Maximum Margin Bayesian Networks

Yuhong Guo* **Dana Wilkinson †** **Linli Xu†** **Dale Schuurmans***

*University of Alberta
†University of Waterloo

Abstract

We consider the problem of learning Bayesian network classifiers that maximize the margin over a set of classification variables. We find that this problem is harder for Bayesian networks than for undirected graphical models like maximum margin Markov networks (M^3N), since the parameters in a Bayesian network must satisfy additional normalization constraints that an undirected graphical model need not respect. Unfortunately, these normalization constraints eliminate the convexity properties of the training problem, and significantly complicate the optimization task. Nevertheless, we derive an effective training algorithm that solves the maximum margin training problem for a range of network topologies, and otherwise converges to a locally optimal set of parameters for arbitrary network topologies. Experimental results show that the method can demonstrate improved generalization performance when the directed graphical structure encodes relevant knowledge. Our intent is to pose and investigate what we believe is a natural machine learning approach, while also pointing out its difficulties.

1 Introduction

When training probability models for classification tasks it is often recommended that the model parameters be optimized under a discriminative training criterion such as conditional likelihood [5]. However, general Bayesian network classifiers have rarely, if ever, been trained to maximize the margin—arguably the most discriminative criterion available. Recently it has been observed that undirected graphical models can be efficiently trained to maximize the margin, even simultaneously over a set of classification variables [10]. (An interesting precursor is [2].) However, these training algorithms have adopted the Euclidean normalization constraint of support vector machines (SVMs), which can be accommodated in their frameworks because they rely on an undirected graphical model representation which allows a single arbitrary normalization.

In this paper we consider applying the maximum margin methodology to Bayesian networks; that is, directed graphical models. Unlike Markov network models, Bayesian networks require that additional normalization constraints be satisfied; namely that the local clique potentials represent conditional probability distributions. These constraints are very different from the standard Euclidean (or L_p) normalization constraints of SVMs. Nevertheless, they do not preclude the possibility of learning large margin classifiers. In fact, our goal is simply to exploit the benefits of learning maximum margin classifiers, while still being able to represent the learned classifier as a Bayesian network.

There are several motivations for attempting to maintain a Bayesian network representation. First, the classification model being learned could be a fragment of a much larger probabilistic causal model. In this case, maintaining a Bayesian network representation could allow one to integrate the learned model with a preexisting background model without additional effort. Second, the normalization constraints asserted by a directed graphical structure capture nonparametric *causal* knowledge about the domain. Therefore, respecting these constraints allows one to exploit the advantages of Bayesian networks for capturing intuitive causal structure. Note that removing the normalization constraints would turn the Bayesian network into a Markov network, and this would necessarily remove the causal knowledge that was originally encoded by the local normalization constraints.

To understand both the prospects and limitations of learning maximum margin Bayesian network classifiers we proceed as follows. First, after preliminary definitions in Section 2, we investigate the notion of *classification margin* for Bayesian network classifiers in Section 3, and relate this to the common conditional likelihood criterion of graphical models. We then derive an effective training algorithm in Section 4 that solves a wide range of problems exactly, and otherwise provides an effective heuristic for finding local solutions. In Section 5 we present experimental results which show that the causal information in Bayesian networks can yield effective generalization performance when the directed graphical structure captures relevant causal knowledge. Finally, we extend the approach to multivariate classification in Section 6.

In the end, we also observe a few drawbacks of large margin Bayesian networks, thus the main message of this paper is necessarily mixed: On one hand, maximum margin Bayesian networks allow one to exploit prior causal knowledge to obtain generalization improvements in many cases. On the other hand they create additional computational difficulty. Nevertheless, maximum margin Bayesian networks are a natural combination of two predominant current learning technologies, and we feel this combination is worth study.

2 Bayesian networks

We assume we are given a Bayesian network which is defined by a directed acyclic graph over variables $X_1, ..., X_n$ where the probability of a complete configuration $\mathbf{x}$ is given by

$$P(\mathbf{x}|\boldsymbol{\theta}) \quad = \quad \prod_{j=1}^{n} P(x_j|\mathbf{x}_{\pi(j)}) \quad = \quad \exp\left(\textstyle\sum_{ja\mathbf{b}} 1_{(\mathbf{x}_j = a\mathbf{b})} \ln \theta_{ja\mathbf{b}}\right) \tag{1}$$

Here $\boldsymbol{\theta}$ denotes the parameters of the model; j ranges over conditional probability tables (CPTs), one for each variable X_j; $1_{(\cdot)}$ denotes the indicator function; $\mathbf{x}_j$ denotes the local subconfiguration of $\mathbf{x}$ on $(x_j, \mathbf{x}_{\pi(j)})$; a denotes the set of values for child variable x_j; and $\mathbf{b}$ denotes the set of configurations for x_j's parents $\mathbf{x}_{\pi(j)}$. The form (1) shows how Bayesian networks are a form of exponential model

$$P(\mathbf{x}|\mathbf{w}) \quad = \quad \exp\left(\boldsymbol{\delta}(\mathbf{x})^\top \mathbf{w}\right) \tag{2}$$

using the substitution $w_{ja\mathbf{b}} = \ln \theta_{ja\mathbf{b}}$, where $\boldsymbol{\delta}(\mathbf{x})$ denotes the feature vector $(...1_{(\mathbf{x}_j = a\mathbf{b})}...)^\top$ over $j, a, \mathbf{b}$. The key aspect of the exponential form is that it expresses $p(\mathbf{x}|\mathbf{w})$ as a convex function of the parameters $\mathbf{w}$, which would seem to suggest convenient optimization problems. However, Bayesian networks also require the imposition of additional normalization constraints over each variable

$$\sum_a e^{w_{ja\mathbf{b}}} \quad = \quad 1 \text{ for all } j, \mathbf{b} \tag{3}$$

Unfortunately, these constraints are nonlinear, even though the objective is linear in $\mathbf{w}$. Removing these constraints greatly reduces the computational challenge of training, but also removes the causal interpretability of the model. Therefore, our goal in this paper is to maintain the Bayesian network constraints while investigating the consequences.

3 Discriminative training criteria

We initially assume there is a single classification variable Y taking on values $y \in \{1, .., K\}$. To make predictions, we will consider the maximum conditional probability prediction $\arg\max_y P(y|\mathbf{x})$. Note that for graphical models the conditional probability depends only on variables that share some common function (CPT) with Y (the Markov blanket of Y), and therefore we will restrict attention to this set of variables henceforth.

We are interested in learning the parameters for a Bayesian network classifier given training data of the form $(\mathbf{x}^1 y^1), ..., (\mathbf{x}^T y^T)$. Two standard training criteria to maximize during training are the joint loglikelihood and the conditional loglikelihood, given respectively by

$$\log L(\boldsymbol{\theta}) = \sum_{i=1}^{T} \log P(y^i|\mathbf{x}^i \boldsymbol{\theta}) + \log P(\mathbf{x}^i|\boldsymbol{\theta}) \tag{4}$$

$$\log CL(\boldsymbol{\theta}) = \sum_{i=1}^{T} \log P(y^i|\mathbf{x}^i \boldsymbol{\theta}) \tag{5}$$

Much of the literature suggests that the latter objective is better suited for classification [6, 5], although conditions have been identified where the former is advantageous [7, 8].

In this paper we investigate an alternative criterion based on the large margin criteria of SVMs. In particular, we adopt the multiclass margin definition of [4]. In our context, this objective can be cast maximizing the minimum conditional likelihood ratio (MCLR)

$$MCLR(\boldsymbol{\theta}) = \min_{i=1}^{T} \min_{y \neq y^i} \frac{P(y^i|\mathbf{x}^i \boldsymbol{\theta})}{P(y|\mathbf{x}^i \boldsymbol{\theta})}$$

$$\log MCLR(\boldsymbol{\theta}) = \min_{i=1}^{T} \min_{y \neq y^i} \log P(\mathbf{x}^i y^i|\boldsymbol{\theta}) - \log P(\mathbf{x}^i y|\boldsymbol{\theta}) \tag{6}$$

Thus our goal is to find a set of parameters $\boldsymbol{\theta}$ that maximizes the minimum margin between the target classification label against the best alternative under the probability model. (Note that we introduce slack variables to obtain a soft margin version of the criterion below.)

To see the connection to SVMs more clearly, note that one can substitute the exponential form of $P(y|\mathbf{x}\mathbf{w})$ into the MCLR objective, to obtain

$$\begin{aligned} \log MCLR(\mathbf{w}) &= \min_{i=1}^{T} \min_{y \neq y^i} \left[\boldsymbol{\delta}(\mathbf{x}^i y^i) - \boldsymbol{\delta}(\mathbf{x}^i y)\right]^\top \mathbf{w} \\ &= \min_{i=1}^{T} \min_{y \neq y^i} \boldsymbol{\Delta}(i, y)\, \mathbf{w} \end{aligned} \tag{7}$$

where $\boldsymbol{\Delta}(i, y) = [\boldsymbol{\delta}(\mathbf{x}^i y^i) - \boldsymbol{\delta}(\mathbf{x}^i y)]^\top$. Here the row vector $\boldsymbol{\Delta}(i, y)$ plays the role of the feature vector for training example i and class label y, and therefore we can write the entire set of feature vectors as a matrix Δ of size $(TK)\times$ (number of features).

Thus, starting with the training objective (6), through a change of parameters, we are led to a training problem that can be cast as a conventional maximum margin problem

$$\max_{\mathbf{w},\gamma} \gamma \quad \text{subject to} \quad \Delta\mathbf{w} \geq \gamma\boldsymbol{\xi}, \quad \|\mathbf{w}\|_2 = 1 \tag{8}$$

where $\xi_{(i,y)} = 1_{(y \neq y^i)}$. Notice that here we have added the normalization constraint $\|\mathbf{w}\|_2 = 1$. Obviously some form of normalization is necessary to avoid making $\Delta\mathbf{w}$ large in a trivial manner just by making $\mathbf{w}$ large. Euclidean normalization happens to yield a solution weight vector that maximizes the Euclidean margin [9]. The resulting constrained optimization problem corresponds to the standard version of multiclass SVMs proposed in [4] (ignoring slacks), here expressed over features determined by the Bayesian network.

In fact, this is the connection between probabilistic and large margin classifiers that is one of the main observations of [10, 2], who then proceed to use standard SVM training criteria over these features. Note however that the solution weight vector for (8) cannot be substituted into the Bayesian network representation, because it will not satisfy the proper normalization constraints (3). The previous techniques of [10, 2] were able to proceed by using an *undirected* graphical model which could accomodate unnormalized weights in the potential function. However, for Bayesian networks this is not sufficient, and there is usually no way to represent the same classifier in the original Bayesian network structure.

The alternative approach we consider, therefore, is to maximize the same objective, but subject to a different set of constraints that preserve representability as a Bayesian network

$$\max_{\mathbf{w},\gamma} \gamma \quad \text{subject to} \quad \Delta\mathbf{w} \geq \gamma\boldsymbol{\xi}, \quad \sum_a e^{w_{ja\mathbf{b}}} = 1 \quad \text{for all } j, \mathbf{b} \tag{9}$$

Unfortunately, these natural constraints on $\mathbf{w}$ are nonlinear and this yields a difficult optimization problem. Attempts to reformulate the problem according to standard transformations also fail. For example, the probability function (1) is neither concave nor convex in the parameters θ, even though the equality constraints are linear. The standard trick of removing the normalization constraints via the transformation $\theta_{ja\mathbf{b}} = e^{\omega_{ja\mathbf{b}}} / \sum_a e^{\omega_{ja\mathbf{b}}}$ also does not work in this case, since it creates terms of the form $\sum_{ja\mathbf{b}} \Delta_{(j,a,\mathbf{b})}(i,y)\left[\omega_{ja\mathbf{b}} - \log \sum_a e^{\omega_{ja\mathbf{b}}}\right]$ which are neither convex nor concave in $\boldsymbol{\omega}$. Thus, if we hope to solve the maximum margin Bayesian network training problem exactly, we require a more subtle approach.

4 A training algorithm

Although solving for the maximum margin Bayesian network parameters appears to be a hard problem, we can derive a practical training algorithm that still solves the problem for a wide range of graph topologies, and otherwise provides a useful foundation for heuristic approaches which seek local maxima. The main idea is to try to exploit convexity in the problem as much as possible, and identify situations where the solutions to a convex subproblem can be maintained.

Henceforth, we will actually consider the alternative soft margin version of the criterion

$$\max_{\mathbf{w},\gamma,\boldsymbol{\epsilon}} \gamma - C\boldsymbol{\epsilon}^\top \mathbf{e} \quad \text{subject to} \quad \Delta\mathbf{w} \geq \gamma\boldsymbol{\xi} - S\boldsymbol{\epsilon}, \ \ \boldsymbol{\epsilon} \geq 0, \ \ \sum_a e^{w_{ja\mathbf{b}}} = 1 \text{ for all } j, \mathbf{b} \tag{10}$$

where $\boldsymbol{\epsilon}$ are the slack variables; $\mathbf{e}$ denotes the vector of all 1 entries; S is a $TK \times T$ sparse matrix with nonzero entries $S((i,y),i) = 1$ (which simply enforces the constraint that $\boldsymbol{\Delta}(i,y)\mathbf{w} \geq \gamma\,\xi_{(i,y)} - \boldsymbol{\epsilon}_i$ for all i,y); and C is a parameter that controls the slack effect. Notice that the objective and first two sets of constraints are linear in the parameters $\mathbf{w}$, γ and $\boldsymbol{\epsilon}$. Unfortunately, the final constraints are nonlinear equalities on $\mathbf{w}$, which eliminates the convexity of the problem. However, the basic observation is that the problem can be made convex simply by relaxing these equality constraints to inequality constraints, thus yielding a simple relaxation of the problem which allows us to obtain a global solution

$$\min_{\mathbf{w},\gamma,\boldsymbol{\epsilon}} -\gamma + C\boldsymbol{\epsilon}^\top \mathbf{e} \quad \text{subject to} \quad \Delta\mathbf{w} \geq \gamma\boldsymbol{\xi} - S\boldsymbol{\epsilon}, \ \ \boldsymbol{\epsilon} \geq 0, \ \ \sum_a e^{w_{ja\mathbf{b}}} \leq 1 \text{ for all } j, \mathbf{b} \tag{11}$$

The solution to this problem will of course be subnormalized. The key fact about the relaxed optimization problem (11) however, is that it is convex in $\mathbf{w}$ and this will permit effective algorithmic approaches. For this problem we can obtain the Lagrangian

$$L_0(\mathbf{w},\gamma,\boldsymbol{\epsilon},\boldsymbol{\mu},\boldsymbol{\lambda}) \;\; = \;\; -\gamma + C\boldsymbol{\epsilon}^\top\mathbf{e} + \boldsymbol{\mu}^\top(\gamma\boldsymbol{\xi} - S\boldsymbol{\epsilon} - \Delta\mathbf{w}) + \sum_{j,\mathbf{b}} \lambda_{j\mathbf{b}} \left(\sum_a e^{w_{ja\mathbf{b}}} - 1 \right)$$

which gives us an equivalent problem to (11)

$$\min_{\mathbf{w},\gamma,\boldsymbol{\epsilon}} \max_{\boldsymbol{\mu},\boldsymbol{\lambda}} \; L_0(\mathbf{w},\gamma,\boldsymbol{\epsilon},\boldsymbol{\mu},\boldsymbol{\lambda}) \text{ subject to } \boldsymbol{\epsilon} \geq 0,\ \boldsymbol{\mu} \geq 0,\ \boldsymbol{\lambda} \geq 0 \tag{12}$$

First, it turns out to be easy to eliminate $\boldsymbol{\epsilon}$ and γ from (11), since $\partial L_0/\partial \boldsymbol{\epsilon} = C\mathbf{e} - S^\top \boldsymbol{\mu}$ and $\partial L_0/\partial \gamma = -1 + \boldsymbol{\mu}^\top \boldsymbol{\xi}$ respectively, and setting these to zero implies $S^\top \boldsymbol{\mu} = C\mathbf{e}$ and $\boldsymbol{\mu}^\top \boldsymbol{\xi} = 1$. If we enforce these constraints, the Lagrangian is reduced to

$$L_1(\mathbf{w},\boldsymbol{\mu},\boldsymbol{\lambda}) \quad = \quad -\boldsymbol{\mu}^\top \Delta \mathbf{w} + \sum_{j,\mathbf{b}} \lambda_{j\mathbf{b}} \left(\sum_a e^{w_{ja\mathbf{b}}} - 1 \right)$$

An equivalent optimization problem to (12) is therefore

$$\min_{\mathbf{w}} \max_{\boldsymbol{\mu},\boldsymbol{\lambda}} \; L_1(\mathbf{w},\boldsymbol{\mu},\boldsymbol{\lambda}) \quad \text{subject to} \quad S^\top \boldsymbol{\mu} = C\mathbf{e},\ \boldsymbol{\mu}^\top \boldsymbol{\xi} = 1,\ \boldsymbol{\mu} \geq 0,\ \boldsymbol{\lambda} \geq 0 \tag{13}$$

Because $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ are nonnegative, L_1 is convex in $\mathbf{w}$ and linear in $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$, and therefore this problem only has global solutions.

Unfortunately, one cannot easily eliminate the remaining primal variable $\mathbf{w}$ from this problem. Taking the partial derivative with repect to $w_{ja\mathbf{b}}$ yields

$$\frac{\partial L_1}{\partial w_{ja\mathbf{b}}} \quad = \quad \lambda_{j\mathbf{b}}\, e^{w_{ja\mathbf{b}}} \; - \; \boldsymbol{\mu}^\top \boldsymbol{\Delta}_{ja\mathbf{b}} \tag{14}$$

where $\boldsymbol{\Delta}_{ja\mathbf{b}}$ denotes the $ja\mathbf{b}$ column of Δ. The difficulty here is that one cannot set this derivative to zero because $\boldsymbol{\mu}^\top \boldsymbol{\Delta}_{ja\mathbf{b}}$ can be negative. Nevertheless, a global minimum can still be found in $\mathbf{w}$ by allowing $-\infty$ as a legal value for $w_{ja\mathbf{b}}$ (which corresponds to a probability of $\theta_{ja\mathbf{b}} = 0$). Despite the use of an extended value set, the problem remains convex and many forms of primal-dual search algorithms are still able to cope effectively with such problems [3]. We use a straightforward alternating gradient approach in our experiments below.

Of course, the solutions obtained to (11) may not be representable in a Bayesian network because the parameters $\mathbf{w}$ are subnormalized, not normalized. The main question that remains is determining when these subnormalized solutions can be converted into properly normalized Bayesian networks obeying the correct equality constraints (3). It turns out that a wide range of network topologies admit a simple procedure for renormalizing the local functions so that they become proper CPTs, without affecting the conditonal probability of y given $\mathbf{x}$. In fact, this observation has been previously made by [11, 12]. We present a simpler view here: It is easy to characterize when an unnormalized Bayesian network classifier can be renormalized to preserve $P(y|\mathbf{x})$. Consider an unnormalized local function $f(x,\mathbf{z})$ in a Bayesian network structure, and assume we want to normalize it over x. Note that this function can always be multiplied by a factor $\rho_{\mathbf{z}}$ for each $\mathbf{z}$, as long as there is another local function $f(\mathbf{z},\mathbf{q})$ that can be divided by the same factor. (I.e. a local function that contains *all* the parents $\mathbf{z}$ of x.) Thus, if an accompanying $f(\mathbf{z},\mathbf{q})$ always exists, we can always renormalize $f(x,\mathbf{z})$. Since the functions and variables follow an acyclic ordering in a Bayesian network, child variables can be sequentially renormalized bottom up without affecting previous normalizations. Finally, the factor containing the y variable can be renormalized to preserve $P(y|\mathbf{x})$.

The above renormalization strategy only fails if, at any stage, the parent variable set $\mathbf{z}$ is not contained in a single local function, but is instead split between separate local functions. In this case, there would be no way to coordinate the compensation for $\rho_{\mathbf{z}}$ (without adding a new local function over $\mathbf{z}$). Thus, in the end, we are left with an intuitive sufficient condition for when a Bayesian network can be renormalized: Any graph can be normalized without affecting $P(y|\mathbf{x})$ if the child variables can be eliminated without adding any new edges. In these cases, we can recover a normalized model without affecting the optimality of the solution to (11), and thus we obtain a global maximum of (7) with respect to (3).

5 Experimental results

To evaluate the utility of learning maximum margin Bayesian networks, we conducted some simple experiments on both real and synthetic data sets. In the synthetic experiments, we fixed a Bayesian network structure and parameters, and used it to generate training and test data. The goal of the synthetic experiments is to run a controlled comparison of maximum margin Bayesian and Markov networks, to determine the effects of having a correct Bayesian network structure. We experimented with several network topologies and parameterizations, and compared maximum margin Bayesian networks (M^2BN) trained according to (11) against maximum margin Markov networks (M^3N) trained according to (8) but with standard slack variables. The results are for 20 repetitions of the training sample.

The first synthetic experiments were conducted on the networks shown in Figure 1. Table 1 and Figure 2(a) shows that the techniques behave similarly, but show a systematic advantage for M^2BN over M^3N. This makes sense given that the data was generated from a Bayesian network with the same structure considered for training. The results show that the only technique which ignores the Bayesian network normalization constraints, M^3N, is slightly behind M^2BN which respects these constraints, vindicating somewhat the claim that the normalization structure of a directed causal model can impose an effective machine learning bias, beyond just providing the features for a generalized linear model.

For a more realistic comparison, we also experimented with real data from the UCI repository. In these cases, we formulated a Bayesian network topology that was intended to capture the causal structure of the domain, but in this case had no guarantee that the presumed structure was correct. These networks are much larger and cannot be easily visualized here. We sampled 5 disjoint training sets out of each data set, tested on the remainder, and report average results. Figure 2(b) shows the results. Interestingly, these results generally show an advantage for M^2BN over M^3N.

6 Multivariate extension

Finally, we consider the key extension of [10, 2] to multivariate classification. In this setting, we observe training data $(\mathbf{x}^1, \mathbf{y}^1), ..., (\mathbf{x}^T, \mathbf{y}^T)$ as before, but now the targets $\mathbf{y}^i$ are *vectors* of correlated classifications. Conceptually, this extension causes no change for M^2BN, and we can seek to maximize the criterion (11) as before. The only new challenge is coping with the exponential sum over $\mathbf{y}$. Fortunately, the derivation of our training algorithm in Section 4 is not significantly affected by this extension. Here we can take the same approach as [10] and reexpress our objective and constraints in terms of "marginal" Lagrange multipliers $\alpha_{(i,\mathbf{y})} = \sum_{\mathbf{y}\backslash\mathbf{y}_{ja\mathbf{b}}} \mu_{(i,\mathbf{y})}$, where $\mathbf{y}_{ja\mathbf{b}}$ denotes the subconfiguration of $\mathbf{y}$ that matches the local function j on pattern $a\mathbf{b}$. For example, the second term in the gradient (14) can be reduced to

$$\sum_{i,\mathbf{y}} \mu_{(i,\mathbf{y})} \boldsymbol{\Delta}_{ja\mathbf{b}}(i, \mathbf{y}_{ja\mathbf{b}}) = \sum_{i,\mathbf{y}_{ja\mathbf{b}}} \boldsymbol{\Delta}_{ja\mathbf{b}}(i, \mathbf{y}_{ja\mathbf{b}}) \sum_{\mathbf{y}\backslash\mathbf{y}_{ja\mathbf{b}}} \mu_{(i,\mathbf{y})} = \sum_{i,\mathbf{y}_{ja\mathbf{b}}} \boldsymbol{\Delta}_{ja\mathbf{b}}(i, \mathbf{y}_{ja\mathbf{b}})\, \alpha_{(i,\mathbf{y}_{ja\mathbf{b}})}$$

involving only a sum over local variables $\mathbf{y}_{ja\mathbf{b}}$. Similarly, all of the remaining terms in (13) can be reduced. Finally, we need only add marginalization constraints $\sum_{y_2} \alpha_{(i,y_1,y_2)} = \alpha_{(i,y_1)}$ so that (13) can be expressed equivalently and compactly in terms of α instead of μ.

We implemented this approach and tested it on both synthetic and real data using HMM models for classifications, where the classification variables $\mathbf{y}$ play the role of the hidden state sequence, and the input variables $\mathbf{x}$ play the role of the observations. In our synthetic experiment, we sampled $(\mathbf{x}, \mathbf{y})$ from a 10 variable HMM and repeated the experiment 20 times to obtain the final results. Table 2 shows that M^2BN again outperforms M^3N in

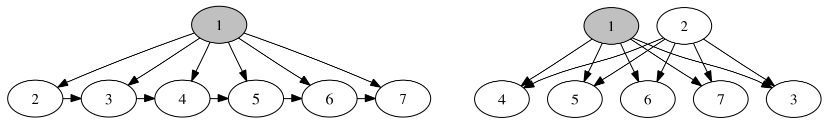

Figure 1: Two Bayesian network models. The classification variable y is shaded.

Table 1: Accuracy results for Figure 1

algorithm	First Network size of training set			Second Network size of training set		
	20	**50**	**100**	**20**	**50**	**100**
M^2BN	0.73235	0.78637	0.80702	0.70073	0.7582	0.77245
M^3N	0.68678	0.74098	0.78537	0.70782	0.72468	0.72138

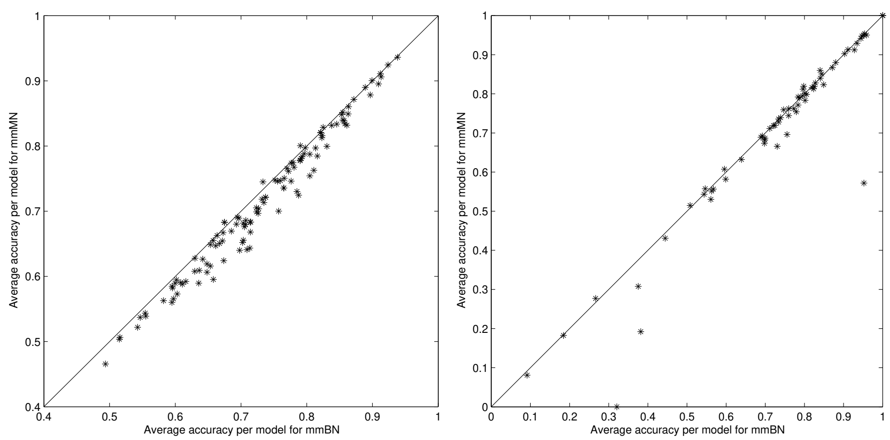

Figure 2: Accuracy comparison between M^2BN and M^3N. Left plot shows accuracy comparison on synthetic Bayesian network models. Right plot shows accuracy comparison on UCI data sets.

Table 2: Accuracy results for M^2BN and M^3N on synthetic multi-variable networks.

algorithm	size of training set			
	5	**20**	**50**	**100**
M^2BN	0.4260	0.4955	0.5325	0.5559
M^3N	0.3513	0.2858	0.3680	0.3790

Table 3: Accuracy results for M^2BN and M^3N on protein secondary structure prediction.

algorithm	size of training set			
	5	**20**	**50**	**100**
M^2BN	0.4814	0.5795	0.6550	0.7495
M^3N	0.2621	0.2838	0.2646	0.2671

controlled experiments where the correct Bayesian network structure is known. We also conducted an experiment on a protein secondary structure database. Here the goal is to predict the sequence of secondary structure labels given an observed amino acid sequence. Table 3 shows that $\mathrm{M}^2\mathrm{BN}$ is competitive with $\mathrm{M}^3\mathrm{N}$ on this data set.

7 Conclusion

We have investigated what we feel is a very natural question; whether a Bayesian network representation can be combined with discriminative training based on the maximum margin criterion of SVMs. We have found that training Bayesian networks under the maximum margin criterion is a hard compuation problem—harder than the standard quadratic program of SVM training. However, reasonable training algorithms can be devised which optimize the margin exactly in special cases, but only heuristically in general cases. Our preliminary experiments show that there might be an advantage to respecting the causal model constraints embodied by a Bayesian network, if indeed these constraints were present during the data generation. In this sense, max margin Bayes nets offer a new way to add prior knowledge to SVMs.

The main directions for future research are to improve the training procedure and add kernels. As in [10], it is possible to consider adding kernels to our local feature representation. The results of [1] show that it is possible to represent our local functions in terms of weighted combinations of training features. Therefore, one extension we are working on is to formulate hybrid classifiers with general kernelized potential functions between $\mathbf{y}$ and $\mathbf{x}$, but standard conditional probability tables between classification variables to preserve their causal relationships.

References

[1] Y. Altun, A. Smola, and T. Hofmann. Exponential families for conditional random fields. In *Proceedings Conference on Uncertainty in Artificial Intelligence (UAI-04)*, 2004.

[2] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov support vector machines. In *Proceedings International Conference on Machine Learning (ICML-03)*, 2003.

[3] D. Bertsekas. *Nonlinear Optimization*. Athena Scientific, 1995.

[4] K. Crammer and Y. Singer. On the algorithmic interpretation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2, 2001.

[5] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[6] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings International Conference on Machine Learning (ICML-01)*, 2001.

[7] A. Ng and M. Jordan. On discriminative vs. generative classifiers. In *Advances in Neural Information Processing Systems 14 (NIPS-01)*, 2001.

[8] R. Raina, Y. Shen, A. Ng, and A. McCallum. Classification with hybrid discriminative / generative models. In *Advances in Neural Information Processing Systems 16 (NIPS-03)*, 2003.

[9] B. Schoelkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[10] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Advances in Neural Information Processing Systems 16 (NIPS-03)*, 2003.

[11] H. Wettig, P. Grunwald, T. Roos, P. Myllymaki, and H. Tirri. On supervised learning of Bayesian network parameters. Technical Report HIIT 2002-1, Helsinki Inst. Info. Tech., 2002.

[12] H. Wettig, P. Grunwald, T. Roos, P. Myllymaki, and H. Tirri. When discriminative learning of Bayesian network parameters is easy. In *Proceedings International Joint Conference on Artificial Intelligence (IJCAI-03)*, 2003.