

Differentiable TAN Structure Learning for Bayesian Network Classifiers

Wolfgang Roth

ROTH@TUGRAZ.AT

Franz Pernkopf

PERNKOPF@TUGRAZ.AT

Graz University of Technology, Austria

Signal Processing and Speech Communication Laboratory

Abstract

Learning the structure of Bayesian networks is a difficult combinatorial optimization problem. In this paper, we consider learning of tree-augmented naïve Bayes (TAN) structures for Bayesian network classifiers with discrete input features. Instead of performing a combinatorial optimization over the space of possible graph structures, the proposed method learns a distribution over graph structures. After training, we select the most probable structure of this distribution. This allows for a joint training of the Bayesian network parameters along with its TAN structure using gradient-based optimization. The proposed method is agnostic to the specific loss and only requires that it is differentiable. We perform extensive experiments using a hybrid generative-discriminative loss based on the discriminative probabilistic margin. Our method consistently outperforms random TAN structures and Chow-Liu TAN structures.

Keywords: Bayesian network classifiers; differentiable structure learning; tree-augmented naïve Bayes; hybrid generative-discriminative learning.

1. Introduction

Many existing approaches to Bayesian network (BN) structure learning are scored-based approaches that maximize a score over the combinatorial space of BN graphs \mathcal{G} with respect to a dataset \mathcal{D} . Structure learning is a highly non-trivial task since the number of graphs \mathcal{G} is typically superexponential in the number of variables and score maximization is known to be NP-hard, even for the favorable case of decomposable scores (Chickering et al., 2004).

We consider the learning of BNs with tree-augmented naïve Bayes (TAN) structure. While there exists a polynomial time algorithm to learn TAN structures according to the likelihood score (Friedman et al., 1997), this is not the case for commonly used discriminative criteria. However, discriminative criteria typically yield superior classification performance, which has led to a bulk of literature investigating TAN structure learning with discriminative scores. Grossman and Domingos (2004) and Pernkopf et al. (2011) applied greedy hill climbing to a conditional likelihood score and to a probabilistic margin score, respectively. Pernkopf and Wohlmayr (2013) reported improved results for the probabilistic margin when optimized with simulated annealing. Peharz and Pernkopf (2012) performed margin based structure learning using a general purpose branch and bound algorithm. However, these methods have in common that their discriminative score is based on generative maximum likelihood parameters or, as in (Grossman and Domingos, 2004), they require a time-consuming iterative optimization procedure within the hill climbing loop.

Interestingly, the deep learning community is facing similar challenges when designing the structure of deep neural networks. Whereas deep neural networks used to perform best when de-

signed by an expert, automatic neural architecture search approaches have recently shown to achieve state-of-the-art performance. In particular, our work is inspired by differentiable architecture search approaches (Liu et al., 2019; Cai et al., 2019) which are appealing as they train the parameters and the structure of a deep neural network according to the same discriminative criterion through gradient-based optimization without requiring combinatorial search heuristics.

We propose a differentiable approach for TAN structure learning of BN classifiers. Our method assumes a fixed variable ordering and is based on a relaxation of the discrete graph structure to a *discrete distribution* over graph structures. Given a differentiable loss \mathcal{L} over the BN parameters, we formulate a new structure learning loss $\mathcal{L}_{\text{SL}} = \mathbb{E}[\mathcal{L}]$ as an expected loss with respect to this distribution. Subsequently, the structure loss \mathcal{L}_{SL} is jointly optimized for the BN parameters and the *continuous* distribution parameters using gradient-based learning. After learning, we select the most probable structure. This allows us to apply commonly used discriminative criteria, such as the conditional likelihood or the probabilistic margin (Roth et al., 2018), for structure learning. In fact, the proposed method is agnostic to the specific loss and only requires that it is differentiable.

The proposed method implicitly utilizes properties from stochastic mini-batch optimization to avoid local minima. This is a well-known problem of combinatorial search heuristics for which several techniques have been proposed, such as more elaborate search spaces (Teyssier and Koller, 2005) and perturbation methods (Elidan et al., 2002).

We perform extensive experiments using a hybrid generative-discriminative loss based on the probabilistic margin (Roth et al., 2018). Our method consistently outperforms random TAN structures and likelihood optimized Chow-Liu TAN structures (Friedman et al., 1997) on all evaluated datasets by a large margin. We show that a heuristic variable ordering for image data based on pixel locality further improves performance. Our method does not require combinatorial optimization heuristics and can be easily implemented using modern automatic differentiation frameworks.¹

An interesting orthogonal approach concerning continuous optimization of graph structures has been recently proposed by Zheng et al. (2018). Their work is based on a continuous formulation of the acyclicity constraint for graphs, allowing for gradient-based optimization.

2. Background

Throughout the paper, uppercase symbols X and C refer to random variables, and lowercase variables x and c refer to concrete instantiations of these variables. Similarly, $p(X)$ refers to a distribution, whereas $p(x)$ refers to the probability mass of a concrete instantiation. We denote vectors of random variables and instantiations using boldface symbols \mathbf{X} and \mathbf{x} , respectively.

2.1 Bayesian Networks

Let $\mathbf{X} = \{X_1, \dots, X_D\}$ be a multivariate random variable. A BN is a graphical representation of a probability distribution $p(\mathbf{X})$ that defines a factorization of $p(\mathbf{X})$ via a directed acyclic graph \mathcal{G} containing D nodes, each corresponding to a random variable X_i . Let $\text{pa}(X_i)$ be the set of parents of X_i in \mathcal{G} . Then the graph \mathcal{G} determines the factorization of the joint distribution as

$$p(\mathbf{X}) = \prod_{i=1}^D p(X_i \mid \text{pa}(X_i)). \quad (1)$$

1. Code available online at <https://github.com/wroth8/bnc>

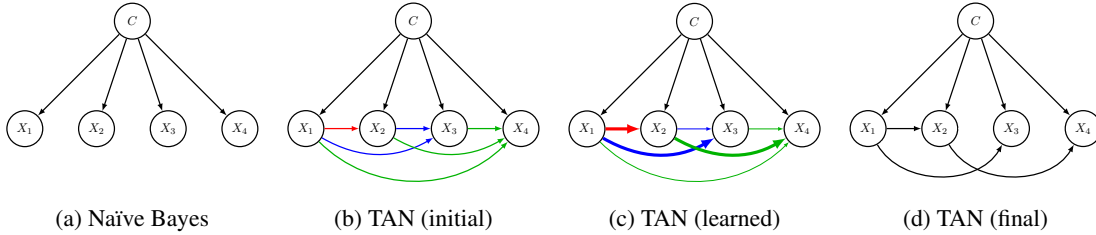


Figure 1: (a) The naïve Bayes model as a BN. (b)-(d) TAN structure learning: We consider every left-to-right edge of the ordered variables as a candidate for the conditioning parents. The TAN structure allows each feature node X_i to have one additional parent besides C . Therefore, each edge is associated with a selection probability indicated by its thickness. (b) Initially, all edges are equally probable. (c) After learning, those edges leading to the best objective value are more probable. (d) The most probable incoming edge of each node is selected as the conditioning parent.

For nodes X_i that do not have any parents in \mathcal{G} , the corresponding factor in (1) is an unconditional distribution $p(X_i)$. The full joint distribution $p(\mathbf{X})$ can now be conveniently specified by the individual factors $p(X_i|\text{pa}(X_i))$. Throughout this paper, we consider distributions over discrete random variables such that each conditional distribution $p(X_i|\text{pa}(X_i))$ can be represented by a conditional probability table (CPT) $\theta_{i|\text{pa}(i)}$, where—with slight abuse of notation— $\text{pa}(i)$ refers to the indices of X_i 's parents. The joint distribution (1) is then determined by the parameters $\theta_{\mathcal{G}} = \{\theta_{1|\text{pa}(1)}, \dots, \theta_{D|\text{pa}(D)}\}$.

For classification tasks, we are given—in addition to the random variables \mathbf{X} which are now called features—another random variable C that takes the role of a class variable. We can then employ a BN to perform classification according to the most probable class c conditioned on \mathbf{x} , i.e.,

$$\underset{c}{\operatorname{argmax}} p(c|\mathbf{x}) = \underset{c}{\operatorname{argmax}} p(\mathbf{x}, c). \quad (2)$$

Assuming that the CPTs $\theta_{\mathcal{G}}$ are given as log-probabilities, we can exploit the factorization (1) to compute (2) efficiently by accumulating only $C \cdot (D + 1)$ values.

However, the size of a CPT $\theta_{i|\text{pa}(i)}$ is given by the number of values that X_i and $\text{pa}(X_i)$ can take jointly. Consequently, assuming that each random variable can take at least two values, the size of X_i 's CPT grows exponentially with the number of parents $|\text{pa}(X_i)|$. Therefore, it is desirable to maintain graph structures where each node has only few parents such that inference tasks remain feasible. In this paper, we consider two particularly simple BN structures, namely the naïve Bayes structure and TAN structures. These structures restrict the number of conditioning parents and, therefore, do not suffer from the exponential growth.

2.2 Naïve Bayes and Tree-augmented Naïve Bayes (TAN) Structures

The naïve Bayes assumption asserts that all input features \mathbf{X} are conditionally independent given the class variable C . In terms of a BN, this can be modeled as a graph \mathcal{G} with a single root node C and all variable nodes X_i having C as their sole parent. This is illustrated in Figure 1a. The

factorization induced by the naïve Bayes assumption is given by

$$p(\mathbf{X}, C) = p(C) \prod_{i=1}^D p(X_i | C). \quad (3)$$

Although this independence assumption rarely holds in practice, naïve Bayes models often perform reasonably well while requiring only few parameters and allowing for fast inference.

To extend the expressiveness of naïve Bayes models, TAN models allow each feature X_i —in addition to the class variable C —to directly depend on one additional feature X_j . This is illustrated in Figure 1d. Consequently, the factorization of a TAN BN is given by

$$p(\mathbf{X}, C) = p(C) \prod_{i=1}^D p(X_i | \text{pa}(X_i)), \quad (4)$$

subject to the constraints $|\text{pa}(X_i)| \leq 2$ and $C \in \text{pa}(X_i)$. This slight relaxation of the graph structure is often sufficient to substantially improve the predictive performance compared to naïve Bayes models (see Section 4). However, as opposed to the naïve Bayes assumption where the corresponding graph \mathcal{G} is fixed, selecting a good TAN structure is non-trivial since the space of TAN graphs grows exponentially in the number of nodes D .

Nevertheless, there exists a polynomial time algorithm to find a maximum likelihood TAN structure (Friedman et al., 1997). This algorithm is an extension of the algorithm proposed by Chow and Liu (1968) to compute maximum likelihood tree structured BNs. The algorithm first computes a maximum spanning tree in a fully-connected graph whose nodes correspond to \mathbf{X} and whose edge weights are determined by the conditional mutual information $I(X_i; X_j | C)$, followed by transforming the resulting undirected tree into a directed tree. We refer to structures discovered by this algorithm as Chow-Liu structures.

However, as reported by several works, generative models tend to perform worse on classification tasks than discriminatively learned models. This suggests that the Chow-Liu structure might be suboptimal for classification. In Section 2.3, we propose a hybrid generative-discriminative training criterion, and in Section 3, we propose our method to jointly learn TAN structures and the CPTs $\theta_{\mathcal{G}}$ according to this criterion by means of gradient-based optimization.

2.3 Hybrid Generative-Discriminative Training

Given a dataset $\mathcal{D} = \{(\mathbf{x}_n, c_n)\}_{n=1}^N$ comprising N samples of input-target pairs, a probabilistic classifier can be trained using a generative or a discriminative approach. A *generative* approach is concerned with modeling the joint distribution $p(\mathbf{X}, C)$. This is typically accomplished by minimizing the negative log-likelihood loss

$$\mathcal{L}_{\text{NLL}}(\theta_{\mathcal{G}}; \mathcal{D}) = - \sum_{n=1}^N \log p(\mathbf{x}_n, c_n). \quad (5)$$

A *discriminative* approach is concerned with modeling the conditional distribution $p(C | \mathbf{X})$ directly. In this work, we consider a discriminative loss based on the notion of a probabilistic margin (Pernkopf et al., 2012; Roth et al., 2018). In particular, we minimize

$$\mathcal{L}_{\text{LM}}(\theta_{\mathcal{G}}; \mathcal{D}) = \sum_{n=1}^N \max(0, \gamma - \beta_n(\theta_{\mathcal{G}})), \quad (6)$$

where $\gamma > 0$ is a desired log-margin hyperparameter and β_n is the probabilistic log-margin of the n^{th} sample defined as

$$\beta_n(\boldsymbol{\theta}_{\mathcal{G}}) = \log \left(\frac{p(c_n | \mathbf{x}_n)}{\max_{c \neq c_n} p(c | \mathbf{x}_n)} \right) = \log p(\mathbf{x}_n, c_n) - \max_{c \neq c_n} \log p(\mathbf{x}_n, c). \quad (7)$$

In our implementation of (7), we employ the softened version of the maximum from (Roth et al., 2018), i.e., $\max(v_1, \dots, v_L) \approx (\log \sum_{i=1}^L \exp(\eta v_i)) / \eta$, where $\eta > 1$ is a hyperparameter.

In practice, the discriminative approach is often superior in terms of classification performance, but often discards most of the probabilistic semantics of the model. This motivates a hybrid generative-discriminative loss to combine the advantages of both approaches as

$$\mathcal{L}_{\text{HYB}}(\boldsymbol{\theta}_{\mathcal{G}}; \mathcal{D}) = \mathcal{L}_{\text{NLL}}(\boldsymbol{\theta}_{\mathcal{G}}; \mathcal{D}) + \lambda \mathcal{L}_{\text{LM}}(\boldsymbol{\theta}_{\mathcal{G}}; \mathcal{D}), \quad (8)$$

where $\lambda > 0$ is a hyperparameter. Previous work has shown that by carefully trading off between the generative and the discriminative loss in (8), most of the probabilistic semantics can be maintained while achieving good accuracy. In many cases a hybrid classifier even outperforms a pure discriminative classifier since the generative term can be seen as a regularizer (Peharz et al., 2013).

3. Differentiable TAN Structure Learning

In this section, we introduce a loss that admits the joint training of the graph \mathcal{G} and the CPTs $\boldsymbol{\theta}_{\mathcal{G}}$. We show how this loss can be trained by means of gradient-based optimization using the reparameterization trick (Kingma and Welling, 2014) and the straight-through gradient estimator (STE) (Bengio et al., 2013), both of which are popular methods in the deep learning community.

3.1 The Structure Learning Loss

Let X_1, \dots, X_D be a fixed ordering of the input features. In our differentiable structure learning approach, for each feature node X_i , we consider every X_j with $j < i$ as a candidate for a conditioning parent. This is illustrated in Figure 1b. Although the search space depends on the particular ordering of \mathbf{X} and does not cover all possible TAN structures, it is convenient as the resulting graph \mathcal{G} is guaranteed to be acyclic.

To enable a joint-treatment of the graph structure \mathcal{G} and the CPT parameters, we reformulate $\log p(\mathbf{X}, C)$ of TAN BNs by introducing new parameters \mathbf{s} governing the graph structure \mathcal{G} . In particular, let $\mathbf{s} = \{\mathbf{s}_2, \dots, \mathbf{s}_D\}$ where $\mathbf{s}_i = (s_{i|1}, \dots, s_{i|i-1})$ is a one-hot vector such that $s_{i|j} = 1$ iff $\text{pa}(X_i) = \{X_j, C\}$.² Furthermore, let $\Theta = \{\boldsymbol{\theta}_C, \boldsymbol{\theta}_1\} \cup \{\boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_D\}$ be the collection of all potential CPTs, where $\boldsymbol{\theta}_C$ is the class prior, $\boldsymbol{\theta}_1$ is the CPT for X_1 , and $\boldsymbol{\theta}_i = \{\boldsymbol{\theta}_{i|1}, \dots, \boldsymbol{\theta}_{i|i-1}\}$ contains the CPTs for all possible conditioning parents of node X_i . Then the log-joint probability for a given structure \mathbf{s} can be expressed as

$$\log p(\mathbf{X}, C) = \log p_{\boldsymbol{\theta}_C}(C) + \log p_{\boldsymbol{\theta}_1}(X_1|C) + \sum_{i=2}^D \sum_{j=1}^{i-1} s_{i|j} \log p_{\boldsymbol{\theta}_{i|j}}(X_i|X_j, C), \quad (9)$$

where we made the dependency on the CPTs Θ explicit in the subscript of p . This allows us to generalize an arbitrary log-likelihood based loss $\mathcal{L}_{\mathcal{G}}(\boldsymbol{\theta}_{\mathcal{G}})$ for a specific graph \mathcal{G} to a loss $\mathcal{L}(\Theta, \mathbf{s})$ including the graph structure \mathbf{s} and to optimize the structure parameters \mathbf{s} and the CPTs Θ jointly.

². This allows us to talk about \mathcal{G} and \mathbf{s} interchangeably.

However, minimizing the combinatorial loss $\mathcal{L}(\Theta, \mathbf{s})$ is problematic as the number of different structures \mathbf{s} scales exponentially in the number of features D . To circumvent the combinatorial nature of $\mathcal{L}(\Theta, \mathbf{s})$, we first introduce *continuous* distribution parameters $\Phi = (\Phi_2, \dots, \Phi_D)$ where $\Phi_i = (\phi_{i|1}, \dots, \phi_{i|i-1})$ and $\sum_{j=1}^{i-1} \phi_{i|j} = 1, \phi_{i|j} \geq 0$. These parameters Φ induce a probability distribution over the one-hot vectors in \mathbf{s} and, consequently, also over the graph structures \mathcal{G} . We can then express a differentiable structure learning loss \mathcal{L}_{SL} as an expectation with respect to the distribution over graph structures as

$$\mathcal{L}_{\text{SL}}(\Theta, \Phi) = \mathbb{E}_{\mathbf{s} \sim p_{\Phi}} [\mathcal{L}(\Theta, \mathbf{s})] = \mathbb{E}_{\mathcal{G} \sim p_{\Phi}} [\mathcal{L}_{\mathcal{G}}(\theta_{\mathcal{G}})]. \quad (10)$$

Here $\theta_{\mathcal{G}}$ are the CPT parameters of Θ required by the graph structure \mathcal{G} . Given an optimal structure \mathbf{s}^* with respect to $\mathcal{L}(\Theta, \mathbf{s})$, an optimal solution to (10) is given by the distribution Φ^* putting all mass on the particular structure, i.e., \mathbf{s}^* itself. Note that the distribution Φ has no particular interpretation and merely serves for the purpose of obtaining a differentiable loss \mathcal{L}_{SL} .

3.2 Minimizing the Structure Learning Loss

The structure loss (10) becomes intractable for a moderate number of features D , but it can be optimized with stochastic gradient descent methods using Monte-Carlo estimates of the gradient of (10). The Monte-Carlo estimates of the gradient are obtained via the reparameterization trick (Kingma and Welling, 2014), which has recently become a popular method for optimizing intractable expectations. The idea of the reparameterization trick is to sample \mathbf{s} by transforming the distribution parameters Φ along with a random sample ε drawn from a *fixed parameter-free* distribution $p(\varepsilon)$ to obtain $\mathbf{s} = g(\Phi, \varepsilon)$. This allows us to compute gradient samples of \mathcal{L}_{SL} with respect to Φ using the backpropagation algorithm.

For a categorical distribution with probabilities Φ_i , we can sample a one-hot encoded vector \mathbf{s}_i by means of a reparameterization using the Gumbel-max trick (Jang et al., 2017; Maddison et al., 2017) according to

$$\mathbf{s}_i = \underset{j}{\operatorname{argmax}} \{ \log \phi_{i|j} + \varepsilon_j \mid j < i \}, \quad (11)$$

where $\varepsilon_j \sim \text{Gumbel}(0, 1)$, and argmax computes a *one-hot* encoding of the maximum argument j .

However, the gradient of $\operatorname{argmax}(\Phi_i)$ is zero almost everywhere and cannot be used for backpropagation. To overcome this, we employ the STE (Bengio et al., 2013). The STE approximates the gradient of a zero-derivative function h during backpropagation with the non-zero gradient of a similar function $\tilde{h} \approx h$, i.e.,

$$\frac{\partial \mathcal{L}}{\partial h} \frac{\partial h}{\partial u} \approx \frac{\partial \mathcal{L}}{\partial h} \frac{\partial \tilde{h}}{\partial u}, \quad (12)$$

which allows us to perform gradient-based learning. In our case, we approximate the gradient of the argmax in (11) during backpropagation by the gradient of a softmax function

$$\operatorname{softmax}_j((\log \Phi_i + \varepsilon)/\tau) = \frac{\exp((\log \phi_{i|j} + \varepsilon_j)/\tau)}{\sum_{j'} \exp((\log \phi_{i|j'} + \varepsilon_{j'})/\tau)}, \quad (13)$$

where $\tau > 0$ is a temperature hyperparameter. For $\tau \rightarrow \infty$ we obtain a uniform distribution and $\tau \rightarrow 0$ recovers the one-hot encoding. This particular sampling procedure is also known as straight-through Gumbel softmax approximation (Jang et al., 2017).

In our approach, the number of possible conditioning parents—and, therefore, also the number of CPTs $\theta_{i|j}$ —is $\mathcal{O}(D^2)$. Note that although most of the structure parameters $s_{i|j}$ in (9) equal zero, we still need to consider every conditional log-probability $\log p_{\theta_{i|j}}(X_i|X_j, C)$ as they are required for the gradient of Φ using the STE. Since this is prohibitive for large D , we propose to consider for each feature node X_i only a fixed randomly selected parent subset $\{X_j \mid j < i\}$ of maximum size $K \ll D$. This results in a linear dependence on the number of features D as $\mathcal{O}(KD)$.

After training has finished, we select the single most probable structure \mathcal{G} and the corresponding CPTs $\theta_{\mathcal{G}}$. We emphasize that the presented method for structure learning is agnostic to the particular loss $\mathcal{L}_{\mathcal{G}}$ and that it can be used in conjunction with any differentiable loss.

4. Experiments

4.1 Datasets

We conducted experiments on the following datasets also used in (Tschitschek et al., 2014).

Letter: This dataset contains 20,000 samples, describing one of 26 English letters using 16 numerical features (statistical moments and edge counts) extracted from images (Dua and Graff, 2019).

Satimage: This dataset contains 6,435 samples containing multi-spectral values of 3×3 pixel neighborhoods in satellite images, resulting in a total of 36 features. The aim is to classify the central pixel of these image patches to one of the seven categories red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, mixture class (all types present), very damp grey soil.

USPS: This dataset contains 11,000 grayscale images of size 16×16 , showing handwritten digits obtained from zip codes of mail envelopes (Hastie et al., 2009). Every pixel is treated as a feature.

MNIST: This dataset contains 70,000 grayscale images showing handwritten digits from 0–9 (LeCun et al., 1998). The original images of size 28×28 are linearly downsampled to 14×14 pixels. Every pixel is treated as a feature.

Except for *satimage*, where we use 5-fold cross-validation, we split each dataset into two thirds of training samples and one third of test samples. The features were discretized using the approach from (Fayyad and Irani, 1993). The average numbers of discrete values per feature are 9.1, 11.5, 3.4, and 13.2 for the respective datasets in the order presented above.

4.2 Experiment Setup

All experiments were performed using the stochastic optimizer Adam (Kingma and Ba, 2015) for 500 epochs. We used mini-batches of size 50 on *satimage*, 100 on *letter* and *usps*, and 250 on *mnist*. Each experiment is performed using the two learning rates $\{3 \cdot 10^{-3}, 3 \cdot 10^{-2}\}$ for the CPTs Θ , and we report the superior result of the two optimization runs. The learning rate is decayed exponentially after each epoch, such that it decreases by a factor of 10^{-3} over the training run. We used a fixed learning rate of 10^{-3} for the structure parameters Φ that is not decayed.

The CPT parameters Θ and the structure parameters Φ are stored as unnormalized log-probabilities. We initialize Θ randomly using a uniform distribution $\mathcal{U}([-0.1, 0.1])$, and we set Φ initially to zero, resulting in a uniform distribution over graphs \mathcal{G} .

We anneal τ in (13) exponentially from 10^1 to 10^{-1} over the training run. This results in a more uniform distribution over structures s at the beginning of training, facilitating exploration of

different structures, whereas the distribution becomes more concentrated at particular structures towards the end of training.

We compared the classification performance of **Naïve Bayes (NB)** and several TAN structures.

TAN Random: TAN structure obtained by a random variable ordering and selecting a random parent X_j for each X_i with $j < i$. We evaluated ten parent selections for five variable orderings, resulting in 50 structures in total.

Chow-Liu: TAN structure obtained using the procedure from Friedman et al. (1997).

TAN Subset (ours): Fixed random variable ordering and randomly selected parent subset of maximum size K satisfying the $j < i$ constraint. We evaluated five parent subsets for five variable orderings, resulting in 25 configurations in total. We selected $K \in \{2, 5, 8\}$.

TAN All (ours): Fixed random variable ordering and considering *all* parents satisfying $j < i$. We evaluated five variable orderings. This setting is only evaluated for the datasets *letter* and *satimage* having fewer features D .

TAN Heuristic (ours): Heuristically determined feature ordering and respective parent subsets based on pixel locality. This setting is evaluated for the image datasets *usps* and *mnist*. We evaluated three ordering heuristics (for details see Section 4.4) and selected $K \in \{1, 2, 5, 8\}$.

We evaluated the same five random variable orderings for TAN Random, TAN Subset, and TAN All. We introduced additional probability parameters $\phi_{i|\emptyset}$ and corresponding CPTs $\theta_{i|\emptyset}$, allowing X_i to also have no additional parent besides C , such that for TAN Subset and TAN Heuristic there are effectively up to $K + 1$ choices for the conditioning parents of X_i . To assess the impact of different K , the parent subsets for smaller K are strict subsets of parent subsets for larger K .

We tuned the hyperparameters of \mathcal{L}_{HYB} using random search in two different settings. In setting (I), we selected 500 hyperparameter configurations according to $\log_{10} \lambda \sim \mathcal{U}([0, 3])$, $\log_{10} \gamma \sim \mathcal{U}([-1, 2])$, and $\eta \sim \mathcal{U}([1, 20])$. In setting (II), we selected 100 hyperparameter configurations according to $\log_{10} \lambda \sim \mathcal{U}([1, 3])$, $\log_{10} \gamma \sim \mathcal{U}([-1, 2])$, and used a fixed $\eta = 10$. We applied (II) to experiments where we evaluated more structural settings, namely TAN Random, TAN Subset, and TAN Heuristic, and we applied setting (I) to all other experiments.

4.3 Classification Results

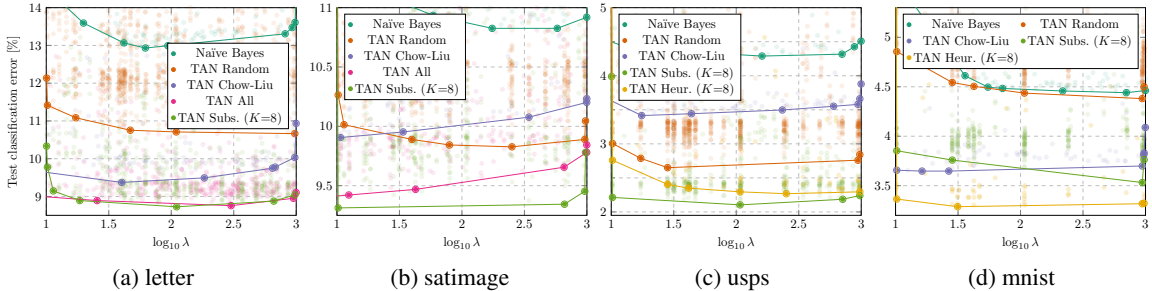
We report the test errors *after* 500 epochs of training. Results for maximum likelihood parameters (ML) are obtained in closed-form. The best classification errors [%] over all parameter settings are shown in Table 1. Results of individual experiments are shown in Figure 2.

Models with generative parameters (ML) perform poorly, showing that discriminative training is beneficial. The TAN structures outperform Naïve Bayes by a large margin, which highlights the benefit of introducing simple TAN interactions, even when they are selected randomly (TAN Random). Note that the data-driven Chow-Liu structure does not outperform TAN Random on all datasets, and there is even a large performance gap on *usps* where we observed overfitting.

Our method (TAN Subset) outperforms TAN Random and Chow-Liu on all datasets. We emphasize that our method outperforms these baseline models on a wide range of settings (cf. Figure 2), and not just using the best setting. TAN Subset achieved its best performance using the largest parent subsets with $K = 8$ on all datasets. Interestingly, TAN All does not benefit when considering *all* conditioning parents compared to TAN Subset with $K = 8$. We attribute this to the fact that

loss \mathcal{L}	\mathcal{L}_{NLL} (ML)		\mathcal{L}_{HYB}			\mathcal{L}_{SL} with \mathcal{L}_{HYB} (ours)		
structure	NB	Chow-Liu	NB	TAN Random	Chow-Liu	TAN Subset	TAN All	TAN Heuristic
letter	25.89	15.23	12.93	10.66	9.37	8.73	8.76	/
satimage	17.95	11.90	10.83	9.83	9.91	9.31	9.39	/
usps	13.11	8.74	4.29	2.65	3.42	2.10	/	2.27
mnist	17.34	7.03	4.44	4.38	3.65	3.53	/	3.29

Table 1: Classification errors [%] of various BN structures on several datasets. See text for details.

Figure 2: Test classification errors [%] of different methods over $\log_{10} \lambda$. Each point corresponds to a different experiment. For better visualization, the lines show the lower part of the convex hull of equally colored points, i.e., no points of this color are below this line.

letter and *satimage* have relatively small numbers of features $D = 16$ and $D = 36$, respectively, and $K = 8$ suffices to cover a good structure with the randomly selected parent subsets with high probability. However, note that using a larger K reduces the gradient updates per CPT since only the gradient of a single CPT $\theta_{i|j}$ for each i is non-zero, resulting in different learning dynamics. Consequently, a different choice of number of epochs or learning rate might yield improved results.

4.4 Heuristic Structures for Image Data

We evaluated three different heuristic feature orderings of quadratically sized images. Feature ordering A considers the pixels row-wise from top to bottom in a left-to-right fashion. Feature ordering B proceeds along the main diagonal by traversing the lower triangular image matrix (including the diagonal) in a row-wise fashion, and including after each pixel the corresponding transposed pixel from the upper triangular matrix. Feature ordering C proceeds from the center of the image outwards. Assuming that a center region of $H \times H$ is already ordered, we add the H pixels directly above and the H pixels directly below in a left-to-right fashion, and then we add the H pixels directly to the left and the H pixels directly to the right in a top-to-bottom fashion. Finally, we also include the four corner pixels in the order left-above, right-above, left-below, and right-below. For each ordering A, B, and C, the subset of at most K conditioning parents of X_i is obtained by selecting $\{X_j | j < i\}$ as the K closest features with respect to Euclidean distance of the corresponding pixel locations.

The heuristic structures show clear improvements on *mnist* (see Table 1 and Figure 2d), and the best result was obtained using ordering C. Overall, when comparing different runs over several random hyperparameter settings in Figure 3a, we found that ordering C slightly outperforms ordering

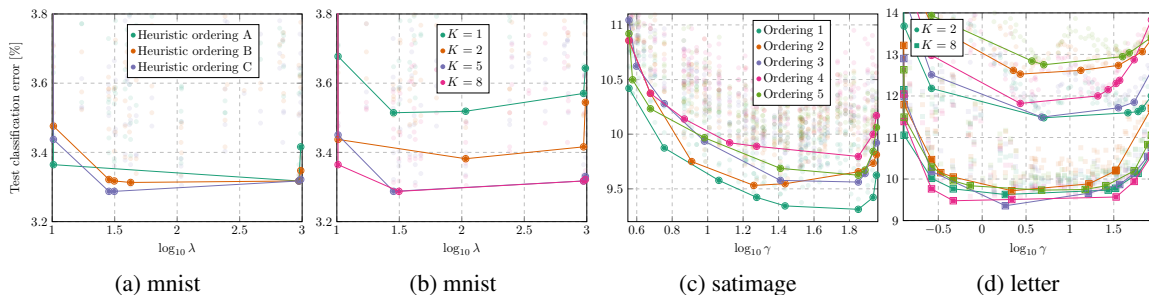


Figure 3: Test classification errors [%] over a hyperparameter. Each point corresponds to a different experiment. For better visualization, the lines show the lower part of the convex hull of equally colored points, i.e., no points of this color are below this line. (a) Results of TAN Heuristic on *mnist* with an emphasis on different heuristic feature orderings. (b) Same results as in (a), but with an emphasis on the maximum number of conditioning parents $K \in \{1, 2, 5, 8\}$. (c) Results of TAN Subset on *satimage* for several feature orderings. (d) Results of TAN Subset on *letter* for several parent subsets (encoded as colors) and $K \in \{2, 8\}$ evaluated for a fixed feature ordering.

B which in turn slightly outperforms ordering A. Note that any of the proposed heuristic orderings outperform the random orderings of TAN Subset.

Figure 3b compares distinct values of K . Overall, using larger parent subsets improves the performance. When comparing $K = 1$ and $K = 2$, we can see that choosing between two neighboring pixels and not just selecting a fixed one improves performance. When going from $K = 5$ to $K = 8$, the accuracy gains are marginal, showing that our method works well for smaller K if the feature ordering and parent subsets are carefully selected. We emphasize that this latter behavior is specific to the heuristic structures, and we generally observed larger gains when going from $K = 5$ to $K = 8$ for TAN Subset.

We did not observe benefits of the heuristic structures on *usps*, which we attribute to overfitting issues similar as to why Chow-Liu performs worse than TAN Random on this dataset.

4.5 Influence of the Feature Ordering and Parent Subsets

The influence of the feature ordering of TAN Subset on *satimage* for $K = 8$ is shown in Figure 3c. We can see that some orderings clearly outperform others over a wide range of parameters, showing that the selected ordering might have a large impact on the overall performance.

To further assess the influence of fixing the feature ordering, we conducted an experiment by allowing conditioning parents that violate the $j < i$ constraint. This results in pseudo TAN structure classifiers which potentially contain cycles and, therefore, are not BNs anymore. The classification errors remained similar on all datasets except *letter* where we achieved 7.98% (0.75% absolute improvement). These findings suggest that more elaborate techniques considering different orderings as in Zheng et al. (2018) are worthy of investigation.

Next, we investigated the influence of particular parent subsets. Figure 3d shows different parent subsets of TAN Subset on *letter* for $K \in \{2, 8\}$. Again, using larger $K = 8$ clearly outperforms the smaller $K = 2$. Similar to feature orderings, some parent subsets clearly outperform others over a wide range of parameters, but here the gap can be reduced by increasing K .

4.6 Recovering the Chow-Liu Structure

We conducted an experiment using the generative loss $\mathcal{L}_{\text{NLL}} (\lambda = 0)$ to see whether our approach is capable of recovering the “ground truth” Chow-Liu structure. Therefore, we computed a Chow-Liu structure and a corresponding ordering such that the Chow-Liu structure is contained in the search space. Our method was able to recover the Chow-Liu structure consistently. Note that this is a minimal requirement of our structure learning approach as the structure learning loss \mathcal{L}_{SL} decomposes to local terms similar as the generative loss \mathcal{L}_{NLL} , substantially simplifying the optimization problem.

5. Conclusion

We have presented an approach to jointly train the parameters of a BN along with its graph structure through gradient-based optimization. The method can be easily implemented using modern automatic differentiation frameworks and does not require combinatorial optimization techniques, such as greedy hill climbing. The presented method is agnostic to the specific loss and can be used with any differentiable loss. We have presented extensive experiments using a hybrid generative-discriminative loss based on the probabilistic margin, showing that our method consistently outperforms random TAN structures and Chow-Liu TAN structures. Our method can be combined with heuristic variable orderings to further improve performance.

The performance of our method depends on the choice of a fixed feature ordering and (for TAN Subset) on the choice of a particular parent subset of maximum size K . Consequently, increasing scalability to larger K and investigating methods not requiring a fixed ordering such as in Zheng et al. (2018) are interesting directions of future research. Furthermore, we believe that the underlying principles of our approach are not restricted to TAN structures and that our approach can be employed to learn other classes of (BN) graph structures.

Acknowledgments

This work was supported by the Austrian Science Fund (FWF) under the project number I2706-N31.

References

- Y. Bengio, N. Léonard, and A. C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.
- H. Cai, L. Zhu, and S. Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations (ICLR)*, 2019.
- D. M. Chickering, D. Heckerman, and C. Meek. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research (JMLR)*, 5:1287–1330, 2004.
- C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.
- D. Dua and C. Graff. UCI machine learning repository, 2019. URL <http://archive.ics.uci.edu/ml>.
- G. Elidan, M. Ninio, N. Friedman, and D. Schuurmans. Data perturbation for escaping local maxima in learning. In *National Conference on Artificial Intelligence (AAAI)*, pages 132–139, 2002.

- U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1022–1027, 1993.
- N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2–3): 131–163, 1997.
- D. Grossman and P. M. Domingos. Learning Bayesian network classifiers by maximizing conditional likelihood. In *International Conference on Machine Learning (ICML)*, volume 69, 2004.
- T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer Series in Statistics. Springer, 2009.
- E. Jang, S. Gu, and B. Poole. Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017.
- D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- H. Liu, K. Simonyan, and Y. Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations (ICLR)*, 2019.
- C. J. Maddison, A. Mnih, and Y. W. Teh. The Concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.
- R. Peharz and F. Pernkopf. Exact maximum margin structure learning of Bayesian networks. In *International Conference on Machine Learning (ICML)*, 2012.
- R. Peharz, S. Tschiatschek, and F. Pernkopf. The most generative maximum margin Bayesian networks. In *International Conference on Machine Learning (ICML)*, volume 28, pages 235–243, 2013.
- F. Pernkopf and M. Wohlmayr. Stochastic margin-based structure learning of Bayesian network classifiers. *Pattern Recognition*, 46(2):464–471, 2013.
- F. Pernkopf, M. Wohlmayr, and M. Mücke. Maximum margin structure learning of Bayesian network classifiers. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2076–2079, 2011.
- F. Pernkopf, M. Wohlmayr, and S. Tschiatschek. Maximum margin Bayesian network classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):521–532, 2012.
- W. Roth, R. Peharz, S. Tschiatschek, and F. Pernkopf. Hybrid generative-discriminative training of Gaussian mixture models. *Pattern Recognition Letters*, 112:131–137, 2018.
- M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 548–549, 2005.
- S. Tschiatschek, K. Paul, and F. Pernkopf. Integer Bayesian network classifiers. In *European Conference on Machine Learning (ECML)*, pages 209–224, 2014.
- X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9492–9503, 2018.