
Discriminative Structure Learning of Bayesian Network Classifiers through Logistic Regression

Ad Feelders
Jevgenijs Ivanovs

AD@CS.UU.NL
JIVANOV@CS.UU.NL

Abstract

We present a new mapping from Bayesian Network Classifiers (BNC) to Logistic Regression (LR) models. It associates with each BNC structure an LR specification with unconstrained parameter space. We prove that a BNC structure and its associated LR specification, index exactly the same set of conditional distributions if and only if the BNC structure has a so-called subperfect independence graph.

The main advantage of our mapping is that it eliminates redundant parameters, thus resulting in an LR specification with a strictly concave log-likelihood function. As a result, discriminative structure learning of BNCs becomes less expensive, because scoring individual structures using the conditional loglikelihood can be performed fairly efficiently.

We illustrate how our theoretical result can be applied to discriminative structure learning by performing experiments with a simple structure learning algorithm that searches in Forest Augmented Naive Bayes (FAN) space.

1. Introduction

The study of Bayesian networks as classifiers has received a considerable impulse by the article of Friedman, Geiger, and Goldszmidt (Friedman et al., 1997) on this subject. They described several extensions of the well-known naive Bayes classifier, and discussed algorithms for learning classifiers from data using the MDL scoring criterion. The likelihood component of this score was based on the *joint* probability of the observed attribute values and class labels in the data. Since the objective is to predict the class label from given attribute values, it has been argued that using the conditional probability of the class labels given the

attribute values might be more appropriate. This scoring criterion has computational disadvantages however, since it does not yield a closed form solution for the maximum likelihood parameter estimates. This means numerical methods have to be applied to score each model, making structure learning on the basis of conditional loglikelihood a rather costly affair.

In this paper we propose a mapping from Bayesian Network Classifiers (BNC) to Logistic Regression (LR) models. We prove that the proposed mapping yields an equivalent LR specification for a large class of BNC structures. The advantage of this mapping is twofold. Firstly, the logistic regression model has fewer parameters than its equivalent BNC counterpart. Secondly, the resulting LR model is known to have a strictly concave loglikelihood function, which means numerical optimization is fairly straightforward and can be performed efficiently. Because the LR specification that corresponds to a given BNC structure according to our mapping can be determined efficiently, it becomes computationally feasible to perform structure learning of BN classifiers using the conditional loglikelihood function.

This paper is structured as follows. In section 2 we introduce the required notation and basic concepts. In section 3 we introduce the proposed mapping by its application to TAN classifiers with binary attributes and class label. Next, we generalize the mapping in section 4 to BN classifiers with discrete (not necessarily binary) attributes and class label. We prove that a BNC structure and its associated LR specification, index exactly the same set of conditional distributions if and only if the BNC structure has a so-called subperfect independence graph. In section 5 we give a short overview of related work. To illustrate the applicability of the theoretical result presented in section 4, we report on experiments with a simple structure learning algorithm in section 6. Finally, we draw some conclusions in section 7.

2. Preliminaries

2.1. Bayesian Networks

We use uppercase letters for random variables and lowercase for their values. Vectors are written in boldface. A Bayesian network (BN) $(\mathbf{X}, G = (K, E), \Theta)$ consists of a discrete random vector $\mathbf{X} = (X_0, \dots, X_n)$, a directed acyclic graph (DAG) G representing the directed independence graph of \mathbf{X} , and a set of conditional probabilities (parameters) Θ . $K = \{0, 1, \dots, n\}$ is the set of nodes of G , and E the set of directed edges. Node i in G corresponds to random variable X_i . With $pa(i)$ ($ch(i)$) we denote the set of parents (children) of node i in G . We write $\mathbf{X}_S, S \subseteq \{0, \dots, n\}$ to denote the projection of random vector \mathbf{X} on components with index in S . The parameter set Θ consists of the conditional probabilities $P(X_i | \mathbf{X}_{pa(i)}), 0 \leq i \leq n$. We use $\mathcal{X}_i = \{0, \dots, d_i - 1\}$ to denote the set of possible values of $X_i, 0 \leq i \leq n$. The set of possible values of random vector \mathbf{X}_S is denoted $\mathcal{X}_S = \times_{i \in S} \mathcal{X}_i$. We also use $\mathcal{X}_i^- = \mathcal{X}_i \setminus \{0\}$.

In a BN classifier there is one distinguished variable called the class variable; the remaining variables are called attributes. We use X_0 to denote the class variable; X_1, \dots, X_n are the attributes. To denote the attributes, we also write \mathbf{X}_A , where $A = \{1, \dots, n\}$. We define $\pi(i) = pa(i) \setminus \{0\}$, the non-class parents of node i , and $\phi(i) = \{i\} \cup \pi(i)$. For a binary variable X , we use \tilde{x} as shorthand for $X = 1$, and \bar{x} as shorthand for $X = 0$.

2.2. Logistic Regression

The basic assumption of logistic regression (Anderson, 1982) for binary class variable $X_0 \in \{0, 1\}$ is

$$\ln \frac{P(\tilde{x}_0 | \mathbf{X}_A)}{P(\bar{x}_0 | \mathbf{X}_A)} = \beta_\emptyset + \sum_{i=1}^k \beta_{\{i\}} Z_i, \quad (1)$$

where the predictors Z_i ($i = 1, \dots, k$) can be single attributes from \mathbf{X}_A , but also functions of one or more attributes from \mathbf{X}_A . In words: the log posterior odds are linear in the parameters, not necessarily in the basic attributes. So for example, the model

$$\ln \frac{P(\tilde{x}_0 | \mathbf{X}_A)}{P(\bar{x}_0 | \mathbf{X}_A)} = \beta_\emptyset + \sum_{i=1}^n \beta_{\{i\}} X_i + \sum_{i \neq j} \beta_{\{i,j\}} X_i X_j, \quad (2)$$

can be written in the form of equation (1) with the $X_i X_j (i \neq j)$ as ‘‘derived’’ variables. In the next section we show that TAN classifiers can be translated to equivalent logistic regression models of the form of (2) with certain elements of β set to zero.

Generalization to non-binary class variable $X_0 \in \mathcal{X}_0$ leads

to the assumption

$$\ln \frac{P(X_0 = j | \mathbf{X}_A)}{P(X_0 = 0 | \mathbf{X}_A)} = \beta_\emptyset(j) + \sum_{i=1}^k \beta_{\{i\}}(j) Z_i, \quad (3)$$

for all $j \in \mathcal{X}_0^-$, which is often referred to as the multinomial logit model or polychotomous logistic regression model.

3. Mapping for TAN classifiers

In this section we introduce our mapping from Bayesian network classifiers to logistic regression models by applying it to TAN classifiers (Friedman et al., 1997) with binary attributes and class variable. Without loss of generality, we assume X_1 is the root of the tree on the attributes. This means X_1 has parent X_0 , and every other attribute $X_i, i \geq 2$ has exactly 2 parents: X_0 and $X_{\pi(i)}$. We write

$$\begin{aligned} \frac{P(\tilde{x}_0 | \mathbf{X}_A)}{P(\bar{x}_0 | \mathbf{X}_A)} &= \frac{P(\tilde{x}_0) P(X_1 | \tilde{x}_0) \prod_{i=2}^n P(X_i | X_{\pi(i)}, \tilde{x}_0)}{P(\bar{x}_0) P(X_1 | \bar{x}_0) \prod_{i=2}^n P(X_i | X_{\pi(i)}, \bar{x}_0)} \\ &= \frac{P(\tilde{x}_0)}{P(\bar{x}_0)} \frac{P(\tilde{x}_1 | \tilde{x}_0)^{(1-X_1)} P(\tilde{x}_1 | \tilde{x}_0)^{X_1}}{P(\bar{x}_1 | \bar{x}_0)} \\ &\prod_{i=2}^n \left(\frac{P(\tilde{x}_i | \tilde{x}_{\pi(i)}, \tilde{x}_0)^{X_i X_{\pi(i)}} P(\tilde{x}_i | \tilde{x}_{\pi(i)}, \tilde{x}_0)^{X_i (1-X_{\pi(i)})}}{P(\tilde{x}_i | \tilde{x}_{\pi(i)}, \bar{x}_0)} \right) \\ &\frac{P(\bar{x}_i | \tilde{x}_{\pi(i)}, \tilde{x}_0)^{(1-X_i) X_{\pi(i)}} P(\bar{x}_i | \tilde{x}_{\pi(i)}, \tilde{x}_0)^{(1-X_i)(1-X_{\pi(i)})}}{P(\bar{x}_i | \tilde{x}_{\pi(i)}, \bar{x}_0)} \end{aligned}$$

By taking the log on both sides, followed by expanding and collecting terms, we get an expression of the form

$$\ln \frac{P(\tilde{x}_0 | \mathbf{X}_A)}{P(\bar{x}_0 | \mathbf{X}_A)} = \beta_\emptyset + \sum_{i=1}^n \beta_{\{i\}} X_i + \sum_{i=2}^n \beta_{\{i, \pi(i)\}} X_i X_{\pi(i)} \quad (4)$$

where the β 's are functions of the TAN parameters. For example

$$\beta_{\{i, \pi(i)\}} = \ln \frac{\text{cpr}(X_i, X_{\pi(i)} | \tilde{x}_0)}{\text{cpr}(X_i, X_{\pi(i)} | \bar{x}_0)},$$

where $\text{cpr}(X_i, X_{\pi(i)} | x_0)$ denotes the conditional cross-product ratio between X_i and $X_{\pi(i)}$ given $X_0 = x_0$. Hence, $\beta_{\{i, \pi(i)\}}$ measures the difference in log cross-product ratios between X_i and $X_{\pi(i)}$ within the two classes. If the cross-product ratio between each attribute and its parent happens to be the same within both classes, then the interaction terms drop out in (4), and in fact the LR model that is linear in the basic attributes is exactly correct.

We can see from equation (4), that a given TAN model maps to a logistic regression model of the form (2) with $\beta_{\{i,j\}} = 0$ unless there is an edge between attributes X_i and X_j in the graph.

Clearly, the mapping is only valid when none of the TAN parameters equals zero, which amounts to the common restriction that $P(\mathbf{X})$ must be strictly positive. Notice that the logistic regression model has only $1 + n + (n - 1) = 2n$ parameters, while the TAN has $1 + 2 + 4(n - 1) = 4n - 1$ parameters. To show the equivalence of the TAN and corresponding logistic regression model, we show how to derive TAN parameters from arbitrary logistic regression parameters $\beta \in \mathbb{R}^{2n}$, in such a way that the TAN yields the same conditional distribution as the logistic regression model. We start with processing attributes that have no children, and work our way back towards the root X_1 . When processing $X_i, i \geq 2$, we determine the values of the TAN parameters $P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0), P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0), P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)$, and $P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)$. As a consequence of the processing order, we have the following equations involving these parameters:

$$\beta_{\{i\}} = c + \ln \frac{P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)}{P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)} - \ln \frac{1 - P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)}{1 - P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)}, \quad (5)$$

where the constant c is a result of values of parameters with X_i in the conditioning set, that were determined in previous steps, and

$$\begin{aligned} \beta_{\{i, \pi(i)\}} &= \ln \frac{P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)}{P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)} - \ln \frac{1 - P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)}{1 - P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)} \\ &+ \ln \frac{P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)}{P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)} - \ln \frac{1 - P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)}{1 - P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0)} \end{aligned} \quad (6)$$

We fix $P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0) = \alpha_{i1}$, where $\alpha_{i1} \in (0, 1)$, and find from (5):

$$P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0) = \frac{\alpha_{i1}}{\alpha_{i1} + (1 - \alpha_{i1}) \exp(\beta_{\{i\}} - c)}$$

Since $\alpha_{i1} > 0$, and $1 - \alpha_{i1} > 0$, we have that $P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0) \in (0, 1)$. Next we fix $P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0) = \alpha_{i2}$, where $\alpha_{i2} \in (0, 1)$, and find from (6):

$$P(\tilde{x}_i|\tilde{x}_{\pi(i)}, \tilde{x}_0) = \frac{\alpha_{i2}}{\alpha_{i2} + (1 - \alpha_{i2}) \exp(\beta_{\{i, \pi(i)\}} - c')}$$

After processing $X_i, i \geq 2$ we process the root X_1 . At that point we have

$$\beta_{\{1\}} = c + \ln \frac{P(\tilde{x}_1|\tilde{x}_0)}{P(\tilde{x}_1|\tilde{x}_0)} - \ln \frac{1 - P(\tilde{x}_1|\tilde{x}_0)}{1 - P(\tilde{x}_1|\tilde{x}_0)},$$

and by choosing $P(\tilde{x}_1|\tilde{x}_0) = \alpha_1 \in (0, 1)$ we can solve for $P(\tilde{x}_1|\tilde{x}_0)$ as before. Finally, we get

$$\beta_{\emptyset} = c + \ln \frac{P(\tilde{x}_0)}{1 - P(\tilde{x}_0)}, \text{ so } P(\tilde{x}_0) = \frac{\exp(\beta_{\emptyset} - c)}{1 + \exp(\beta_{\emptyset} - c)}$$

Notice that we indeed fixed $2(n - 1) + 1 = 2n - 1$ parameters. Since we can fix these parameters at arbitrary values between 0 and 1, there are obviously infinitely many TAN parameter values that yield the same conditional distribution as given parameter values of the corresponding logistic regression model. By using the proposed mapping, we can score a given TAN model by optimizing a strictly concave function of $2n$ parameters, instead of a non-concave function (Roos et al., 2005) of $4n - 1$ parameters.

4. Generalization of the Mapping

In this section we generalize the mapping introduced in section 3 to Bayesian network classifiers with discrete valued attributes and class variable, and arbitrary independence graph. We prove that a BNC structure and its associated LR specification, index exactly the same set of conditional distributions if (section 4.2) and only if (section 4.3) the BNC structure has a so-called subperfect independence graph.

We use two common assumptions throughout. Firstly, we assume that all conditional probabilities exist and are strictly positive. Secondly, we consider only graphs restricted to the Markov blanket of X_0 , as other random variables can be dropped from the conditioning set.

4.1. The general mapping

Exploiting the factorisation of the joint probability according to G , we can write

$$\begin{aligned} \ln \frac{P(\tilde{x}_0|\mathbf{X}_A)}{P(\tilde{x}_0|\mathbf{X}_A)} &= \ln \frac{P(\tilde{x}_0|\mathbf{X}_{pa(0)})}{P(\tilde{x}_0|\mathbf{X}_{pa(0)})} + \sum_{i \in ch(0)} \ln \frac{P(X_i|\mathbf{X}_{\pi(i)}\tilde{x}_0)}{P(X_i|\mathbf{X}_{\pi(i)}\tilde{x}_0)} \\ &= \sum_{\mathbf{x} \in \mathcal{X}_{pa(0)}} \ln \frac{P(\tilde{x}_0|\mathbf{x})}{P(\tilde{x}_0|\mathbf{x})} \prod_{j \in pa(0)} I(X_j = x_j) + \\ &\sum_{i \in ch(0)} \sum_{\mathbf{x} \in \mathcal{X}_{\phi(i)}} \ln \frac{P(x_i|\mathbf{x}_{\pi(i)}\tilde{x}_0)}{P(x_i|\mathbf{x}_{\pi(i)}\tilde{x}_0)} \prod_{j \in \phi(i)} I(X_j = x_j), \end{aligned} \quad (7)$$

where $I(\cdot)$ is the indicator function taking value 1 if its argument is true and 0 otherwise. For each $1 \leq i \leq n$ and for each $a \in \mathcal{X}_i^-$ we define a binary logistic regression variable $I_i^a = I(X_i = a)$. Analogous to the binary case, we exclude I_i^0 from the model, in order to avoid the creation of linear dependencies between the predictors. We have

$$I(X_i = x_i) = \begin{cases} I_i^{x_i} & \text{if } 1 \leq x_i \leq d_i - 1 \\ 1 - \sum_{a=1}^{d_i-1} I_i^a & \text{if } x_i = 0 \end{cases} \quad (8)$$

We combine this with equation (7) to get a logistic regression model with variables I_i^a . Based on DAG G , we define a family \mathcal{S} of index sets as follows:

$$\mathcal{S} = \{S | S \subseteq pa(0)\} \cup \{S | S \subseteq \phi(i), i \in ch(0)\} \quad (9)$$

Furthermore, let $F(S)$ be the set of all valid non-zero assignments to variables with index in S , that is $F(S) = \{f : S \rightarrow \cup_{i \in S} \mathcal{X}_i^- \mid \forall i \in S : f(i) \in \mathcal{X}_i^-\}$. The logistic regression specification corresponding to G is then given by:

$$\ln \frac{P(\tilde{x}_0 | \mathbf{X}_A)}{P(\bar{x}_0 | \mathbf{X}_A)} = \sum_{\substack{S \subseteq \mathcal{S} \\ f \in F(S)}} \beta_S^f \prod_{i \in S} I_i^{f(i)} \quad (10)$$

In case of a non-binary class variable $X_0 \in \{0, \dots, d_0 - 1\}$ the logistic regression model is defined by $d_0 - 1$ vectors of parameters $\beta(a)$, $1 \leq a \leq d_0 - 1$, as stated in equation (3). Similarly to equation (10) we get for all $a \in \mathcal{X}_0^-$:

$$\ln \frac{P(X_0 = a | \mathbf{X}_A)}{P(X_0 = 0 | \mathbf{X}_A)} = \sum_{\substack{S \subseteq \mathcal{S} \\ f \in F(S)}} \beta_S^f(a) \prod_{i \in S} I_i^{f(i)} \quad (11)$$

4.2. Equivalence for subperfect independence graphs

A directed graph in which all nodes that have a common child are connected is called perfect.

Definition 1. We call a directed acyclic graph subperfect, if it is perfect, except that some arcs between parents of the class variable may be missing.

We assumed G is restricted to the Markov blanket of X_0 , thus subperfectness of G implies that every attribute is connected to X_0 .

Consider a subperfect graph G and the corresponding LRS \mathcal{S} . Let $M^{BN(G)}$ be the set of all conditional distributions modeled by all possible BN classifiers having independence graph G . Let $M^{L(S)}$ be the set of all conditional distributions modeled by all possible logistic regression models restricted by \mathcal{S} . In the following we will prove the main theorem:

Theorem 2. $M^{BN(G)} = M^{L(S)}$ if and only if graph G is subperfect.

The general idea to prove the if part is to show that for subperfect graph G and corresponding set \mathcal{S} , we can find for any value of the vector $\beta_S^f (S \subseteq \mathcal{S}, f \in F(S))$, values of the parameters $\theta_{X_i | \mathbf{X}_{pa(i)}} (i \in \{0\} \cup ch(0))$ that yield the same conditional distribution. The proof is constructive: we give an algorithm to determine θ from β . The algorithm makes use of the observations in Lemma 3 and Lemma 4.

Lemma 3. a) Let $i \in ch(0)$, and let $S \subseteq \phi(i)$, $f \in F(S)$, where S is such that $i \in S$, then

$$\beta_S^f = \sum_{\substack{\mathbf{x} \in \mathcal{X}_{\phi(i)}: \\ x_j \in \{f(j), 0\}, j \in S \\ x_j = 0, j \in \phi(i) \setminus S}} (-1)^{|\{j \in S \mid x_j = 0\}|} \ln \frac{P(x_i | \mathbf{X}_{\pi(i)} \tilde{x}_0)}{P(x_i | \mathbf{X}_{\pi(i)} \bar{x}_0)} + g(i), \quad (12)$$

where $g(i)$ depends only on $P(X_j | \mathbf{X}_{pa(j)})$, $j \in ch(0)$, $i \in pa(j)$.

b) Let $S \subseteq pa(0)$ and $f \in F(S)$, then

$$\beta_S^f = \sum_{\substack{\mathbf{x} \in \mathcal{X}_{pa(0)}: \\ x_j \in \{f(j), 0\}, j \in S \\ x_j = 0, j \in pa(0) \setminus S}} (-1)^{|\{j \in S \mid x_j = 0\}|} \ln \frac{P(\tilde{x}_0 | \mathbf{x})}{P(\bar{x}_0 | \mathbf{x})} + g(0), \quad (13)$$

where $g(0)$ depends only on $P(X_i | \mathbf{X}_{pa(i)})$, $i \in ch(0)$.

Proof: a) From (7), (8) and (10) it follows that β_S^f is a sum of terms $\pm \ln \frac{p}{q}$, where p, q are conditional probabilities of X_j , $j \in ch(0)$ and q is obtained from p by substitution of \tilde{x}_0 with \bar{x}_0 , moreover p, q contain X_i . If X_i is in the conditioning set of p and thus q , then the term $\pm \ln \frac{p}{q}$ just contributes to $g(i)$. Now we consider only the terms $\pm \ln \frac{P(X_i | \mathbf{X}_{\pi(i)} \tilde{x}_0)}{P(X_i | \mathbf{X}_{\pi(i)} \bar{x}_0)}$. To be consistent with f we have to have $x_j \in \{f(j), 0\}$, $j \in \phi(i)$. Clearly, $x_j = 0$ if $j \in \phi(i) \setminus S$, otherwise S would contain j . With each $x_j = 0$, $j \in S$ comes a minus sign from $I(X_j = 0) = 1 - \sum_{a=1}^{d_j-1} I_j^a$, so each term has sign $(-1)^{|\{j \in S \mid x_j = 0\}|}$.

b) The assumption of the lemma immediately restricts our attention to terms associated with conditional probabilities of X_0 . Our further reasoning is similar to part a). \square

Lemma 4. The system of equations and constraints

$$\begin{aligned} \beta_a &= \ln \frac{p_a}{q_a} - \ln \frac{p_0}{q_0} + c_a, \text{ for } 1 \leq a \leq d-1 \\ 0 &< p_a, q_a < 1, \text{ for } 0 \leq a \leq d-1 \\ \sum_{a=0}^{d-1} p_a &= 1, \sum_{a=0}^{d-1} q_a = 1 \end{aligned} \quad (14)$$

has only solutions that satisfy

$$\begin{aligned} p_a &= \frac{q_a \exp(\beta_a - c_a)}{q_0 + \sum_{j=1}^{d-1} q_j \exp(\beta_j - c_j)}, \text{ for } 1 \leq a \leq d-1 \\ p_0 &= \frac{q_0}{q_0 + \sum_{j=1}^{d-1} q_j \exp(\beta_j - c_j)}, \end{aligned} \quad (15)$$

where we can choose q_0, \dots, q_{d-1} freely as long as they satisfy the constraints expressed in (14).

Proof: We have $\frac{p_a q_0}{q_a p_0} = \exp(\beta_a - c_a)$, $1 \leq a \leq d-1$. Using $p_0 = 1 - \sum_{j=1}^{d-1} p_j$ we get $p_a(1 + \frac{q_a \exp(\beta_a - c_a)}{q_0}) + \sum_{j \in \{1, \dots, d-1\} \setminus \{a\}} p_j \frac{q_a \exp(\beta_a - c_a)}{q_0} = \frac{q_a \exp(\beta_a - c_a)}{q_0}$, which is a linear system of $d-1$ equalities in p_a , $1 \leq a \leq d-1$. It is easy to see that the equations are linearly independent and thus have a unique solution. One can verify that $p_a = \frac{q_a \exp(\beta_a - c_a)}{q_0 + \sum_{j=1}^{d-1} q_j \exp(\beta_j - c_j)}$. We compute

Input: DAG G , vector of β_S^f , where $S \in \mathcal{S}$, $f \in F(S)$.

Output: vector of $\theta_{X_i|\mathbf{x}_{pa(i)}}$, where $i \in \{0\} \cup ch(0)$.

(1)

init(topological-order)

processed:= \emptyset

repeat $|ch(0)|$ times

$i := \text{index-of-last}(\text{topological-order})$

for $l := 0$ to $|\pi(i)|$ do

for all $W \subseteq \pi(i) \& |W| = l$ do

for all $f : \forall j \in W, f(j) \in \mathcal{X}_j^-$ do $/(*)$

evaluatefor($\beta_{W \cup \{i\}}^{f \cup \{(i,1)\}}, \dots, \beta_{W \cup \{i\}}^{f \cup \{(i,d_i-1)\}}$)

processed:=processed $\cup \{X_i\}$

drop-last(topological-order)

(2)

for $l := 0$ to $|pa(0)|$ do

for all $W \subseteq pa(0) \& |W| = l$ do

for all $f : \forall j \in W, f(j) \in \mathcal{X}_j^-$ do $/(*)$

evaluatefor0(β_W^f)

Figure 1. Algorithm for evaluation of BN parameters

$p_0 = \frac{q_0}{q_0 + \sum_{j=1}^{d-1} q_j \exp(\beta_j - c_j)}$. We conclude with noticing that when $q_a > 0$, for $0 \leq a \leq d-1$, then all numbers in each fraction are positive and the numerator is contained in the denominator, thus $0 < p_a < 1$, for $0 \leq a \leq d-1$. \square

Algorithm 1 takes as input an arbitrary DAG G , and a vector of values of β_S^f ($S \in \mathcal{S}$, $f \in F(S)$), and returns values for $\theta_{X_i|\mathbf{x}_{pa(i)}}$ ($i \in \{0\} \cup ch(0)$). First, let us consider part (1) of the algorithm. In the first step we sort the children of X_0 in topological order. In the main loop we go through all indices of children of X_0 in reverse topological order (first children then parents). In step (*) we just iterate for all possible valid non-zero assignments to variables with index in W . The function evaluatefor($\beta_{W \cup \{i\}}^{f \cup \{(i,1)\}}, \dots, \beta_{W \cup \{i\}}^{f \cup \{(i,d_i-1)\}}$) evaluates the following $2 \times d_i$ BN parameters: $\theta_{X_i|\mathbf{x}_{\pi(i)}\tilde{x}_0}, \theta_{X_i|\mathbf{x}_{\pi(i)}\bar{x}_0}$, where $\mathbf{x}_{\pi(i)}$ is such that $x_j = f(j)$ for $j \in W$, and $x_j = 0$ otherwise. So the set W and function f determine the configuration of $\mathbf{X}_{\pi(i)}$.

Note, that the processing order of vertices guarantees that all children of X_i , which are also children of X_0 , have already been processed and therefore all conditional probabilities of $X_j, j \in ch(0)$ with X_i in the conditioning set have already been evaluated. We apply lemma 3(a) for each of $\beta_{W \cup \{i\}}^{f \cup \{(i,a)\}}, 1 \leq a \leq d_i - 1$. The algorithm guarantees that conditional probabilities of X_i , specified by $W' \subset W$ ($|W'| < |W|$) and all possible assignments on W' have been evaluated. Therefore any term in equation (12), containing $\mathbf{x} \in \mathcal{X}_{\pi(i)}, \exists j \in W : x_j = 0$, was evaluated in some step associated with $W' \subseteq W \setminus \{j\}$ and thus

contributes to the constant. So we have

$$\begin{aligned} \beta_{W \cup \{i\}}^{f \cup \{(i,a)\}} &= \\ &\sum_{\substack{\mathbf{x} \in \mathcal{X}_{\phi(i)}: \\ x_j = f(j), j \in W \\ x_j = 0, j \in \pi(i) \setminus W \\ x_i \in \{0, a\}}} (-1)^{|\{j \in W \cup \{i\} | x_j = 0\}|} \ln \frac{P(x_i | \mathbf{x}_{\pi(i)} \tilde{x}_0)}{P(x_i | \mathbf{x}_{\pi(i)} \bar{x}_0)} + c_a \\ &= \ln \frac{P(X_i = a | \mathbf{x}_{\pi(i)} \tilde{x}_0)}{P(X_i = a | \mathbf{x}_{\pi(i)} \bar{x}_0)} - \ln \frac{P(X_i = 0 | \mathbf{x}_{\pi(i)} \tilde{x}_0)}{P(X_i = 0 | \mathbf{x}_{\pi(i)} \bar{x}_0)} + c_a, \end{aligned} \quad (16)$$

for $1 \leq a \leq d_i - 1$, where $x_j = f(j)$ if $j \in W$, and $x_j = 0$ if $j \in \pi(i) \setminus W$.

Evaluation is a procedure for determining a solution to this system of equations. We use lemma 4 to do this. Note that we choose a solution from an infinitely big set.

Next we consider part (2) of the algorithm. The function evaluatefor0(β_W^f) evaluates the following 2 BN parameters: $\theta_{\tilde{x}_0|\mathbf{x}_{pa(0)}}, \theta_{\bar{x}_0|\mathbf{x}_{pa(0)}}$, where $\mathbf{x}_{pa(0)}$ is such that $x_j = f(j)$ for $j \in W$, and $x_j = 0$ otherwise. Note that the assumption of lemma 3(b) is satisfied. Again, the algorithm guarantees that the conditional probabilities of X_0 , specified by $W' \subset W$ ($|W'| < |W|$) and all possible assignments on W' have been evaluated. Therefore we have

$$\beta_W^f = \ln \frac{P(\tilde{x}_0 | \mathbf{x}_{pa(0)})}{P(\bar{x}_0 | \mathbf{x}_{pa(0)})} + c, \quad (17)$$

where $x_j = f(j)$ for $j \in W$, and $x_j = 0$ for $j \in pa(0) \setminus W$. We solve it to: $P(\tilde{x}_0 | \mathbf{x}_{pa(0)}) = \frac{\exp(\beta_W^f - c)}{1 + \exp(\beta_W^f - c)}$ and $P(\bar{x}_0 | \mathbf{x}_{pa(0)}) = \frac{1}{1 + \exp(\beta_W^f - c)}$. Again $0 < P(\tilde{x}_0 | \mathbf{x}_{pa(0)}), P(\bar{x}_0 | \mathbf{x}_{pa(0)}) < 1$. \square

This concludes our discussion of algorithm 1.

Lemma 5. Let G be a DAG and let the vector with components $\beta_S^f, S \in \mathcal{S}, f \in F(S)$ be a vector of parameters of a logistic regression model. We apply algorithm 1 and get $\theta_{X_i|\mathbf{x}_{pa(i)}}, i \in \{0\} \cup ch(0)$ - a subset of the BN classifier parameters. Take any BN classifier on graph G consistent with this subset of parameters and map it to a logistic regression model obtaining parameters β_S^f , then $\beta_S^f = \beta_S^f$ for all $S \in \mathcal{S}'$ and all valid f defined on S , where

$$\mathcal{S}' = \{S \cup \{i\} | i \in ch(0), S \subseteq \pi(i)\} \cup \{S | S \subseteq pa(0)\} \quad (18)$$

Obviously $\mathcal{S}' \subseteq \mathcal{S}$. We say that \mathcal{S}' defines the set of β_S^f used by the algorithm.

Proof: Note that in equation (7) we use only $P(X_i | \mathbf{x}_{pa(i)}), i \in \{0\} \cup ch(0)$. To show that \mathcal{S}' , indeed, is such as stated above, consider all β_S^f used in algorithm 1.

For each $i \in ch(0)$ we solve equations for each β_S^f , where $S = \{i\} \cup W, W \subseteq \pi(i)$ and $f \in F(S)$. When $i = 0$ we solve for each β_S^f , where $S \subseteq pa(0)$ and $f \in F(S)$. \square

Lemma 6. *Suppose we are given a BN classifier on DAG G with vector of parameters θ having components $\theta_{X_i|\mathbf{x}_{pa(i)}}, 0 \leq i \leq n$. If application of our mapping returns vector β of logistic regression parameters, then running algorithm 1 on G and β and making the correct choices at the evaluatefor step will provide us with $\theta_{X_i|\mathbf{x}_{pa(i)}}, i \in \{0\} \cup ch(0)$ consistent with θ .*

Proof: The algorithm only determines the order of evaluation. It does not restrict the set of possible solutions (see lemma 4). \square

Lemma 7. *If graph G is subperfect then $S' = S$.*

Proof: This follows from the fact that $\{S|i \in ch(0), S \subseteq \phi(i)\} \subseteq \{S \cup \{i\}|i \in ch(0), S \subseteq \pi(i)\} \cup \{S|S \subseteq pa(0)\}$.

Theorem 8. *If DAG G is subperfect and S is its associated family of index sets, then $M^{BN(G)} = M^{L(S)}$.*

Proof: Our mapping is defined on the set of all BN classifiers, therefore $M^{BN(G)} \subseteq M^{L(S)}$. Left to prove that $M^{L(S)} \subseteq M^{BN(G)}$. Suppose we are given a vector of β_S^f , which agrees with logistic regression specification S . We apply algorithm 1 and find $\theta_{X_i|\mathbf{x}_{pa(i)}}, i \in \{0\} \cup ch(0)$. We assign values to the other classifier parameters, which, of course, we can achieve. Lemma 5 states that the mapping applied to the classifier we have got will produce the same logistic regression model with regard to all β_S^f , where $S \in S'$, and f is a valid assignment defined on S . We are left to prove that these are all possible parameters, that is $S' = S$ for subperfect graph G . Here we use lemma 7. So we have constructed a BN classifier from a logistic regression model. This classifier maps back to this model and thus represents the same conditional distribution of the class variable. \square

4.3. Necessity of subperfectness

In this section we show that subperfectness of G is a necessary condition for equivalence. In the proof we use the fact that, in our parameterization, different logistic regression models index different conditional distributions of X_0 .

Lemma 9. *Let us consider DAG G and vector β of parameters of logistic regression model. Suppose algorithm 1 is applied to G and β to calculate $\theta_{X_i|\mathbf{x}_{pa(i)}}, i \in \{0\} \cup ch(0)$, then $\forall i \in ch(0), \mathbf{x} \in \mathcal{X}_{\phi(i)} : \min_{x_i|\mathbf{x}_{\pi(i)}} \leq \ln \frac{P(x_i|\mathbf{x}_{\pi(i)}\tilde{x}_0)}{P(x_i|\mathbf{x}_{\pi(i)}\bar{x}_0)} \leq \max_{x_i|\mathbf{x}_{\pi(i)}}$, where $\min_{x_i|\mathbf{x}_{\pi(i)}}, \max_{x_i|\mathbf{x}_{\pi(i)}} \in \mathbb{R}$ and depend only on β_S^f used before and including the step in which $P(x_i|\mathbf{x}_{\pi(i)}\tilde{x}_0), P(x_i|\mathbf{x}_{\pi(i)}\bar{x}_0)$ were evaluated.*

Proof: We use induction. Consider equation 16. The constant (on the first step it is 0) is a sum of bounded terms (see equation 7), where the bounds depend on already used β_S^f , so all $d_i - 1$ following terms b_a are bounded, where the bounds depend on already used and current $\beta_{\{i\} \cup W}^{\{(i,a)\} \cup f}, 1 \leq i \leq d_i - 1$:

$$b_a = \ln \frac{p_a}{q_a} - \ln \frac{p_0}{q_0}, \quad 1 \leq a \leq d_i - 1, \quad (19)$$

where

$$p_c = P(X_i = c|\mathbf{x}_{\pi(i)}\tilde{x}_0), q_c = P(X_i = c|\mathbf{x}_{\pi(i)}\bar{x}_0),$$

for $0 \leq c \leq d_i - 1$, and where $x_j = f(j)$ if $j \in W$ and $x_j = 0$ if $j \in \pi(i) \setminus W$.

We use lemma 4 and find that $\frac{p_a}{q_a} = \frac{\exp b_a}{\sum_{j=0}^{d_i-1} q_j \exp b_j}, 1 \leq a \leq d_i - 1$ if we define $b_0 = 0$. To conclude we need to show that $0 < \min_a \leq \frac{\exp b_a}{\sum_{j=0}^{d_i-1} q_j \exp b_j} \leq \max_a$ or $0 < \min \leq \sum_{j=0}^{d_i-1} q_j \exp b_j \leq \max$. We know that $\exp b_j > 0$ and $\sum q_j = 1, q_j > 0$, so $0 < \min_{j=0}^{d_i-1} (\exp b_j) \leq \sum_{j=0}^{d_i-1} q_j \exp b_j \leq \max_{j=0}^{d_i-1} (\exp b_j) \square$

Lemma 10. *If graph G is not subperfect then $S \setminus S'$ is not empty. It contains some set S , that is not contained in $pa(0)$.*

Proof: This follows from the fact that, since G is not subperfect, it must contain some $X_j, j \in ch(0)$ that has two unconnected parents X_a, X_b , which are not both parents of X_0 . \square

Theorem 11. *If DAG G is not subperfect, then $M^{BN(G)} \neq M^{L(S)}$.*

Proof: In our parameterization, different logistic regression models index different conditional distributions of X_0 . Therefore, if $M^{BN(G)} = M^{L(S)}$ then for any value of β there should be a value of θ , which is mapped back to that same value of β . Lemma 6 states that using algorithm 1 we can determine values $\theta_{X_i|\mathbf{x}_{pa(i)}}, i \in \{0\} \cup ch(0)$ of θ . Lemma 9 states that $\ln \frac{P(X_i|\mathbf{x}_{\pi(i)}\tilde{x}_0)}{P(X_i|\mathbf{x}_{\pi(i)}\bar{x}_0)}, i \in ch(0)$ is bounded by numbers dependent on β_S^f used in the algorithm, that is $S \in S'$. Lemma 10 states that there exists a parameter $\beta_{S^*}^f (S^* \in S \setminus S')$ of the logistic regression model, that was not used by the algorithm. This means that changing the value of this parameter will not affect the bounds. It follows from the definition of our mapping that this $\beta_{S^*}^f (S^*$ is not contained in $pa(0))$ can be expressed as a sum of bounded terms $\pm \ln \frac{P(X_i|\mathbf{x}_{\pi(i)}\tilde{x}_0)}{P(X_i|\mathbf{x}_{\pi(i)}\bar{x}_0)}, i \in ch(0)$. So, for equivalence to hold, the value of $\beta_{S^*}^f$ should be contained in a certain interval determined by $\beta_S^f (S \in S')$. To conclude, we pick a value of $\beta_{S^*}^f$ outside this interval. \square

4.4. Non-binary class variables

Algorithm 1 needs to be changed only slightly to handle the case of a non-binary class variable. In part (1) the function `evaluatefor` should be called for each $a \in \mathcal{X}_0^-$. In the first call we would be able to choose a solution to the associated equations from an infinitely big set. We would fix probabilities $P(X_i | \mathbf{x}_{\pi(i)} X_0 = 0)$ and obtain $P(X_i | \mathbf{x}_{\pi(i)} X_0 = 1)$. For other $a \neq 1$ we will have probabilities $P(X_i | \mathbf{x}_{\pi(i)} X_0 = a)$ uniquely determined. Similarly in part (2) of the algorithm we should call function `evaluatefor0` for each $a \in \mathcal{X}_0^-$. We get equations

$$\beta_W^f(a) = \ln \frac{P(X_0 = a | \mathbf{x}_{pa(0)})}{P(X_0 = 0 | \mathbf{x}_{pa(0)})} + c_a, \quad a \in \mathcal{X}_0^- \quad (20)$$

which has a unique solution $P(X_0 = a | \mathbf{x}_{pa(0)}) = \frac{\exp(\beta_W^f(a) - c_a)}{\sum_{i=0}^{d_0-1} \exp(\beta_W^f(i) - c_i)}$ (put $\beta(0) = \mathbf{0}$ and $c_0 = 0$). Clearly, this conditional distribution is valid.

We apply exactly the same reasoning as in the binary class variable case to arrive at the proof of the main theorem for the general case: $M^{BN(G)} = M^{L(S)}$ if and only if graph G is subperfect.

5. Related Work

Greiner et al. (Greiner et al., 2005) propose an algorithm called ELR for discriminative structure learning of Bayesian network classifiers. ELR parameterizes a Bayesian network as a system of logistic regression equations. It uses gradient ascent to find local maxima of the likelihood function. ELR is both more general (it can handle incomplete data and arbitrary network structures) and less efficient than our approach. Experiments demonstrate that ELR often works better than generative methods, also in the incomplete data case (Greiner et al., 2005).

Grossman and Domingos (Grossman & Domingos, 2004), propose a simple heuristic for efficient structure learning of Bayesian network classifiers: parameters are fitted using the generative estimates and the structure is chosen on the basis of conditional loglikelihood. They show experimentally that this simple heuristic works quite well on a large collection of datasets.

Our work is most closely related to that of Roos et al. (Roos et al., 2005). They propose a mapping from BN classifiers to logistic regression models and prove equivalence for the class of subperfect independence graphs. Their mapping, however, yields an overparameterized logistic regression model with a likelihood function that is concave, but not *strictly* concave. In the mapping we propose, the redundant parameters are removed. More specifically, the parameters obtained in our mapping are sums of parameters obtained in the mapping proposed by Roos et al. (Roos et al., 2005).

Moreover, one can prove that the LR specification obtained by application of their mapping models exactly the same set of conditional distributions as the LR specification obtained by our mapping. This implies that subperfectness of G is a necessary condition for their mapping as well, which provides a partial answer to what was left as an open problem in (Roos et al., 2005).

6. Experiments

To illustrate the applicability of our mapping, we present a simple hill-climbing structure learning algorithm based on conditional loglikelihood scoring. The algorithm searches in the space of so-called Forest Augmented Networks (FANs) (Lucas, 2004). The search starts with a naive Bayes model that includes all attributes. In each step edges may be added or removed, with the following restrictions: (1) An edge from X_0 to X_i can only be removed if X_i is not connected to any other attribute; (2) An undirected edge can be added between attributes X_i and X_j only if they are both connected to X_0 , and the edge does not create a cycle on the attributes.

During search, edges between attributes are treated as undirected edges. To get a directed graph on the attributes, simply direct the edges in such a way that no cycles or V-structures are created.

In a first experiment, we sample data from the network G depicted in the left part of figure 2, where all variables are binary. The logistic regression specification corresponding to G is given by

$$\ln \frac{P(\tilde{x}_0)}{P(\bar{x}_0)} = \beta_\emptyset + \beta_{\{1\}} X_1 + \beta_{\{2\}} X_2 + \beta_{\{3\}} X_3 + \beta_{\{4\}} X_4 + \beta_{\{5\}} X_5 + \beta_{\{1,2\}} X_1 X_2 + \beta_{\{1,3\}} X_1 X_3 + \beta_{\{3,4\}} X_3 X_4 \quad (21)$$

Since G is subperfect, this logistic regression specification is equivalent to it, in the sense that it indexes the same set of conditional distributions of X_0 as G does. Furthermore, note that this conditional distribution is in FAN-space; the equivalent FAN is depicted in the right part of figure 2. We specified four different sets of parameter values for G , differing in the strength of the three-way interactions between (X_0, X_1, X_2) , (X_0, X_1, X_3) and (X_0, X_3, X_4) . For each interaction strength, we drew samples of different sizes and fitted a naive Bayes and FAN model on this sample. The FAN model was selected using the hill-climbing algorithm described above, where models were scored on the basis of AIC, i.e. conditional loglikelihood – number of parameters. The accuracy of each fitted model was computed on a test sample of size 10,000. For each sample size/interaction strength combination the experiment was replicated 20 times; then we computed the difference between the average accuracy of FAN and NB (see table 1).

| | 50 | 100 | 500 | 1000 |
|--------|-------|-------|-------|-------|
| None | -3.5 | -0.9 | 0.0 | 0.0 |
| Weak | -0.6 | -0.8 | +0.4 | +0.5 |
| Mild | +0.9 | +1.1 | +4.8 | +4.1 |
| Strong | +19.7 | +19.2 | +19.6 | +19.7 |

Table 1. Difference in percentage correctly predicted between FAN and NB for different sample sizes and interaction strengths

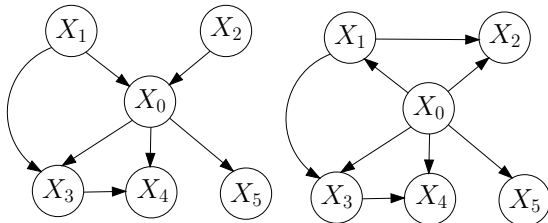


Figure 2. Example network G (left); equivalent FAN (right).

Models were fitted in R using the `multinom` function from the `nnet` package.

In the no interaction case, where the conditional naive Bayes assumption is exactly correct (i.e. all two-way interactions drop out from (21)), the FAN does worse for small sample sizes, due to overfitting, but at sample size 500 it equalizes. At the other extreme, in the case of strong interaction the FAN scores much better than NB, even for small sample size. In the more realistic inbetween cases, the FAN goes from slightly worse to slightly better, or from slightly better to clearly better as the sample size increases.

In a second experiment we consider a somewhat larger problem with 15 attributes (not all of them binary) and ternary class variable. The Markov blanket of the class variable contains only 7 of the 15 attributes. We specified mild three-way interactions involving the class variable. The results are given in table 2. Again, for each sample size 20 replications were performed, and the accuracy was computed on a test sample with 10,000 observations. In this experiment, we included pruned naive Bayes in the comparison, since 8 irrelevant attributes are included. The pruning was done with a simple hill-climbing strategy: starting from the full naive Bayes model, attributes were removed until no improvement of the AIC-score was possible. The experiment was performed on a Pentium 4 machine with 2.8GHz CPU.

Again, we observe that for small sample sizes FANs are worse than NB and pruned NB, but for larger sample sizes FANs have higher accuracy. Overall, pruned NB is only slightly better than NB, confirming NB’s resistance against overfitting. Times to select a model are rounded to the nearest second and averaged over 20 replications. For pNB and

| | NB | pNB | FAN |
|------|--------------------------|--------------------------|----------------------------|
| 50 | 49.8 \pm 4.0(0) | 48.7 \pm 5.4(2) | 46.9 \pm 4.5(15) |
| 100 | 55.9 \pm 1.8(0) | 57.5 \pm 2.6(4) | 52.3 \pm 3.5(33) |
| 500 | 63.3 \pm 0.7(0) | 64.2 \pm 0.9(14) | 65.6 \pm 1.0(279) |
| 1000 | 64.4 \pm 0.5(1) | 64.8 \pm 0.8(25) | 68.3 \pm 0.8(456) |

Table 2. Percentage correctly predicted \pm standard error for NB, pruned NB and FAN for different sample sizes. Time to select a model (averaged over 20 replications) rounded to the nearest second is given between brackets.

FAN these times are for the whole selection process; for example, the FAN algorithm has to score $\binom{15}{2} + 15 = 120$ models in the first step.

7. Conclusion

We have proposed a mapping that translates BNC structures with subperfect independence graphs to equivalent LR specifications. The LR model has less parameters and a strictly concave likelihood function which allows for efficient numerical optimization. This result is a step towards efficient discriminative structure learning of BN classifiers. This is supported by our experiments with a structure learning algorithm that searches in the space of Forest Augmented Naive Bayes (FAN) models.

References

- Anderson, J. A. (1982). Logistic discrimination. In P. R. Krishnaiah and L. N. Kanal (Eds.), *Classification, pattern recognition and reduction of dimensionality*, vol. 2 of *Handbook of Statistics*, 169–191. North-Holland.
- Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29, 131–163.
- Greiner, R., Xiaoyuan, S., Shen, B., & Zhou, W. (2005). Structural extension to logistic regression: Discriminative parameter learning of belief net classifiers. *Machine Learning*, 59, 297–322.
- Grossman, D., & Domingos, P. (2004). Learning Bayesian network classifiers by maximizing conditional likelihood. *Proceedings of the 21st International Conference on Machine Learning* (pp. 361–368). Omnipress.
- Lucas, P. (2004). Restricted bayesian network structure learning. *Advances in Bayesian Networks, Studies in Fuzziness and Soft Computing* (pp. 217–232). Springer.
- Roos, T., Wettig, H., Grünwald, P., Myllymäki, P., & Tirri, H. (2005). On discriminative Bayesian network classifiers and logistic regression. *Machine Learning*, 59, 267–296.