Order-based Discriminative Structure Learning for Bayesian Network Classifiers

Franz Pernkopf
Department of Electrical Engineering
Graz University of Technology
A-8010 Graz, Austria
pernkopf@tugraz.at

Jeff Bilmes
Department of Electrical Engineering
University of Washington
Seattle, WA 98195
bilmes@ee.washington.edu

Abstract

We introduce a simple empirical order-based greedy heuristic for learning discriminative Bayesian network structures. We propose two metrics for establishing the ordering of N features. They are based on the conditional mutual information. Given an ordering, we can find the discriminative classifier structure with $\mathcal{O}(N^q)$ score evaluations (where constant q is the maximum number of parents per node). We present classification results on the UCI repository (Merz, Murphy, & Aha 1997), for a phonetic classification task using the TIMIT database (Lamel, Kassel, & Seneff 1986), and for the MNIST handwritten digit recognition task (LeCun *et al.* 1998). The discriminative structure found by our new procedures significantly outperforms generatively produced structures, and achieves a classification accuracy on par with the best discriminative (naive greedy) Bayesian network learning approach, but does so with a factor of ∼10 speedup. We also show that the advantages of generative discriminatively structured Bayesian network classifiers still hold in the case of missing features.

1 Introduction

Learning the structure of a Bayesian networks is typically hard. There have been a number of negative results over the past years, showing that learning various forms of optimal constrained Bayesian network in a maximum likelihood (ML) sense is NP-complete (including paths (Meek 1995), polytrees (Dasgupta 1997), k-trees (Arnborg, Corneil, & Proskurowski 1987), and general Bayesian networks (Geiger & Heckerman 1996)). Learning the best "discriminative structure" is no less difficult, largely because the cost functions that are needed to be optimized do not in general decompose[1]. As of yet, however, there has not been any hardness results in the discriminative case.

There have been a number of recent *heuristic* approaches proposed for learning discriminative models. For example, standard logistic regression is extended to more general Bayesian networks in (Greiner *et al.* 2005) – they optimize parameters with respect to the conditional likelihood (CL) using a conjugate gradient method. Similarly, in (Roos *et al.* 2005) conditions are provided for general Bayesian networks under which correspondence to logistic regression holds. In (Grossman & Domingos 2004) the CL function is used to learn a discriminative structure. The parameters are set using ML learning. They use a greedy hill climbing search with the CL function as a scoring measure, where at each iteration one edge is added to the structure which conforms with the restrictions of the network topology (e.g., tree augmented naive Bayes (TAN)) and the acyclicity property of Bayesian networks. In a similar algorithm, the classification rate (CR) has also been used for discriminative structure learning (Keogh & Pazzani 1999). This approach is computationally expensive, as a complete re-evaluation of the training set is needed for each considered edge. The CR (equivalently, empirical risk) is the discriminative criterion with the fewest approximations, so it is expected to perform well with sufficient training data. Bilmes (Bilmes 2000; 1999) introduced the *explaining away residual* (EAR) for discriminative structure learning of dynamic Bayesian networks for speech recognition applications. The EAR measure is in fact an approximation to the expected log class posterior distribution. Many generative structure learning algorithms have been proposed. An excellent overview is provided in (Murphy 2002).

An empirical and theoretical comparison of discriminative and generative classifiers (logistic regression and naive Bayes (NB)) is given in (Ng & Jordan 2002). It is shown that for small sample sizes the generative NB classifier can outperform a discriminatively trained model. An experimental comparison of discriminative and generative parameter training on both discriminatively and generatively structured Bayesian network classifiers has been performed in (Pernkopf & Bilmes 2005).

In this work, we introduce order-based greedy algorithms for learning a discriminative network structure. The classifiers are restricted to NB, TAN (i.e. 1-tree) and 2-tree structures. We look first for an ordering of the N features according to a classification based information measures. Given the ordering, we can find the discriminative network structure with $\mathcal{O}(N^q)$ score evaluations (constant q limits the number of parents per node). We learn a e.g., TAN classifier,

[1] By using the term "discriminative structure learning", we mean simply that the goal of discrete optimization is to minimize a cost function that is suitable for reducing classification errors, such as conditional likelihood (CL) or classification rate (CR).

which can be discriminatively optimized in $\mathcal{O}\left(N^2\right)$ using the CR. Our order-based structure learning is based on the observations in (Buntine 1991) and the framework is similar to the K2 algorithm proposed in (Cooper & Herskovits 1992), however, we use a discriminative scoring metric and suggest approaches for establishing the variable ordering based on conditional mutual information (CMI) (Cover & Thomas 1991). We provide results showing that the order-based heuristic provides comparable results to the best procedure - the naive greedy heuristic using the CR score, but it requires only one tenth of the time. Furthermore, we empirically show that the chosen approaches for ordering the variables improve the classification performance compared to simple random orderings. Additionally, we experimentally compare both discriminative and generative parameter training on *both* discriminative *and* generatively structured Bayesian network classifiers. Finally, classification results are shown when missing features are present.

The paper is organized as follows: In Section 2, we briefly review Bayesian networks. In Section 3, a practical case is made for why discriminative structure can be desirable. Section 4 introduces our order-based greedy heuristic. Experiments are shown in Section 5. Section 6 concludes.

2 Bayesian network classifiers

A Bayesian network (Pearl 1988) $\mathcal{B} = \langle \mathcal{G}, \Theta \rangle$ is a directed acyclic graph $\mathcal{G} = (\mathbf{Z}, \mathbf{E})$ consisting of a set of nodes $\mathbf{Z}$ and a set of directed edges $\mathbf{E}$ connecting the nodes. This graph represents factorization properties of the distribution of a set of random variables $\mathbf{Z} = \{Z_1, \ldots, Z_{N+1}\}$, where each variable in $\mathbf{Z}$ has values denoted by lower case letters $\{z_1, z_2, \ldots, z_{N+1}\}$. We use boldface capital letters, e.g., $\mathbf{Z}$, to denote a set of random variables and correspondingly lower case boldface letters denote a set of instantiations (values). Without loss of generality, in Bayesian network classifiers the random variable Z_1 represents the class variable $C \in \{1, \ldots, |C|\}$, $|C|$ is the cardinality of C, $\mathbf{X}_{1:N} = \{X_1, \ldots, X_N\} = \{Z_2, \ldots, Z_{N+1}\}$ denote the N attributes of the classifier. Each node represents a random variable, while missing edges encodes conditional independence properties (Pearl 1988). These relationships reduce both number of parameters and required computation. The set of parameters which quantify the network are represented by Θ. Each node Z_j is represented as a local conditional probability distribution given its parents Z_{Π_j}. The joint probability distribution is given as a function of the local conditional probability distributions according to $P_\Theta(\mathbf{Z}) = \prod_{j=1}^{N+1} P_\Theta\left(Z_j | Z_{\Pi_j}\right)$.

3 Why discriminative structures

Finding a discriminative structure really means several things. First, a commitment has been made to use a generative model for classification purposes; the alternative being a "discriminative" classifier such as logistic regression or support vector machines (SVMs) (Schölkopf & Smola 2001). There are a number of reasons why one might, in certain contexts, prefer a generative to a discriminative model including: parameter tying and domain knowledge-based hierarchical decomposition is facilitated, it is easy to work with structured data, there is less sensitivity to training data class skew, generative models can still be trained and structured discriminatively, and it is easy to work with missing features by marginalizing over the unknown variables.

Secondly, there is a "discriminative" cost function that scores the quality of each structure. The ideal cost function is empirical risk (what we call CR), which can be implicitly regularized by constraining optimization to be over only a given model family (e.g., k-trees), assuming sufficient training data. We are given training data $\mathcal{S}$ consisting of M samples $\mathcal{S} = \{(c^m, \mathbf{x}^m_{1:N})\}_{m=1}^M$. Also, the expression $\delta\left(\mathcal{B}_\mathcal{S}\left(\mathbf{x}^m_{1:N}\right), c^m\right) = 1$ if the classifier $\mathcal{B}_\mathcal{S}\left(\mathbf{x}^m_{1:N}\right)$ assigns the correct class label c^m to the attribute values $\mathbf{x}^m_{1:N}$ and 0 otherwise. CR is defined as $CR = \frac{1}{M}\sum_{m=1}^M \delta\left(\mathcal{B}_\mathcal{S}\left(\mathbf{x}^m_{1:N}\right), c^m\right)$, (a multi-class generalization of 0/1-loss) which is hard to optimize. Alternative continuous, (often) differentiable, and (sometimes) convex, cost functions exist which may upper-bound CR are thus used and include conditional (log) likelihood $CLL\left(\mathcal{B}|\mathcal{S}\right) = \log \prod_{m=1}^M P_\Theta\left(C = c^m | \mathbf{X}_{1:N} = \mathbf{x}^m_{1:N}\right)$. These are typically augmented by a weighted regularization term (to bias against complex models).

It is well known (Friedman, Geiger, & Goldszmidt 1997) that optimizing the log likelihood $LL\left(\mathcal{B}|\mathcal{S}\right) = \log \prod_{m=1}^M P_\Theta\left(C = c^m, \mathbf{X}_{1:N} = \mathbf{x}^m_{1:N}\right)$ does not necessarily optimize either of the above two, although LL is widely used. The bad news is that neither CL nor CR is decomposable as is LL.

This paper deals with the last two aforementioned aspects of generative models. In particular, we show that not only the right discriminative structure learning procedure can improve classification performance and render generative training less important (Section 5), but also that the loss of a "generative meaning" of a generative model (when it is structured discriminatively) does not impair the generative model's ability to easily deal with missing features (Figure 3).

In the following, we present a simple synthetic example (similar to (Narasimhan & Bilmes 2005)) and results which indicate when a discriminative structure would be necessary for good classification performance in a generative model, regardless of the parameter learning method. The model consists of 3 binary valued attributes X_1, X_2, X_3 and a binary uniformly distributed class variable C. $\bar{X}_1$ denotes the negation of X_1. We have the following probabilities for both classes:

$$X_1 := \begin{cases} 0 & \text{with probability } 0.5 \\ 1 & \text{with probability } 0.5 \end{cases} \tag{1}$$

$$X_2 := \begin{cases} X_1 & \text{with probability } 0.5 \\ 0 & \text{with probability } 0.25 \\ 1 & \text{with probability } 0.25 \end{cases} \tag{2}$$

For class 1, X_3 is determined according to the following:

$$X_3 := \begin{cases} X_1 & \text{with probability } 0.3 \\ X_2 & \text{with probability } 0.5 \\ 0 & \text{with probability } 0.1 \\ 1 & \text{with probability } 0.1 \end{cases}. \tag{3}$$

For class 2, X_3 is given by:

$$X_3 := \begin{cases} \bar{X}_1 & \text{with probability } 0.3 \\ X_2 & \text{with probability } 0.5 \\ 0 & \text{with probability } 0.1 \\ 1 & \text{with probability } 0.1 \end{cases} . \qquad (4)$$

For both classes, the dependence between $X_1 - X_2$ is strong. The dependence $X_2 - X_3$ is stronger than $X_1 - X_3$, but only from a generative perspective (i.e., $I(X_2; X_3) > I(X_1; X_3)$ and $I(X_2; X_3|C) > I(X_1; X_3|C)$). Hence, if we were to use the strength of mutual information, or conditional mutual information, to choose the edge, we would choose $X_2 - X_3$. However, it is the $X_1 - X_3$ dependency that enables discrimination between the classes. Sampling from this distribution, we first learn structures using generative and discriminative methods, and then we perform parameter training on these structures using either ML or CL (Greiner *et al.* 2005). For learning a generative TAN structure, we use the algorithm proposed by (Friedman, Geiger, & Goldszmidt 1997) which is based on optimizing the CMI between attributes given the class variable. For learning a discriminative structure, we apply our order-based algorithm proposed in Section 4 (we note that optimizing the EAR measure (Pernkopf & Bilmes 2005) leads to similar results in this case).

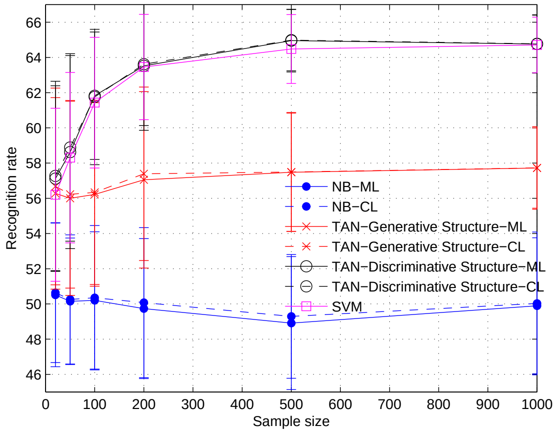

Figure 1: Generative and discriminative learning of Bayesian network classifiers on synthetic data.

Figure 1 compares the classification performance of these various cases, and in addition we show results for a NB classifier, which resorts only to random guessing. Additionally, we provide the classification performance achieved with SVM using a radial basis function (RBF) kernel[2]. On the x-axis, the training set *sample size* varies according to $\{20, 50, 100, 200, 500, 1000\}$ and the test data set contains 1000 samples. Plots are averaged over 100 independent simulations. The solid line is the performance of the classifier with ML parameter learning, whereas, the dashed line corresponds to CL parameter training.

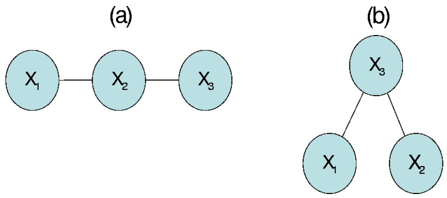

Figure 2: (a) Generatively learned 1-tree, (b) Discriminatively learned 1-tree.

Figure 2 shows (a) the generative (b) the discriminative 1-tree over the attributes of the resulting TAN network (the class variable which is the parent of each feature is not shown in this figure). A generative model prefers edges between $X_1 - X_2$ and $X_2 - X_3$ which do not help discrimination. The dependency between X_1 and X_3 enables discrimination to occur. Note that discriminative parameter learning is irrelevant and for the generative model, only a discriminative structure enables correct classification. The performance of the SVM is similar to our discriminatively structured Bayesian network classifier. However, the SVM is not generative. Therefore, when a generative model is desirable (see the reasons why this might be the case above), there is clearly a need for good discriminative structure learning procedures.

4 Order-based greedy algorithms

It was first noticed in (Buntine 1991; Cooper & Herskovits 1992) that the best network consistent with a given variable ordering can be found with $\mathcal{O}(N^q)$ score evaluations where q is the upper bound of parents per node. These facts were recently exploited in (Teyssier & Koller 2005) where generative structures were learned. Here, we are inspired by these ideas but applied to the case of learning of discriminative structures. Also, unlike (Teyssier & Koller 2005), we establish only one ordering, and since our scoring cost is discriminative, it does not decompose and the learned discriminative structure is not necessarily optimal. However, experiments show good results at lower computational costs.

Our procedure first looks for a total ordering $\prec$ of the variables $\mathbf{X}_{1:N}$ according to the CMI. If the graph is consistent with the ordering $X_i \prec X_j$ then the parent $X_{\Pi_j} \in \mathbf{X}_{\Pi_j}$ is one of the variables which appears before X_j in the ordering, where $\mathbf{X}_{\Pi_j}$ is the set of possible parents for X_j. This constraint ensures that the network stays acyclic. In the second step of the algorithm, we select X_{Π_j} for X_j under constant k maximizing either CL or CR.

4.1 Step 1: Establishing an order $\prec$

We propose and evaluate two separate procedures for establishing the ordering $\prec$ of the nodes. In particular, we use CMI. In the experiments, both metrics are compared against various *random orderings* (RO) of the attributes (see Section 5) to show that they are doing better than chance. The two procedures are defined next.

1: CMI The mutual information $I(C; \mathbf{X}_{1:N})$ measures the degree of dependence between the features $\mathbf{X}_{1:N}$ and

[2] The SVM uses two parameters C^* and σ, where C^* is the penalty parameter for the errors of the non-separable case and σ is the parameter for the RBF kernel. We set the values for these parameters to $C^* = 3$ and $\sigma = 1$.

the class, and we have that $I\left(C;\mathbf{X}_{1:N}\right) = H\left(C\right) - H\left(C|\mathbf{X}_{1:N}\right)$ where the negative entropy $-H\left(C|\mathbf{X}_{1:N}\right) = E_{P(C,\mathbf{X}_{1:N})}\log P\left(C|\mathbf{X}_{1:N}\right)$ is related to what ideally should be optimized.

Our greedy approach to finding an order first chooses a feature that is most informative about C. The next node in the order is the node that is most informative about C conditioned on the first node. More specifically, our algorithm forms an ordered sequence of nodes $\mathbf{X}_{\prec}^{1:N} = \left\{X_{\prec}^{1}, X_{\prec}^{2}, \dots, X_{\prec}^{N}\right\}$ according to

$$X_{\prec}^{j} \leftarrow \arg\max_{X\in\mathbf{X}_{1:N}\setminus\mathbf{X}_{\prec}^{1:j-1}} \left[I\left(C;X|\mathbf{X}_{\prec}^{1:j-1}\right)\right], \tag{5}$$

where $j \in \{1,\dots,N\}$. We note that any conditional mutual information query can be computed efficiently making use of the sparsity of the joint probability distribution (i.e. by essentially making one pass over the training data). Therefore, we split $I\left(C;X|\mathbf{X}_{\prec}^{1:j-1}\right)$ into joint entropy terms as $I\left(C;A|\mathbf{B}\right) = H\left(C,\mathbf{B}\right) - H\left(\mathbf{B}\right) - H\left(C,A,\mathbf{B}\right) + H\left(A,\mathbf{B}\right)$. Utilizing the sparsity of the joint distribution, the nonzero elements are represented by one discrete random variable Y which is further used to determine the joint entropy according to $H\left(Y\right) = -\sum_{y=1}^{|Y|} P\left(Y=y\right)\log P\left(Y=y\right)$ where $|Y|$ denotes the cardinality of Y which is determined by the number of different patterns in the data. Of course, as the number of variables in $\mathbf{X}_{\prec}^{1:j-1}$ increases the estimates of the joint probability suffer and the ordering becomes less reliable. In practice, the number of variables in $\mathbf{X}_{\prec}^{1:j-1}$ should be restricted (e.g., as in the following).

2: CMISP: For a 1-tree each variable $X_{\prec}^{j}$ has one single parent (SP) X_{Π_j} which is selected from the variables $\mathbf{X}_{\prec}^{1:j-1}$ appearing before $X_{\prec}^{j}$ in the ordering. This leads to a simple variant of CMI where we condition the CMI only on a single variable out of $\mathbf{X}_{\prec}^{1:j-1}$. In particular, an ordered sequence of nodes $\mathbf{X}_{\prec}^{1:N}$ is determined by

$$X_{\prec}^{j} \leftarrow \arg\max_{X\in\mathbf{X}_{1:N}\setminus\mathbf{X}_{\prec}^{1:j-1}} \left[\max_{X_{\prec}\in\mathbf{X}_{\prec}^{1:j-1}} \left[I\left(C;X|X_{\prec}\right)\right]\right]. \tag{6}$$

4.2 Step 2: Selecting parents w.r.t. a given order to form a k-tree

Once we have the ordering $\mathbf{X}_{\prec}^{1:N}$, we select $X_{\Pi_j} \in \mathbf{X}_{\Pi_j} = \mathbf{X}_{\prec}^{1:j-1}$ for each $X_{\prec}^{j}$ ($j \in \{3,\dots,N\}$). When the size of $\mathbf{X}_{\Pi_j}$ (i.e. N) and of k are small we can even use a computational costly scoring function to find X_{Π_j}. In case of a large N, we can restrict the size of the parent set $\mathbf{X}_{\Pi_j}$ similar to the *sparse candidate algorithm* (Friedman, Nachman, & Peer 1999). Basically, either the CL or the CR can be used as cost function to select the parents for learning a discriminative structure. We restrict our experiments to CR for parent selection (empirical results show it performed better). The parameters are trained using ML learning. We connect a parent to $X_{\prec}^{j}$ only when CR is improved, and otherwise leave $X_{\prec}^{j}$ parentless (except C). This might result in a partial 1-tree (forest) over the attributes. Our algorithm can be easily extended to learn k-trees ($k > 1$) by choosing more than one parent, using $\mathcal{O}\left(N^{1+k}\right)$ score evaluations (corresponds to $\mathcal{O}\left(N^{q}\right)$).

5 Experiments

We present classification results on 25 data sets from the UCI repository (Merz, Murphy, & Aha 1997), for frame- and segment-based phonetic classification using the TIMIT database (Lamel, Kassel, & Seneff 1986), and for handwritten digit recognition (LeCun *et al.* 1998). We use NB, TAN, and 2-tree network structures. All different combinations of the following parameter/structure learning approaches are used to learn the classifiers:

- Generative (ML) (Pearl 1988) and discriminative (CL) (Greiner *et al.* 2005) parameter learning.
- CMI: Generative structure learning using CMI as proposed in (Friedman, Geiger, & Goldszmidt 1997).
- CR: Discriminative structure learning with naive greedy heuristic using CR as scoring function (Keogh & Pazzani 1999).
- RO-CR: Discriminative structure learning using random ordering (RO) in step 1 and CR for parent selection in step 2 of the order-based heuristic.
- OMI-CR: Discriminative structure learning using CMI for ordering the variables (step 1) and CR for parent selection in step 2 of the order-based heuristic.
- OMISP-CR: Discriminative structure learning using CMI conditioned on a single variable for ordering the variables (step1) and CR for parent selection in step 2 of the order-based heuristic.

Any continuous features were discretized using the procedure from (Fayyad & Irani 1993) where the codebook is produced using only the training data. Throughout our experiments, we use exactly the same data partitioning for each training procedure. We performed simple smoothing, where zero probabilities in the conditional probability tables are replaced with small values ($\varepsilon = 0.00001$). For discriminative parameter learning, the parameters are initialized to the values obtained by the ML approach (Greiner *et al.* 2005). The gradient descent parameter optimization is currently terminated after a specified number of iterations (specifically 20).

5.1 Data characteristics

UCI Data: We use 25 data sets from the UCI repository (Merz, Murphy, & Aha 1997) and from (Kohavi & John 1997). The same data sets, 5-fold cross-validation, and train/test learning schemes as in (Friedman, Geiger, & Goldszmidt 1997) are employed.

TIMIT-4/6 Data: This data set is extracted from the TIMIT speech corpus using the dialect speaking region 4 which consists of 320 utterances from 16 male and 16 female speakers. Speech frames are classified into either four or six classes using 110134 and 121629 samples, respectively. Each sample is represented by 20 features. We perform classification experiments on data of male speakers (Ma), female speakers (Fe), and both genders (Ma+Fe). The data have been split into 2 mutually exclusive subsets of where 70% is used for training and 30% for testing. More details can be found in (Pernkopf & Bilmes 2007).

TIMIT-39 Data: The difference to TIMIT-4/6 is as follows: The phonetic transcription boundaries specify a set of frames belonging to a particular phoneme. From this set of frames - the phonetic segment - a single feature vector is derived. In accordance with (Halberstadt & Glass 1997) we combine the 61 phonetic labels into 39 classes, ignoring glottal stops. For training, 462 speakers from the standard NIST training set have been used. For testing the remaining 168 speakers from the overall 630 speakers were employed. We derive from each phonetic segment 66 features, i.e. MFCC's, Derivatives, and log duration. All together we have 140173 training samples and 50735 testing samples. More information on the data set is given in (Pernkopf & Bilmes 2007).

MNIST Data: We evaluate our classifiers on the MNIST dataset of handwritten digits (LeCun *et al.* 1998) which contains 60000 samples for training and 10000 digits for testing. The digits are centered in a 28×28 gray-level image. We resample these images at a resolution of 14×14 pixels which results in 196 features.

5.2 Results

Table 1 presents the averaged classification rates over the 25 UCI and 6 TIMIT-4/6 data sets. Additionally, we report the CR on TIMIT-39 and MNIST. The classification performance on individual data sets can be found in (Pernkopf & Bilmes 2007). For RO-CR we summarize the performance over 1000 random orderings using the mean (Mean), minimum (Min), and maximum (Max) CR (we use only 100 random orders for TIMIT-4/6 though). For Max (Min), we take the structure which achieves the maximum (minimum) CR over the 1000 random orderings (resp. 100 orders for TIMIT-4/6) on the training set and report the performance on the test set. For TAN-RO-CR on the UCI and TIMIT-4/6 data, the structure with maximum performance on the training set sometimes performs poorly on the test set. The average over the data sets shows that the worst structures on the training sets perform better on the test sets than the best structures on the training sets, presumably due to overfitting. These results do show, however, that choosing from a collection of arbitrary orders and judging based on training set performance is not likely to perform well on the test set. Our heuristics do improve over these orders.

The discriminative 2-tree performs best, i.e. for TIMIT-4/6 the difference is significant. The structure of Bayesian networks is implicitly regularized when the optimization is fixed over a given model family (e.g., 1-trees) assuming sufficient training data. For 2-trees we noticed that the data are overfitted without regularization. Therefore, we introduce 5-fold cross validation on the *training* data to find the optimal classifier structure.

For TAN structures, the CR objective function produces the best performing networks. The evaluation of the CR measure is computationally very expensive, since a complete re-evaluation of the training set is needed for each considered edge. However, due to the ordering of the variables in the order-based heuristics, we can reduce the number of CR evaluations from $\mathcal{O}\left(N^3\right)$ to $\mathcal{O}\left(N^2\right)$. The order-based heuristics, i.e. RO-CR, OMI-CR, OMISP-CR, achieve a similar performance at a much lower computational cost.

Table 1: Averaged classification results for 25 UCI and 6 TIMIT-4/6 data sets and classification results for TIMIT-39 and MNIST with standard deviation. Best results use bold font.

Data set		UCI	TIMIT-4/6	TIMIT-39	MNIST
Classifier					
NB-ML		83.82	85.04	61.70± 0.22	83.73 ± 0.37
NB-CL		84.10	85.13	61.73± 0.22	83.77 ± 0.37
TAN-CMI-ML		85.00	86.47	65.40 ± 0.2	91.28 ± 0.28
TAN-CMI-CL		85.09	86.48	65.41 ± 0.2	91.28 ± 0.28
TAN-RO-CR-ML	Mean	85.59	87.62	-	-
TAN-RO-CR-ML	Min	85.51	**87.77**	-	-
TAN-RO-CR-ML	Max	85.42	87.60	-	-
TAN-OMI-CR-ML		**85.72**	**87.72**	**66.61** ± 0.21	92.01 ± 0.27
TAN-OMI-CR-CL		**85.74**	**87.73**	**66.62** ± 0.21	92.01 ± 0.27
TAN-OMISP-CR-ML		85.56	87.42	**66.77** ± 0.21	92.10 ± 0.27
TAN-OMISP-CR-CL		85.61	87.42	**66.77** ± 0.21	92.10 ± 0.27
TAN-CR-ML		**85.79**	**87.78**	**66.78** ± 0.21	**92.58** ± 0.26
TAN-CR-CL		**85.78**	**87.78**	**66.78** ± 0.21	**92.58** ± 0.26
2-tree-RO-CR-ML	Mean	-	88.05	-	-
2-tree-RO-CR-ML	Min	-	88.04	-	-
2-tree-RO-CR-ML	Max	-	88.07	-	-
2-tree-OMI-CR-ML		**85.74**	**88.21**	**66.94** ± 0.21	**92.69** ± 0.26
2-tree-OMI-CR-CL		**85.83**	**88.21**	**66.94** ± 0.21	**92.69** ± 0.26

Discriminative parameter learning (CL) produces (most often) a slightly but not significantly better classification performance than ML parameter learning. We use generative parameter training during establishing the discriminative structures of the order-based heuristics or TAN-CR. Once the structure is determined, we use discriminative parameter optimization. It is computationally expensive to perform discriminative parameter learning while optimizing the structure of the network discriminatively.

The TIMIT-39 and MNIST experiments show that we can perform discriminative structure learning for relatively large classification problems (∼140000 samples, 66 features, 39 classes and ∼60000 samples, 196 features, 10 classes). For these data sets, OMI-CR and OMISP-CR significantly outperform NB and TAN-CMI.

On MNIST we achieve a classification performance of $\sim 92.58\%$ with the discriminative TAN classifier. A number of state-of-the-art algorithms (LeCun *et al.* 1998), i.e. convolutional net and virtual SVM, achieve an error rate below 1%. For this reason, we extended our OMI-CR algorithm to learn a discriminative 2-tree with parameter smoothing similar as in (Friedman, Geiger, & Goldszmidt 1997) for regularization. This improves the classification performance to 93.74%. Due to resampling we use only 196 features in contrast to the 784 features of the original data set which might explain the loss in classification rate.

Table 2 and Table 3 present a summary of the classification results over all experiments of the UCI and TIMIT-4/6 data sets. We compare all pairs of classifiers using the one-sided paired t-test. The t-test determines whether the classifiers differ significantly under the assumption that the classification differences over the data set are independent and identically normally distributed. In these tables, each entry gives the significance of the difference in classification rate of two classification approaches. The arrow points to the superior learning algorithm and a double arrow indicates whether the difference is significant at a level of 0.05.

Table 2: Comparison of different classifiers using the one-sided paired t-test for the 25 UCI data sets: Each entry of the table gives the significance of the difference of the classification rate of two classifiers over the data sets. The arrow points to the superior learning algorithm. We use a double arrow if the difference is significant at the level of 0.05.

Classifier	TAN	TAN	TAN	TAN	TAN	2-tree
Struct.Learn.	CMI	RO-CR	OMI-CR	OMISP-CR	CR	OMI-CR
Param.Learn.	ML	ML	ML	ML	ML	ML
		Max				
NB-ML	$\uparrow 0.0977$	$\Uparrow 0.0300$	$\Uparrow 0.0242$	$\Uparrow 0.0371$	$\Uparrow 0.0154$	$\Uparrow 0.0316$
TAN-CMI-ML		$\uparrow 0.120$	$\Uparrow 0.0154$	$\Uparrow 0.0277$	$\Uparrow 0.0140$	$\Uparrow 0.0271$
TAN-RO-CR-ML			$\uparrow 0.144$	$\uparrow 0.184$	$\Uparrow 0.0446$	$\uparrow 0.148$
TAN-OMI-CR-ML				$\leftarrow 0.153$	$\uparrow 0.190$	$\uparrow 0.197$
TAN-OMISP-CR-ML					$\uparrow 0.141$	$\uparrow 0.167$
TAN-CR-ML						$\leftarrow 0.194$

Table 3: Comparison of different classifiers using the one-sided paired t-test for the 12 TIMIT-4/6 data sets: Each entry of the table gives the significance of the difference of the classification rate of two classifiers over the data sets. The arrow points to the superior learning algorithm. We use a double arrow if the difference is significant at the level of 0.05.

Classifier	TAN	2-tree	TAN	TAN	TAN	2-tree
Struct.Learn.	CMI	RO-CR	OMI-CR	OMISP-CR	CR	OMI-CR
Param.Learn.	ML	ML	ML	ML	ML	ML
		Max				
NB-ML	$\Uparrow 0.00181$	$\Uparrow 0.00000277$	$\Uparrow 0.0000189$	$\Uparrow 0.0000294$	$\Uparrow 0.00000360$	$\Uparrow 0.00000237$
TAN-CMI-ML		$\Uparrow 0.000324$	$\Uparrow 0.00159$	$\Uparrow 0.00562$	$\Uparrow 0.00116$	$\Uparrow 0.000185$
2-tree-RO-CR-ML			$\Leftarrow 0.00240$	$\Leftarrow 0.000401$	$\Leftarrow 0.00113$	$\Uparrow 0.00113$
TAN-OMI-CR-ML				$\Leftarrow 0.00568$	$\uparrow 0.140$	$\Uparrow 0.000417$
TAN-OMISP-CR-ML					$\Uparrow 0.000487$	$\Uparrow 0.000149$
TAN-CR-ML						$\Uparrow 0.000154$

These tables show that TAN-OMI-CR, TAN-OMISP-CR, and TAN-CR significantly outperform the generative structure learning approach. However, the naive greedy approach TAN-CR does not significantly outperform our discriminative order-based heuristics, i.e TAN-OMI-CR.

As mentioned in Section 3, generative models can easily deal with missing features simply by marginalizing out from the model the missing feature. We are particularly interested in a testing context which has known, unanticipated at training time, and arbitrary sets of missing features for each classification sample. In such case, it is not possible to re-train the model for each potential set of missing features without also memorizing the training set. Due to the local-normalization property of Bayesian networks and the structure of any model with a parentless class node, marginalization is as easy as an $O(r^{k+1})$ operation for a k-tree, where r is the domain size of each feature.

In Figure 3, we present the classification performance of discriminative and generative structures assuming missing features using the Ma+Fe data of TIMIT-4/6. The x-axis denotes the number of missing features. The curves are the average over 100 classifications of the test data with uniformly at random selected missing features. Variance bars are omitted to improve readability, but indicate that the resulting differences are significant. We use exactly the same missing features for each classifier. We observe that discriminatively structured Bayesian network classifiers outperform TAN-CMI-ML even in the case of missing features. This demonstrates, at least empirically, that discriminative structured generative models do not lose their ability to impute missing features.

The running time of the TAN-CMI, TAN-OMI-CR, and TAN-CR structure learning algorithms for the data sets is summarized in Table 4. The numbers represent the percentage of time that is needed for a particular algorithm compared to TAN-CR. TAN-CMI is roughly 3-10 times faster than TAN-OMI-CR and TAN-CR takes about 10-40 times longer for establishing the structure than TAN-OMI-CR.

Table 4: Running time of structure learning algorithms relative to TAN-CR.

Data	TAN-CMI	TAN-OMI-CR	TAN-CR
UCI	0.649%	3.155%	100.00%
TIMIT-4/6	3.56%	11.47%	100.00%
TIMIT-39	0.11%	2.08%	100.00%
MNIST	0.21%	2.23%	100.00%

6 Conclusion

We introduced a simple order-based heuristic for learning a discriminative network structure. The metric for establishing the ordering of N features is based on either the conditional mutual information or the classification rate. Given an ordering, we can find the discriminative classifier structure using $\mathcal{O}\left(N^q\right)$ score evaluations (where constant q is the maximum number of parents per node).

We empirically compare the performance of our algorithms to state-of-the-art discriminative and generative parameter and structure learning algorithms using real data from the TIMIT speech corpus, the UCI repository, and from a handwritten digit recognition task. The experiments show that the discriminative structures found by our order-based heuristics achieve on average a significantly better classification performance than the generative approach. Our obtained classification performance is very similar to the greedy search using CR. Our order-based heuristics however, are about 10 times faster. Additionally, we show that discriminatively structured Bayesian network classifiers are superior even in the case of missing features.

References

Arnborg, S.; Corneil, D.; and Proskurowski, A. 1987. Complexity of finding embeddings in a k-tree. *SIAM Journal of Algebraic and Discrete Methods* 8(2):277–284.

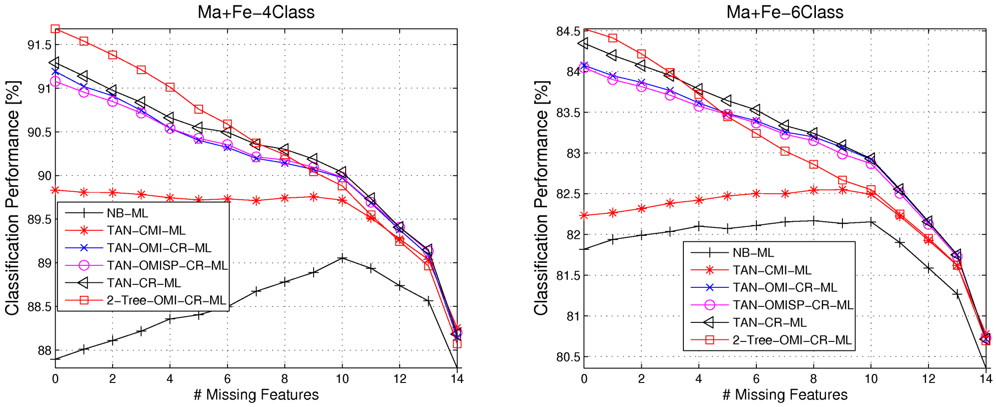

Figure 3: Classification performance assuming missing features using Ma+Fe data of TIMIT-4/6. The x-axis denotes the number of missing features.

Bilmes, J. 1999. *Natural Statistical Models for Automatic Speech Recognition*. Ph.D. Dissertation, U.C. Berkeley.

Bilmes, J. 2000. Dynamic Bayesian multinets. In *16th Inter. Conf. of Uncertainty in Artificial Intelligence (UAI)*, 38–45.

Buntine, W. 1991. Theory refinement on Bayesian networks. In *Proceedings of the Seventh Conference on Uncertainty in AI (UAI)*, 52–60.

Cooper, G., and Herskovits, E. 1992. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning* 9:309–347.

Cover, T., and Thomas, J. 1991. *Elements of information theory*. John Wiley & Sons.

Dasgupta, S. 1997. The sample complexity of learning fixed-structure Bayesian networks. *Machine Learning* 29(2):165–180.

Fayyad, U., and Irani, K. 1993. Multi-interval discretizaton of continuous-valued attributes for classification learning. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 1022–1027.

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

Friedman, N.; Nachman, I.; and Peer, D. 1999. Learning Bayesian network structure form massive datasets: The Sparse Candidate Algorithm. In *Proceedings of the 15th Conference on Uncertainty in AI (UAI)*, 196–205.

Geiger, D., and Heckerman, D. 1996. Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence* 82:45–74.

Greiner, R.; Su, X.; Shen, S.; and Zhou, W. 2005. Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Machine Learning* 59:297–322.

Grossman, D., and Domingos, P. 2004. Learning bayesian network classifiers by maximizing conditional likelihood. In *21st Inter. Conf. of Machine Lerning (ICML)*, 361–368.

Halberstadt, A., and Glass, J. 1997. Heterogeneous measurements for phonetic classification. In *Proceedings of EUROSPEECH*, 401–404.

Keogh, E., and Pazzani, M. 1999. Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *Proceedings of 7th International Workshop on Artificial Intelligence and Statistics*, 225–230.

Kohavi, R., and John, G. 1997. Wrappers for feature subset selection. *Artificial Intelligence* 97:273–324.

Lamel, L.; Kassel, R.; and Seneff, S. 1986. Speech database development: Design and analysis of the acoustic-phonetic corpus. In *Proceedings of the DARPA Speech Recognition Workshop, Report No. SAIC-86/1546*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings fo the IEEE* 86(11):2278–2324.

Meek, C. 1995. Causal inference and causal explanation with background knowledge. In *11th Inter. Conf. on Uncertainty in Artificial Intelligence (UAI'95)*, 403–410.

Merz, C.; Murphy, P.; and Aha, D. 1997. UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, URL: www.ics.uci.edu/~mlearn/MLRepository.html.

Murphy, K. 2002. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD Thesis, University of California, Berkeley.

Narasimhan, N., and Bilmes, J. 2005. A supermodular-submodular procedure with applications to discriminative structure learning. In *21st Inter. Conf. on Uncertainty in Artificial Intelligence (UAI)*.

Ng, A., and Jordan, M. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14*.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann.

Pernkopf, F., and Bilmes, J. 2005. Discriminative versus generative parameter and structure learning of Bayesian network classifiers. In *International Conference on Machine Learning (ICML)*, 657 – 664.

Pernkopf, F., and Bilmes, J. 2007. Discriminative learning for Bayesian network classifiers. Technical report, Graz University of Technology, Department of EE.

Roos, T.; Wettig, H.; Grünwald, P.; Myllymäki, P.; and Tirri, H. 2005. On discriminative Bayesian network classifiers and logistic regression. *Machine Learning* 59:267–296.

Schölkopf, B., and Smola, A. 2001. *Learning with kernels: Support Vector Machines, regularization, optimization, and beyond*. MIT Press.

Teyssier, M., and Koller, D. 2005. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI)*, 584 – 590.