Maximum Margin Bayesian Networks

Yuhong Guo
Department of Computing Science
University of Alberta
yuhong@cs.ualberta.ca

Linli Xu
School of Computer Science
University of Waterloo
l5xu@cs.uwaterloo.ca

Dale Schuurmans
Department of Computing Science
University of Alberta
dale@cs.ualberta.ca

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, since the parameters in a Bayesian network must satisfy additional normalization constraints that an undirected graphical model need not respect. Unfortunately, these normalization constraints destroy 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 has promise, although the complexity of the optimization poses a nontrivial barrier in practice. Our main intent is simply 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 (Friedman et al., 1997). 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 (Taskar et al., 2003). (An interesting precursor is (Altun et al., 2003).) However, following SVMs, these training algorithms have adopted the Euclidean normalization constraint of support vector machines, which can be accommodated in their frameworks because they rely on an undirected graphical model representation.

In this paper we consider applying the maximum margin methodology to Bayesian networks. Unlike Markov network models, Bayesian networks require the strong local normalization constraints be satisfied. These constraints are at odds with the standard Euclidean (or L_p) normalization constraints of SVMs. Nevertheless, our goal is to explore the possibility 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 problem could be a fragment of a much larger probabilistic causal model, and maintaining a Bayesian network representation will allow one to integrate the learned model with the rest of the model seamlessly. Second, the normalization constraints asserted by a Bayesian network structure capture causal knowledge about the domain. Respecting these constraints is one way to exploit the advantage of Bayesian networks exhibit for intuitively modeling the causal structure of a domain. Removing the normalization constraints turns the Bayesian network into a Markov network, and therefore must necessarily lose the original causal knowledge that was encoded in the constraints.

The remainder of the paper is organized as follows. First, after preliminary definitions, we investigate the notion of margin for Bayesian network classifiers in Section 3, and relate this both to the common conditional likelihood criterion of graphical models, and to the standard margin definition of SVMs. 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 some evidence that the causal information in Bayesian networks can help maximum margin training. Finally, we extend the approach to multivariable classification in Section 6.

In the end, we observe a few drawbacks of maximum margin Bayesian networks (including the fact that they do not allow the kernel trick to be conventiently applied), and thus the main message of this paper is necessarily mixed: On the one hand, maximum margin Bayesian networks allow one to exploit causal prior knowledge effectively, but on the other hand they create additional computational difficulty while blocking the standard kernel trick. 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

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

Here $\boldsymbol{\theta}$ denotes the parameters of the model, j ranges over 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

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

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

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

Unfortunately, these constraints are nonlinear, even though the objective is convex in $\mathbf{w}$. Removing these constraints improves the computational difficulty of training, but also removes the causal interpretability of the model. In this paper, our goal is to stick with the Bayesian network constraints and discover where this leads.

3 DISCRIMINATIVE TRAINING

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 $\max_y P(y|\mathbf{x})$. 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 by

$$\begin{aligned} \log L(\boldsymbol{\theta}) &= \sum_{i=1}^{t} \log P(y^i|\mathbf{x}^i\boldsymbol{\theta}) + \log P(\mathbf{x}^i|\boldsymbol{\theta}) (4) \\ \log CLL(\boldsymbol{\theta}) &= \sum_{i=1}^{t} \log P(y^i|\mathbf{x}^i\boldsymbol{\theta}) \quad (5) \end{aligned}$$

Much of the literature suggests that the latter objective is better suited for classification (Lafferty et al., 2001; Friedman et al., 1997), although recent studies have identified conditions where the former objective is advantageous (Ng & Jordan, 2001).

In this paper we consider two alternative criteria based on the large margin criteria of SVMs, which we refer to as minimum conditional likelihood (MCL) and minimum conditional likelihood ratio (MCLR) respectively

$$\begin{aligned} \log MCL(\boldsymbol{\theta}) &= \min_i \log P(y^i|\mathbf{x}^i\boldsymbol{\theta}) \quad (6) \\ \log MCLR(\boldsymbol{\theta}) &= \min_i \log P(y^i|\mathbf{x}^i\boldsymbol{\theta}) \\ &\quad - \frac{1}{K}\sum_{y=1}^{K} \log P(y|\mathbf{x}^i\boldsymbol{\theta}) \quad (7) \end{aligned}$$

For the two class case, $K = 2$, these two criteria are in fact equivalent. Also in this case, they are both very similar to conditional loglikelihood (5), differing only in taking a min instead of a sum across training examples.

Now, by plugging in the exponential form of the definition of $P(y|\mathbf{x}\mathbf{w})$ into these criteria we will be able to relate the resulting training problem to that of linear

SVMs

$$\log MCL(\mathbf{w}) \;=\; \min_i \boldsymbol{\delta}(\mathbf{x}^i y^i)^\top \mathbf{w} - \log \sum_y e^{\boldsymbol{\delta}(\mathbf{x}^i y)^\top \mathbf{w}} \tag{8}$$

$$\log MCLR(\mathbf{w}) \;=\; \min_i \sum_{y=1}^{K} \left[\boldsymbol{\delta}(\mathbf{x}^i y^i) - \boldsymbol{\delta}(\mathbf{x}^i y)\right]^\top \mathbf{w} \tag{9}$$

The goal here is to maximize these quantities with respect to the weight vector $\mathbf{w}$. Of course, maximizing these inner products is trivial if $\mathbf{w}$ is not constrained. At this point, the standard SVM formulation imposes a Euclidean normalization constraint that $\|\mathbf{w}\|_2 = 1$, which sets the weights to maximize the Euclidean margin (Schoelkopf & Smola, 2002). For the second criterion specifically, our formulation recovers standard versions of multiclass SVMs proposed in (Crammer & Singer, 2001) (ignoring slacks) expressed over features determined by the Bayesian network.

This specific connection is the main observation of (Taskar et al., 2003; Altun et al., 2003), who proceed to use standard SVM training criteria over these features. (We consider multvariable classification in Section 6 below.) Note however that the solution weight vector for this problem cannot be substituted into the Bayesian network representation, because it will not satisfy the proper normalization constraints (3). The previous techniques of (Taskar et al., 2003; Altun et al., 2003) were able to proceed by using an undirected graphical model which can 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.

Our approach that we consider in this paper is to preserve representability as a Bayesian network, which requires one to solve the maximum margin training criteria (8) and (9) with respect to the alternative normalization constraints (3). Unfortunately, the constraints in $\mathbf{w}$ are highly 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 to remove the normalization constraints entirely also does not work in this case, since the standard reparameterization $\theta_{ja\mathbf{b}} = e^{\omega_{ja\mathbf{b}}} / \sum_a e^{\omega_{ja\mathbf{b}}}$ creates an objective

$$P(\mathbf{x}|\boldsymbol{\omega}) \;=\; \exp\left(\sum_{ja\mathbf{b}} 1_{(\mathbf{x}_j = a\mathbf{b})} \left[\omega_{ja\mathbf{b}} - \log \sum_a e^{\omega_{ja\mathbf{b}}}\right]\right)$$

that is neither convex nor concave over $\boldsymbol{\omega}$. Thus, if we hope to solve the maximum margin Bayesian network training problem exactly, even for special cases, we require a more subtle approach.

4 A TRAINING ALGORITHM

Although solving for the maximum margin Bayesian network parameters appears to be hard in general, 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. Below we will work with the MCL criterion (8) although a similar derivation also works for (9). Note first that (8) is a convex objective function in $\mathbf{w}$. Unfortunately, we have to maximize (8) with respect to the nonlinear equality constraints (3). However, the basic observation is that the problem can be made convex simply by relaxing these equality constraints to inequality constraints, and thus obtain a simple relaxation of the problem which allows us to obtain a global solution

$$\arg\max_{\mathbf{w}} \min_i \;\; \boldsymbol{\delta}(\mathbf{x}^i y^i)^\top \mathbf{w} - \log \sum_y e^{\boldsymbol{\delta}(\mathbf{x}^i y)^\top \mathbf{w}} \tag{10}$$

$$\text{subject to } \sum_a e^{w_{ja\mathbf{b}}} \le 1 \text{ for all } j, \mathbf{b}$$

$$\begin{aligned} = \;\; & \arg\min_{\mathbf{w},\beta} \;\; -\beta \quad \text{subject to} \\ & \beta - \boldsymbol{\delta}(\mathbf{x}^i y^i)^\top \mathbf{w} + \log \sum_y e^{\boldsymbol{\delta}(\mathbf{x}^i y)^\top \mathbf{w}} \;\le\; 0 \;\; \forall i \\ & \sum_a e^{w_{ja\mathbf{b}}} - 1 \le 0 \text{ for all } j, \mathbf{b} \end{aligned} \tag{11}$$

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

$$\begin{aligned} L_0(\mathbf{w}, \beta, \boldsymbol{\mu}, \boldsymbol{\lambda}) \;=\; & \beta \; + \\ & \sum_i \mu_i \left(\beta - \boldsymbol{\delta}(\mathbf{x}^i y^i)^\top \mathbf{w} + \log \sum_y e^{\boldsymbol{\delta}(\mathbf{x}^i y)^\top \mathbf{w}}\right) \\ & + \sum_{j,\mathbf{b}} \lambda_{j\mathbf{b}} \left(\sum_a e^{w_{ja\mathbf{b}}} - 1\right) \end{aligned}$$

This gives us an equivalent problem to (11)

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

First, it turns out to be easy to eliminate β from this problem, since $\partial L_0 / \partial \beta = -1 + \sum_i \mu_i$, and setting this to 0 implies $\sum_i \mu_i = 1$. If we enforce this constraint, we can plug this equation back into the Lagrangian

$$\begin{aligned} L_1(\mathbf{w}, \boldsymbol{\mu}, \boldsymbol{\lambda}) &= \sum_i \mu_i \left(\log \sum_y e^{\boldsymbol{\delta}(\mathbf{x}^i y)^\top \mathbf{w}} - \boldsymbol{\delta}(\mathbf{x}^i y^i)^\top \mathbf{w} \right) \\ &\quad + \sum_{j,\mathbf{b}} \lambda_{j\mathbf{b}} \left(\sum_a e^{w_{ja\mathbf{b}}} - 1 \right) \end{aligned}$$

An equivalent optimization problem to (12) is therefore

$$\min_{\mathbf{w}} \max_{\boldsymbol{\mu},\boldsymbol{\lambda}} L_1(\mathbf{w}, \boldsymbol{\mu}, \boldsymbol{\lambda}) \text{ s.t. } \boldsymbol{\mu} \geq 0, \; \sum_i \mu_i = 1, \; \boldsymbol{\lambda} \geq 0 \quad (13)$$

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.

We now attempt to solve for $\mathbf{w}$ for a given $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$. Taking the partial derivative with repect to $w_{ja\mathbf{b}}$ and setting this equal to 0 shows that we seek a $\mathbf{w}$ that satisfies

$$\sum_i \mu_i \, p(y_{ja\mathbf{b}} | \mathbf{x}^i) \; + \; \lambda_{j\mathbf{b}} \, e^{w_{ja\mathbf{b}}} \quad = \quad \textstyle\sum_i \mu_i \, \delta_{ja\mathbf{b}}(y^i, \mathbf{x}^i) \quad (14)$$

for all $j, a, \mathbf{b}$, where here we have used the substitution

$$p(y_{ja\mathbf{b}} | \mathbf{x}^i) \quad = \quad \sum_y \delta_{ja\mathbf{b}}(y, \mathbf{x}^i) \, e^{\boldsymbol{\delta}(\mathbf{x}^i y)^\top \mathbf{w}} / \sum_y e^{\boldsymbol{\delta}(\mathbf{x}^i y)^\top \mathbf{w}}$$

This primal problem can be solved in many ways, including iterative proportional fitting. Thus, many forms of primal-dual search algorithms are able to effectively solve the problem (10). We use a straightforward alternating gradient approach in our experiments below.

Of course, the solutions obtained to (10) may not be representable in a Bayesian network because the parameters $\mathbf{w}$ are sub-normalized, not normalized. The main question that remains is when can these subnormalized solutions 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 (Wettig et al., 2002; Wettig et al., 2003). 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 $P(y|\mathbf{x}\boldsymbol{\theta})$ 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 (10), and therefore we obtain a global maximum of (8) with respect to (3).

5 EXPERIMENTAL RESULTS

To evaluate the utility of learning maximum margin Bayesian networks, we conducted some preliminary 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. We experimented with several network topologies and parameterizations, and compared maximum margin Bayesian networks trained according to (8) s.t. (3) to several other approaches, including: maximum margin Markov networks (SVMs) trained according to (9) with slacks, maximum conditional likelihood (5), and maximum joint likelihood (4). The results are for 20 repetitions of the training sample, for the networks shown in Figures 1 and 2.

Tables 1 and 2 show that the techniques behave similarly, but show an advantage for maxmargBN over maxmargMN. 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, maxmargMN, is slightly behind the other methods which respect 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.

We also experimented with real data from the UCI repository. In these cases, we formulated a Bayesian

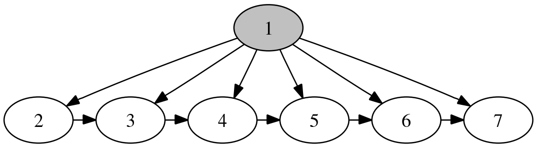

Figure 1: 8-node chain augmented Naive Bayes model. The classification variable y is shaded.

Table 1: Accuracy results for Figure 1

algorithm	size of training set		
	20	**50**	**100**
maxL	0.7335	0.79085	0.82102
maxCL	0.71735	0.79242	0.82203
maxmargBN	0.73235	0.78637	0.80702
maxmargMN	0.68678	0.74098	0.78537

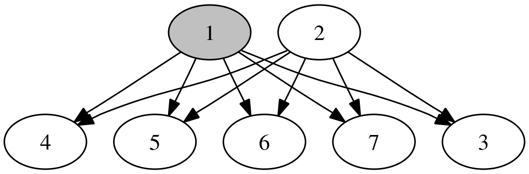

Figure 2: 8-node twin-parent Naive Bayes model. The classification variable y is shaded.

Table 2: Accuracy results for Figure 2

algorithm	size of training set		
	20	**50**	**100**
maxL	0.69462	0.7693	0.80235
maxCL	0.6887	0.76477	0.80383
maxmargBN	0.70073	0.7582	0.77245
maxmargMN	0.70782	0.72468	0.72138

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.

Tables 3–7 show the results. Interestingly, these results generally show an advantage for learning techniques that respect the Bayesian network constraints, and a disadvantage for those that ignore this information. Surprisingly, maximum joint likelihood performed well in our experiments. Unsurprisingly, maximum conditional likelihood performed very well. MaxmargBN performed best on one data set.

6 MULTIVARIABLE EXTENSION

Finally, we extend the maximum margin Bayesian network approach to multivariable classification. This was the main idea of (Taskar et al., 2003; Altun et al., 2003). 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 in approach, and we can seek to maximize the criteria (8) and (9) as before. The only new challenge is coping with the exponential sum over $\mathbf{y}$. However, the derivation of our training algorithm in Section 4 is not significantly affected by this extension. In fact, we find that the derivative $\partial L_1 / \partial w_{ja\mathbf{b}}$ now computes the marginal probability over the local y values $y_{ja\mathbf{b}}$ that match the local function j on pattern $a\mathbf{b}$. We use standard probabilistic inference techniques (for example, forward-backward) to calculate these marginals, which then allows us to calculate the gradients for the primal-dual optimizer.

We implemented this approach and tested it on a synthetic HMM model, 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. We sampled $(\mathbf{x}, \mathbf{y})$ from a 10 variable HMM and repeated the experiment 20 times to obtain the final results.

Table 8 shows that the maximum margin approach is competetive with maxL and maxCL, and outperforms them at sample size 100. Unfortunately, at the time of submission we did not have a multivariable version of maximum margin Markov networks available for a comparison. This will be added. Nevertheless, the preliminary results show credible performance for max margin Bayesian networks.

Table 3: Accuracy on UCI data sets

	Australian	**Breast**	**Chess**
maxL	0.85906	0.95539	0.6875
maxCL	0.85145	0.95137	0.7465
maxmargBN	0.77065	0.9521	0.6875
maxmargMN	0.80072	0.93163	0.721

Table 4: Accuracy on UCI data sets

	Corral	**Crx**	**Diabetes**
maxL	0.75146	0.69637	0.77138
maxCL	0.81748	0.7044	0.77138
maxmargBN	0.75922	0.63327	0.74667
maxmargMN	0.68738	0.52352	0.7239

Table 5: Accuracy on UCI data sets

	Flare	**MofN**	**Vote**
maxL	0.8211	0.893	0.94943
maxCL	0.8211	0.9915	0.9477
maxmargBN	0.8211	1.0	0.94483
maxmargMN	0.8211	0.762	0.94138

Table 6: Accuracy on UCI data sets

	Iris	**Vehicle**	**Glass**
maxL	0.9267	0.5479	0.6511
maxCL	0.9267	0.5494	0.6128
maxmargBN	0.9333	0.4686	0.6340
maxmarginMN	0.5267	0.5216	0.6106

Table 7: Accuracy on UCI data sets

	Lymphography	**Waveform-21**
maxL	0.7900	0.5552
maxCL	0.7905	0.5785
maxmargBN	0.8033	0.5696
maxmarginMN	0.7414	0.6663

Table 8: Accuracy on an 8 node HMM model

algorithm	size of training set		
	20	**50**	**100**
maxL	0.7354	0.7951	0.7946
maxCL	0.7249	0.7765	0.7856
maxmargBN	0.6863	0.7541	0.8103

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 the outcome of this investigation are mixed: 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.

On the other hand, 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. Unfortunately, this opportunity also comes with a cost: max margin Bayes nets do not conveniently allow the kernel trick, which loses one of the biggest advantages of SVMs.

In the end, it appears that maximum margin Bayesian networks might be a viable learning technique in multivariable classification problems where there is strong prior causal knowledge. However, their utility my be limited by computational intractability and lack of a kernel extension.

References

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

Crammer, K., & Singer, Y. (2001). On the algorithmic interpretation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, *2*.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, *29*, 131–163.

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

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

Schoelkopf, B., & Smola, A. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.

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

Wettig, H., Grunwald, P., Roos, T., Myllymaki, P., & Tirri, H. (2002). *On supervised learning of Bayesian network parameters* (Technical Report HIIT Technical Report 2002-1). Helsinki Institute for Information Technology.

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