al.*, 2004], **Yeast** [Elisseeff and Weston, 2002], and **Mediamill10/Mediamill50** [Snoek *et al.*, 2006]. **Mediamill10** selects the first 10 labels and data containing at least one of them. **Mediamill50** selects the first 50 labels. The basic characteristics of these datasets are listed in Table 1.

Algorithm 1 The dual coordinate descent algorithm for large margin Boltzmann machines

$\alpha \leftarrow 0, \mathbf{w} \leftarrow 0$
while α is not optimal **do**
  for all α_{il} **do**
    $\alpha_o \leftarrow \alpha_{il}$
    $G = \sum_{j:V_i\in C_j} w_j f_{jl} - \gamma$
    $PG = \begin{cases} \min(G,0) & \alpha_o = 0, \\ \max(G,0) & \alpha_o = U, \\ G & 0 < \alpha_o < U \end{cases}$
    if $|PG| \neq 0$ **then**
      $\alpha_{il} \leftarrow \min(\max(\alpha_o - \frac{G}{\sum_{j:V_i\in C_j} Q_{jll}}, 0), U)$
      $w_j \leftarrow w_j + (\alpha_{il} - \alpha_o) f_{jl}$, if $V_i \in C_j$
    end if
  end for
end while
return $\mathbf{w}$

Table 1: Characteristics of the datasets used in the experiments. The table shows the number of training and testing examples, number of features, number of labels, and number of weights for each dataset.

name	train	test	features	labels	weights
scene	1211	1196	294	6	1788
yeast	1500	917	103	14	1547
mediamill10	2718	1087	120	10	1255
mediamill50	25737	10919	120	50	7500

LMBM and SSVM model the same distribution family, i.e., a completely connected graph with a maximal clique size of two. The approximation algorithm for SSVM implements relaxed linear programming (LP) and is guaranteed to reach a possibly sub-optimal solution in polynomial time [Finley and Joachims, 2008]. In our experiments involving SSVM, we adopt a popular LP relaxation that enforces a *marginalization* constraint [Wainwright *et al.*, 2005a] and solve it with the software package *lp_solve*. We optimize the margin-rescaling objective with the Hamming loss. For LMBMs, we implement a mixed integer programming (MIP) algorithm for testing and cross-validation. We observe that for the optimization problem 3.4, MIP always outperforms LP, but interestingly, for SSVMs, integer programming produces the same results as LP on these datasets.

Five methods are compared in these experiments: LMBM trained without the extra regularization $\eta_0 = 0$, LMBMr trained with the extra regularization $\eta_0 \neq 0$, SSVM the structural SVM, ILSVM the independent linear SVM that treats $\mathbf{Y}$ mutually independent, and MLSVM which transforms the multi-label classification problem into a multi-class classification problem by treating each label combination as one class. According to [Tsoumakas and Katakis, 2007], in **Yeast** and **Scene**, MLSVM outperforms many other methods, but it is not scalable to large numbers of Y since the number of classes increases exponentially.

In multi-label datasets, the labels are typically highly unbalanced, with negative labels usually outnumbering positive

labels. So the Hamming loss (H) is not an informative performance metric. We adopt four additional popular performance metrics: accuracy (A), precision (P), recall (R) and F1 score (F) as defined in Eqn. 3 below, where $\hat{\mathbf{Y}}$ are the predicted positive labels and $\mathbf{Y}$ are the true positive labels.

$$A = \frac{|\mathbf{Y} \cap \hat{\mathbf{Y}}|}{|\mathbf{Y} \cup \hat{\mathbf{Y}}|}, P = \frac{|\mathbf{Y} \cap \hat{\mathbf{Y}}|}{|\hat{\mathbf{Y}}|}, R = \frac{|\mathbf{Y} \cap \hat{\mathbf{Y}}|}{|\mathbf{Y}|}, F = \frac{2PR}{P+R} \tag{3}$$

For ILSVM, LMBM, LMBMr and SSVM, 5-fold cross-validation was used to select the parameters individually. For SSVM, C was selected from $\{0.01, 0.1, 1, 10, 100\}$. For LMBMr, LMBM and ILSVM, U was selected from $\{0.01, 0.1, 1, 10, 100\}$. For LMBMr, η_0 was selected from $\{5, 10, 100\}$.

The results of MLSVM come from [Tsoumakas and Katakis, 2007]. The metric used in cross-validation is the accuracy (A). We also recorded the training time for each machine trained with the selected optimal parameters. The CPU time is measured on a 2.8Ghz Pentium4 desktop computer.

The final results are shown in Table 2 ($\downarrow$ means smaller numbers are better while $\uparrow$ means the opposite).

A key observation from Table 2 is that LMBMs are consistently orders of magnitude faster than SSVMs and the other algorithms, while producing results that are in the same range as the other methods. The speed of SSVMs depends on the speed of the *lp_solve* and the number of times *lp_solve* is called. The actual number of variables in the relaxed LP is quadratic of the number of labels in the problem. Typically, *lp_solve* solves a 10-variable problem in about 0.01s, and a 50-variable problem in about 2s. In our experiments, SSVMs fails to produce results for **Mediamill50** in 10 hours. For smaller scale problems such as **Scene** and **Yeast**, MLSVM can outperform LMBMr but MLSVM becomes exponentially expensive as the number of labels increases, and the sample complexity increases exponentially as well. ILSVM performs well on **Mediamill10** and **Mediamill50** since the labels are relatively independent of each other, but the performance drops when there are dependencies among labels as in **Scene**. Both LMBM and SSVM perform well in datasets with strong label-dependency, but not in datasets with weak label-dependency. LMBMr demonstrates good performance on both types of datasets, which might indicate that the threshold T is the key factor governing generalization performance.

More importantly, LMBMr is able to model the dependency structure among the variables that leads to improved accuracy (A) of prediction compared with ILSVM. Figure 2 shows the different dependency patterns discovered by LMBMr for the different datasets. Interestingly, the dependency pattern changes significantly from small scale problems to large scale problems. For example, in **Mediamill10**, the label 4 (outdoor) has a strong correlation (negative) with the label 1 (people) and a weak correlation (positive) with label 10 (crowd). But in **Mediamill50**, the correlation between people and outdoor disappears, while the weak correlation between crowd and outdoor remains. Another interesting characteristic of the **Mediamill50** dataset is that unlike the other three datasets, the most frequent labels (upper left portion of the weights matrix) appear relatively independent of each other, while the least frequent labels (lower right portion of the weights matrix) tend to have stronger correlations with the others.

Table 2: Performance results on the four datasets.

Scene						
Method	time(s)↓	H↓	A↑	P↑	R↑	F↑
ILSVM	0.63	0.109	0.567	0.782	0.605	0.682
LMBM	0.25	0.108	0.681	0.721	0.686	0.703
LMBMr	**0.24**	0.107	0.682	0.722	0.687	0.704
SSVM	81.8	**0.100**	0.619	**0.795**	0.635	0.706
MLSVM	N.A.	**0.100**	**0.704**	0.713	**0.737**	**0.725**
Yeast						
Method	time(s)↓	H↓	A↑	P↑	R↑	F↑
ILSVM	0.95	**0.199**	0.498	0.717	0.570	0.635
LMBM	0.47	0.368	0.329	0.414	0.561	0.476
LMBMr	**0.36**	**0.199**	0.504	0.709	0.584	0.640
SSVM	81.7	0.233	0.337	**0.748**	0.337	0.465
MLSVM	N.A.	0.206	**0.530**	0.615	**0.672**	**0.642**
Mediamill10						
Method	time(s)↓	H↓	A↑	P↑	R↑	F↑
ILSVM	75.57	**0.049**	0.636	**0.859**	0.636	0.731
LMBM	**36.36**	0.082	0.603	0.605	0.603	0.604
LMBMr	45.2	0.052	**0.731**	0.752	**0.731**	**0.741**
SSVM	459.9	0.082	0.589	0.609	0.589	0.599
Mediamill50						
Method	time(s)↓	H↓	A↑	P↑	R↑	F↑
ILSVM	**32.29**	**0.025**	**0.480**	0.739	**0.480**	0.582
LMBM	49.53	0.225	0.044	0.048	0.286	0.082
LMBMr	33.00	**0.025**	**0.480**	**0.741**	**0.480**	**0.583**
SSVM	$>$ 10hrs					

7 Conclusions and Future work

We have proposed an efficient SVM-style learning algorithm for Boltzmann machines based on a new large-margin variational approximation to the partition function. For multi-label classification tasks, these large margin Boltzmann machines provide an efficient alternative to recently proposed structural SVMs. Experimental results on four benchmark multi-label classification problems demonstrate that LMBMs are significantly faster than existing approaches and can match or beat the performance of other methods.

Several challenges and open questions remain. For example, can a better approximation be achieved through the use of hidden variables in large margin Boltzmann machines? Although the learning algorithm for LMBMs is fast, the inference is still slow and approximate. To address this problem, we have recently explored large margin *directed* Boltzmann machines (LMDBMs) that can again be trained using an SVM-style objective function [Miao and Rao, 2009]. More importantly, for LMDBMs, there exists a *branch and bound* exact inference algorithm that is many orders of magnitude faster than LP.

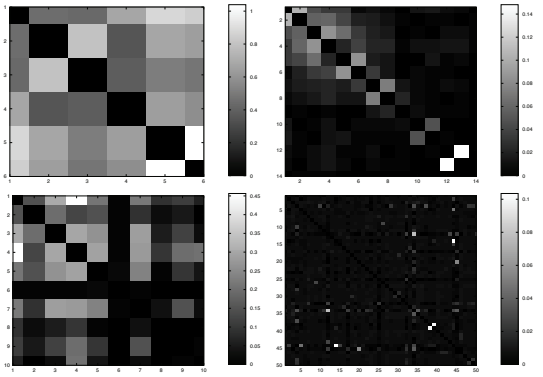

Figure 2: The different dependency structures discovered by LMBMr for different datasets. The colorbar of each figure shows the scale of weights in terms of the gray level(absolute value). The top row, left to right: **Scene** and **Yeast**; The bottom row, left to right: **Mediamill10** and **Mediamill50**.

Acknowledgments

We would like to thank Andrew Guillory, James Lee, Aaron Shon, and the anonymous reviewers for valuable suggestions.

References

[Bartlett and Mendelson, 2002] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[Boutell *et al.*, 2004] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37:1757–1771, 2004.

[Collins, 2002] Michael Collins. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *EMNLP*, pages 1–8, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[Crammer and Singer, 2001] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[Elisseeff and Weston, 2002] Andr Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *In Advances in Neural Information Processing Systems 14*, 2002.

[Finley and Joachims, 2008] Thomas Finley and Thorsten Joachims. Training structural svms when exact inference is intractable. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 304–311, New York, NY, USA, 2008. ACM.

[Geman and Geman, 1984] 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, 1984.

[Hinton and Sejnowski, 1983] G. E. Hinton and T. J. Sejnowski. Optimal perceptual inference. In *CVPR, Washington DC*, pages 448–53, 1983.

[Hsieh *et al.*, 2008] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 408–415, New York, NY, USA, 2008. ACM.

[Joachims *et al.*, to appear] T. Joachims, T. Finley, and Chun-Nam Yu. Cutting-plane training of structural svms. *Machine Learning*, to appear.

[Kulesza and Pereira, 2007] Alex Kulesza and Fernando Pereira. Structured learning with approximate inference. In *In Advances in Neural Information Processing Systems*, 2007.

[Lafferty *et al.*, 2001] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, 2001.

[Miao and Rao, 2009] Xu Miao and Rajesh P. R. Rao. Large margin directed boltzmann machines for learning in structured output spaces. Technical report, University of Washington, 2009.

[Neal, 2003] Radford M. Neal. Slice sampling. *The Annals of Statistics*, 31:705–767, 2003.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[Shalev-Shwartz *et al.*, 2008] Shai Shalev-Shwartz, Nathan Srebro, and Karthik Sridharan. Fast rates for regularized objectives. In *NIPS*, 2008.

[Snoek *et al.*, 2006] Cees G.M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W.M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *In Proceedings of ACM Multimedia*, pages 421–430, Santa Barbara, USA, 2006.

[Sudderth *et al.*, 2007] E. B. Sudderth, M. J. Wainwright, and A. S. Willsky. Loop series and bethe variational bounds in attractive graphical models. In *Proceedings of the NIPS conference*, December 2007.

[Taskar *et al.*, 2004] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Advances in Neural Information Processing Systems (NIPS 2003)*, Vancouver, Canada, 2004.

[Tsochantaridis *et al.*, 2004] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, page 104, 2004.

[Tsoumakas and Katakis, 2007] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3:1–13, 2007.

[Wainwright *et al.*, 2005a] M. Wainwright, T. Jaakola, and A. Willsky. Map estimation via agreement on trees: Message passing and linear programming. *IEEE Trans. on Information Theory*, 51:3697–3717, 2005.

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

[Yedidia *et al.*, 2005] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Trans. IT*, 51:2282–2312, July 2005.