International Journal of Approximate Reasoning 161 (2023) 108990

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Structural learning of mixed noisy-OR Bayesian networks ☆

Jiří Vomlel [a,*], Václav Kratochvíl [a], František Kratochvíl [b]

[a] Institute of Information Theory and Automation, Czech Academy of Sciences, Pod Vodárenskou věží 4, Prague 8, 182 00, Czechia
[b] Department of Asian Studies, Palacký University Olomouc, třída Svobody 26, Olomouc, 779 00, Czechia

ARTICLE INFO

Article history:
Received 30 January 2023
Received in revised form 22 June 2023
Accepted 19 July 2023
Available online 26 July 2023

Keywords:
Bayesian networks
Learning Bayesian networks
Noisy-OR model
Applications of Bayesian networks
Linguistics
Loanwords

ABSTRACT

In this paper we discuss learning Bayesian networks whose conditional probability tables are either Noisy-OR models or general conditional probability tables. We refer to these models as Mixed Noisy-OR Bayesian Networks. To learn their structure, we modify the Bayesian Information Criterion used for standard Bayesian networks to reflect the number of parameters of a Noisy-OR model. We prove that the log-likelihood function of a Noisy-OR model has a unique maximum and adapt the EM-learning method for the leaky Noisy-OR model. We propose a structure learning algorithm that learns optimal Mixed Noisy-OR Bayesian Networks. We evaluate the proposed approach on synthetic data, where it performs substantially better than standard Bayesian networks. We perform experiments with Bipartite Noisy-OR Bayesian networks of different complexity to find out when the results of Mixed Noisy-OR Bayesian Networks are significantly better than the results of standard Bayesian networks and when they perform similarly. We also study how different penalties based on the number of model parameters affect the quality of the results. Finally, we apply the suggested approach to a problem from the domain of linguistics. Specifically, we use Mixed Noisy-OR Bayesian Networks to model the spread of loanwords in the South-East Asian Archipelago. We perform numerical experiments in which we compare the prediction ability of standard Bayesian networks with Mixed Noisy-OR Bayesian networks and test different pruning methods to reduce the number of parent sets considered.

1. Introduction

Bayesian networks (BNs) [12,20] are popular models for problems with uncertainty. Learning the structure of BNs from data is a well-studied problem with many interesting results [3,6,27]. Since it is well known that the structure learning problem is NP-hard [4], optimal learning can only be performed for smaller models. However, sophisticated learning methods such as those proposed by [6]) keep shifting the tractability border for optimal learning. BNs with a certain local structure of their conditional probability tables (CPTs) [8] represent a special subclass of Bayesian networks well applicable to many real-world problems.

Much less attention has been paid to learning the structure of such Bayesian networks [10]. A commonly used model of CPTs with a local structure is the Noisy-OR model [9,20,28]. This model has found its way to several applications of Bayesian networks due to its natural interpretation and low number of parameters, which is linear in the number of variables in the

☆ This work was supported by the Czech Science Foundation Project Nr. 20-18407S.
* Corresponding author.
E-mail addresses: vomlel@utia.cas.cz (J. Vomlel), velorex@utia.cas.cz (V. Kratochvíl), frantisek.kratochvil@upol.cz (F. Kratochvíl).

https://doi.org/10.1016/j.ijar.2023.108990
0888-613X/

corresponding conditional probability table. Several authors use the Noisy-OR assumption in learning the CPTs from data, e.g., [18,19]. These papers assume the structure is known and all CPTs are represented by Noisy-OR models. [29] analyzed several existing Bayesian networks that they knew were not constructed using the Noisy-OR/MAX assumption and investigated how well the CPTs in these networks matched the Noisy-OR/MAX distribution. Their result is that roughly half of all nodes with parents can be approximated using the Noisy-OR/MAX distribution.

In this paper, we study the problem of learning a BN structure where each CPT can be represented by either a general CPT or a Noisy-OR model, choosing the alternative that leads to a better resulting Bayesian network model. We refer to these models as Mixed Noisy-OR Bayesian networks. We compare this approach with the standard BN structure learning and apply both methods to the problem of modeling the spread of loanwords in the South-East Asian Archipelago.

Our work is most closely related to [25], where the authors also consider structural learning of Mixed Noisy-OR Bayesian Networks. However, our work differs in several aspects. We work with leaky Noisy-OR models, i.e., Noisy-OR models extended by a leak probability first proposed by [11]. We use an EM learning method adapted from [28] to learn parameters of Noisy-OR models. Additionally, we prove the log-likelihood function of a Noisy-OR model has a unique maximum. To reduce the complexity of optimal model search we study and implement different pruning methods for Mixed Noisy-OR Bayesian Networks. Finally, we use different datasets for experimental evaluations and perform extensive systematic tests with them.

In this paper we extend our previous work published in the proceedings of the PGM 2022 conference [15]. We have extended the paper with a number of additional experiments. Namely:

- We performed additional experiments with models of different complexity. For low complexity models, Mixed Noisy-OR BNs and standard BNs provide similar results, while for more complex models, Mixed Noisy-OR BNs perform significantly better.
- We also confirmed that Mixed Noisy-OR BNs and Standard BNs are close when the number of free parameters is similar.
- We performed additional experiments with different penalizations instead of (log n)/2, which is used for BIC. We observed that when there is relatively enough data with respect to model complexity, a rather large range of penalties leads to similar results. However, the situation is reversed when the training data sets are small or the models are complex. In this case, for Mixed Noisy-OR BNs, the best results are obtained with BIC, while for standard BNs, the results are significantly worse, but the best results are obtained with penalties similar to AIC.
- Also in the case of complete BN2O networks (BNs with a bipartite graph structure, where all children have all top-layer nodes as parents). Mixed Noisy-OR BNs perform significantly better than standard BNs.
- We have added new results on pruning methods to reduce the number of parent sets considered.

The paper is organized as follows. Section 2 describes the Bayesian Information Criterion (BIC) for Noisy-OR Bayesian networks. We show that the log-likelihood function of Noisy-OR has a unique maximum. We propose an EM-algorithm for Noisy-OR parameter learning. In Section 3, we describe a structure learning algorithm to decide which type of CPT will be used for each node. We discuss pruning methods that can be applied when evaluating parent sets before they are used as input to the GOBNILP algorithm. In Section 4, we report the results of experiments designed to verify that the proposed algorithm can identify Noisy-OR models correctly. We also compare the predictive performance of the learned model with the model learned by maximizing BIC score without considering Noisy-OR models. For these experiments, we used a class of BN models common in practical applications – BNs consisting of two layers of nodes with CPTs of all nodes from the second layer being Noisy-OR models. We study how the quality of the learned model is influenced by the model complexity, training data size, and by different penalties based on the number of model parameters. In Section 5, we apply Mixed Noisy-OR Bayesian networks to a problem from linguistic analysis - we use Mixed Noisy-OR BNs to model the spread of loanwords in the South-East Asia Archipelago. This section also discusses pruning methods for learning optimal Mixed Noisy-OR BNs. Even though our existing pruning proposals have led to good models, it turns out that none of the pruning methods proposed so far guarantee optimal Mixed Noisy-OR BNs. We conclude the paper by summarizing our work and suggesting future research topics.

2. Bayesian information criterion for mixed noisy-OR Bayesian networks

Let $V = \{1, \ldots, n\}$ be the set of indexes of random variables X_v, $v \in V$, each taking states x_v from a finite set $\mathcal{X}_v$. In this paper, all variables will be assumed to be Boolean, taking states *false* and *true* represented by numerical values 0 and 1, respectively. It means that $\mathcal{X}_v = \{0, 1\}$ for all $v \in V$.

Assume a Bayesian network model representing a joint probability distribution P that assigns a probability value $P(\mathbf{x})$ to each possible realization $\mathbf{x} = (x_1, \ldots, x_n)$ of multidimensional variable $\mathbf{X} = (X_1, \ldots, X_n)$, i.e. $P : \{0, 1\}^n \to [0, 1]$ and $\sum_{\mathbf{x} \in \{0,1\}^n} P(\mathbf{x}) = 1$. The structure of the Bayesian network is defined by an acyclic directed graph G, which defines a set-valued function $pa(v)$ giving parent nodes of node v in graph G – a node u is a parent node of node v if there is an edge $u \to v$ in G. When it is clear from the context, we will interchangeably refer to nodes or their corresponding variables.

Let $\mathbf{D}$ be a set of data vectors $\mathbf{x} = (x_1, \ldots, x_n)$, i.e., the set of realizations of variables $\mathbf{X} = (X_1, \ldots, X_n)$. In the text, we will use small boldface letters $\mathbf{x}_A$ to denote a configuration of a multidimensional variable $\mathbf{X}_A$ where A is a subset of indexes V. In case $A = \{v\} \cup U$ for $U \subset V$ we will abbreviate $\mathbf{X}_{\{v\} \cup U}$ as $\mathbf{X}_{v,U}$. For any $A \subseteq V$ the symbol $\mathbf{x}_A$ denotes a

Table 1
An example of a CPT $P(X_v|X_u, X_w)$ representing a Noisy-OR model.

	$X_u=0$ $X_w=0$	$X_u=0$ $X_w=1$	$X_u=1$ $X_w=0$	$X_u=1$ $X_w=1$
$X_v=0$	$p_{v,0}$	$p_{v,0}\cdot p_{v,w}$	$p_{v,0}\cdot p_{v,u}$	$p_{v,0}\cdot p_{v,u}\cdot p_{v,w}$
$X_v=1$	$1-p_{v,0}$	$1-p_{v,0}\cdot p_{v,w}$	$1-p_{v,0}\cdot p_{v,u}$	$1-p_{v,0}\cdot p_{v,u}\cdot p_{v,w}$

realization of a multidimensional random variable X_A, $\mathbf{x}_A \in \mathcal{X}_A = \{0,1\}^{|A|}$, and it refers to the subvector of vector $\mathbf{x} = \mathbf{x}_V$ restricted to values of variables $X_i, i \in A$. If the set A is a one-element set, i.e., $A = \{a\}$, then we denote the realization of one-dimensional variable X_a as x_a, and it refers to the value of vector $\mathbf{x}$ corresponding to variable X_a.

The probability of observing i.i.d. data $\mathbf{D}$ given a Bayesian network model P is:

$$L(P|\mathbf{D}) = \prod_{\mathbf{x}\in\mathbf{D}} P(\mathbf{x}) \tag{1}$$

$$= \prod_{\mathbf{x}\in\mathbf{D}} \prod_{v\in V} P(x_v|\mathbf{x}_{pa(v)}) \ . \tag{2}$$

It is referred to as the likelihood of a model with respect to data $\mathbf{D}$. Assume $A \subseteq V$, then the function $N : \mathcal{X}_A \to \mathbb{N}$ provides the number of occurrences of $\mathbf{x}_A \in \mathcal{X}_A = \times_{a\in A}\mathcal{X}_a$ in data $\mathbf{D}$ and $fa(v) = \{v\} \cup pa(v)$ denotes the family of v. The logarithm of the likelihood, abbreviated as log-likelihood, can be decomposed:

$$LL(P|\mathbf{D}) = \log \prod_{\mathbf{x}\in\mathbf{D}} P(\mathbf{x}) \ = \ \sum_{\mathbf{x}\in\mathbf{D}} \sum_{v\in V} \log P(x_v|\mathbf{x}_{pa(v)}) \tag{3}$$

$$= \sum_{v\in V} \sum_{\mathbf{x}\in\{0,1\}^n} N(\mathbf{x}) \cdot \log P(x_v|\mathbf{x}_{pa(v)}) \tag{4}$$

$$= \sum_{v\in V} LL_v(P|\mathbf{D}) \ , \text{ where} \tag{5}$$

$$LL_v(P|\mathbf{D}) = \sum_{\mathbf{x}_{fa(v)}\in\{0,1\}^{|fa(v)|}} N(\mathbf{x}_{fa(v)}) \cdot \log P(x_v|\mathbf{x}_{pa(v)}) \ . \tag{6}$$

This means that the log-likelihood of a Bayesian network can be computed locally, i.e., it is a sum of local functions $LL_v(P|\mathbf{D})$, $v \in V$ computed for each node v and its parents $pa(v)$, which together form a family $fa(v)$.

A specific CPT type is the Noisy-OR model. The CPT of the Noisy-OR model of a variable $X_v, v \in V$ is defined for $\mathbf{x}_{v,pa(v)} \in \{0,1\}^J$, $J = 1 + |pa(v)|$ as

$$P(x_v|\mathbf{x}_{pa(v)}) = \begin{cases} p_{v,0}\cdot\prod_{j\in pa(v)}(p_{v,j})^{x_j} & \text{if } x_v = 0 \\ 1 - p_{v,0}\cdot\prod_{j\in pa(v)}(p_{v,j})^{x_j} & \text{if } x_v = 1, \end{cases} \tag{7}$$

where $p_{v,j}$ represents the probability that the positive influence of parent X_j on its child X_v is inhibited. Please note that x_j is the exponent (not an upper index) and takes value 0 or 1. Therefore, if $x_j = 0$ then $(p_{v,j})^{x_j} = 1$ and if $x_j = 1$ then $(p_{v,j})^{x_j} = p_{v,j}$. The parameter $p_{v,0}$ is called the leak probability, and the value $1 - p_{v,0}$ specifies the probability that variable X_v takes value 1 although all its parents have value 0. In Table 1, we present an example of a CPT of a variable X_v with two parents X_u and X_w specified by a Noisy-OR model.

Formula (7) can also be written in the following form, which is convenient for deriving the log-likelihood function.

$$P(x_v|\mathbf{x}_{pa(v)}) = \left(p_{v,0}\cdot\prod_{j\in pa(v)}(p_{v,j})^{x_j}\right)^{(1-x_v)} \cdot \left(1 - p_{v,0}\cdot\prod_{j\in pa(v)}(p_{v,j})^{x_j}\right)^{(x_v)} \ . \tag{8}$$

The local log-likelihood score[1] of a Noisy-OR model of $P(X_v|X_{pa(v)})$ can be written as

$$LL_v^{\diamond}(P|\mathbf{D}) = \sum_{\mathbf{x}_{fa(v)}\in\{0,1\}^{|fa(v)|}} N(\mathbf{x}_{fa(v)}) \cdot \left(\begin{array}{l} (1-x_v)\cdot\left(\log p_{v,0} + \sum_{j\in pa(v)} x_j \log p_{v,j}\right) \\ +x_v \cdot \log\left(1 - p_{v,0}\cdot\prod_{j\in pa(v)}(p_{v,j})^{x_j}\right) \end{array} \right). \tag{9}$$

[1] We will use the diamond symbol $\diamond$ to denote the scores of a Noisy-OR.

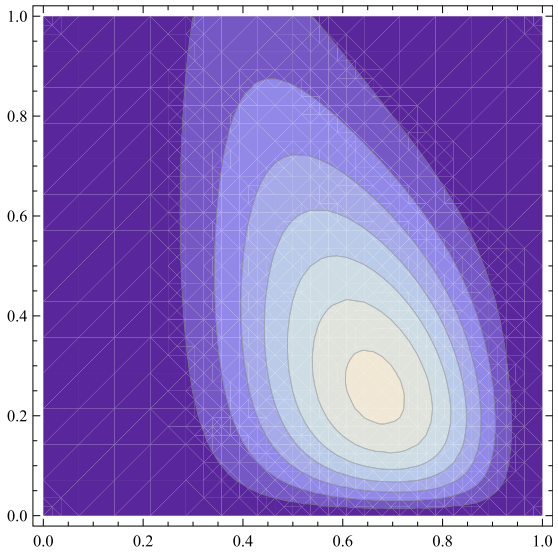

Fig. 1. The contour plot of a likelihood function of a Noisy-OR.

It is well known that for a Bayesian network with a given graph structure, the conditional probability distributions P^* that maximize the log-likelihood $LL(P|\mathbf{D})$ can be computed from data $\mathbf{D}$ as relative frequencies, which means that for $(x_v, \mathbf{x}_{pa(v)})$

$$P^*(x_v|pa(v)) = \frac{N(\mathbf{x}_{fa(v)})}{N(\mathbf{x}_{pa(v)})} \ . \tag{10}$$

In the case of Noisy-OR, no closed-form solution for the conditional probability distributions P^* maximizing the log-likelihood is known. However, due to the decomposability of the log-likelihood, optimal parameters can still be computed locally for each node $v \in V$. In the next lemma, we claim that the local log-likelihood score of a Noisy-OR is strictly concave.

Lemma 1. *The local log-likelihood score of a Noisy-OR $LL_v^\diamond(P|\mathbf{D})$ is a strictly concave function of its parameters $p_{v,0}$ and $p_{v,j}$, $j \in pa(v)$.*

Proof. We will check the terms of (9). Function $\log p_{v,j}$ is a strictly concave function of $p_{v,j}$ for $v \in V$ and $j \in \{0\} \cup pa(v)$. Function

$$\log\left(1 - p_{v,0} \cdot \prod_{j \in pa(v)} (p_{v,j})^{x_j}\right)$$

is a strictly concave function of $p_{v,j}$ for $v \in V$ and $j \in \{0\} \cup pa(v)$. The sum of strictly concave functions is itself strictly concave.

A strictly concave function has a unique maximum. See Fig. 1 for an example. In this figure, we present the contour plot of the likelihood function[2]

$$p_0^4 \cdot (p_0 p_1)^1 \cdot (1 - p_0)^2 \cdot (1 - p_0 p_1)^5 \ ,$$

which is a function of p_0 and p_1. The horizontal axis corresponds to p_0, and the vertical axis to p_1. We plot the likelihood instead of the log-likelihood[3] since the contours are better spaced. The lighter the color, the higher the value of the likelihood.

Since the local log-likelihood score of Noisy-OR $LL_v^\diamond(P|\mathbf{D})$ is a strictly concave function, it has a unique global maximum. This result is consistent with a previous weaker result of [13] stating that every stationary point of the log-likelihood of Noisy-OR is a global maximum. Now, we can introduce an important lemma about the log-likelihood function $LL^\diamond(P|\mathbf{D})$ of a Bayesian network with Noisy-OR models.

[2] Note the exponents correspond to frequencies of corresponding configurations in data $\mathbf{D}$.
[3] The log-likelihood is just the logarithm of the presented likelihood, which is strictly concave as well.

Lemma 2. *The log-likelihood $LL^{\diamond}(P|\mathbf{D})$ of a Bayesian network with Noisy-OR models has a unique maximum.*

Proof. $LL^{\diamond}(P|\mathbf{D}) = \sum_{v \in V} LL^{\diamond}_v(P|\mathbf{D})$ and functions $LL^{\diamond}_v(P|\mathbf{D})$ are strictly concave due to Lemma 1. Therefore, also $LL^{\diamond}(P|\mathbf{D})$ is concave and has a unique maximum.

In Algorithm 1, we present a method that can be used to learn maximum likelihood estimates of parameters of a Noisy-OR model. The presented method is derived from the EM learning method presented in Vomlel [28] and adapted for the leaky Noisy-OR model. The algorithm alternates between *E-step* and *M-step* until either a convergence criterion is met or a predefined maximum number of iterations is performed. We use symbol $\mathbf{p}_v$ as an abbreviation for vector $\big(p_{v,0}, (p_{v,j})_{j \in pa(v)}\big)$. Symbols $\oplus$, $\ominus$, $\otimes$, and $\oslash$ denote pointwise addition, subtraction, multiplication, and division of two vectors, respectively. The symbol $\mathbf{p}^{\mathbf{a}}$ denotes a vector $\big((p_0)^{a_0}, \ldots, (p_J)^{a_J}\big)$ of cardinality $|pa(v)|$, i.e. it denotes a pointwise exponentiation. If needed, we define $0/0 = 0$. In the *E-step*, the expected counts based on all data vectors from $\mathbf{D}$ are computed using the Noisy-OR model with parameters $\mathbf{p}_v$ from the previous algorithm iteration. In the *M-step*, these expected counts are used to update the parameters $\mathbf{p}_v$ of the Noisy-OR model.

input : v – the child node
U – parents of node v
$\mathbf{D} = \{\mathbf{x}_n\}_{n=1}^{N}$ – dataset of N complete data vectors $\mathbf{x}_n$
output: $\mathbf{p}_v$ – parameters of Noisy-OR maximizing log-likelihood

$\ell = |U| + 1$; /* the length of considered vectors */
$\delta = \ell$; /* the initial sum of squared differences */
$\Delta = 10^{-6}$; /* the maximal sum of squared differences */
$m = 0$; /* the initial number of iterations */
$M = 100$; /* the maximal number of iterations */
$\mathbf{p}_v = (0.5, \ldots, 0.5)$; /* vector of initial values of length ℓ */
while $(m < M) \wedge (\delta > \Delta)$ **do**
  $m = m + 1$; /* increase iteration counter */
  $\mathbf{p}'_v = \mathbf{p}_v$;
  $\mathbf{n}_0 = \mathbf{0}_\ell$; /* initialize as a zero vector of length ℓ */
  $\mathbf{n}_1 = \mathbf{0}_\ell$; /* initialize as a zero vector of length ℓ */
  for $n = 1, \ldots, N$; /* E-step: for all vectors from $\mathbf{D}$ do */
  do
    $c = x_{n,v}$; /* the child value in vector $\mathbf{x}_n$ */
    $\mathbf{a} = (1, \mathbf{x}_{n,pa(v)})$; /* vector of 1 and parents' values */
    if *(c==0)* **then**
      $\mathbf{r}_0 = \mathbf{1}_\ell$; /* a unit vector of length ℓ */
      $\mathbf{r}_1 = \mathbf{0}_\ell$; /* a zero vector of length ℓ */
    end
    else
      $q = \prod_{j=1}^{\ell} \mathbf{p}_j^{\mathbf{a}_j}$; /* the product of values in $\mathbf{p}^{\mathbf{a}}$ */
      $\mathbf{r}_0 = \mathbf{p}^{\mathbf{a}} \ominus \mathbf{q}_\ell$; /* $\mathbf{q}_\ell$ vector of length ℓ padded by q */
      $\mathbf{r}_1 = \mathbf{1}_\ell \ominus \mathbf{p}^{\mathbf{a}}$;
      $\mathbf{r} = \mathbf{r}_0 \oplus \mathbf{r}_1$; /* the normalization vector */
      $\mathbf{r}_0 = \mathbf{r}_0 \oslash \mathbf{r}$; /* pointwise normalization of $\mathbf{r}_0$ */
      $\mathbf{r}_1 = \mathbf{r}_1 \oslash \mathbf{r}$; /* pointwise normalization of $\mathbf{r}_1$ */
    end
    $\mathbf{n}_0 = \mathbf{n}_0 \oplus (\mathbf{a} \odot \mathbf{r}_0)$; /* pointwise addition and product */
    $\mathbf{n}_1 = \mathbf{n}_1 \oplus (\mathbf{a} \odot \mathbf{r}_1)$; /* pointwise addition and product */
  end
  $\mathbf{p}_v = \mathbf{n}_0 \oslash (\mathbf{n}_0 \oplus \mathbf{n}_1)$; /* M-step of the algorithm */
  $\delta = \|\mathbf{p}_v - \mathbf{p}'_v\|^2$;
end

Algorithm 1: The EM-algorithm for the leaky Noisy-OR model.

The computational complexity of one algorithm iteration is proportional to $N \cdot |fa(v)|$, which is well-tractable even for large parent sets.[4] Typically, the convergence of the algorithm is fast at the beginning and slow near the optimal value. In Fig. 2, we present an example of the development of the log-likelihood of a Noisy-OR model with five parents learned (1) from data used in experiments reported in Section 4 and (2) from randomly generated data.

Since the log-likelihood of Noisy-OR is well shaped (see Lemma 1), box-constrained gradient methods can also be used to find optimal parameters of Noisy-OR models. We have experimented with several methods: the Nedler-Mead method [17, Chapter 9.5], a box-constrained BFGS [17, Chapter 6.1], and the gradient projection method [17, Chapter 16.7]. Of these,

[4] In the practical implementation, in each *E-step*, it is convenient to perform the computation only once for each data record that is repeated in $\mathbf{D}$ and add the result as many times as the record is repeated.

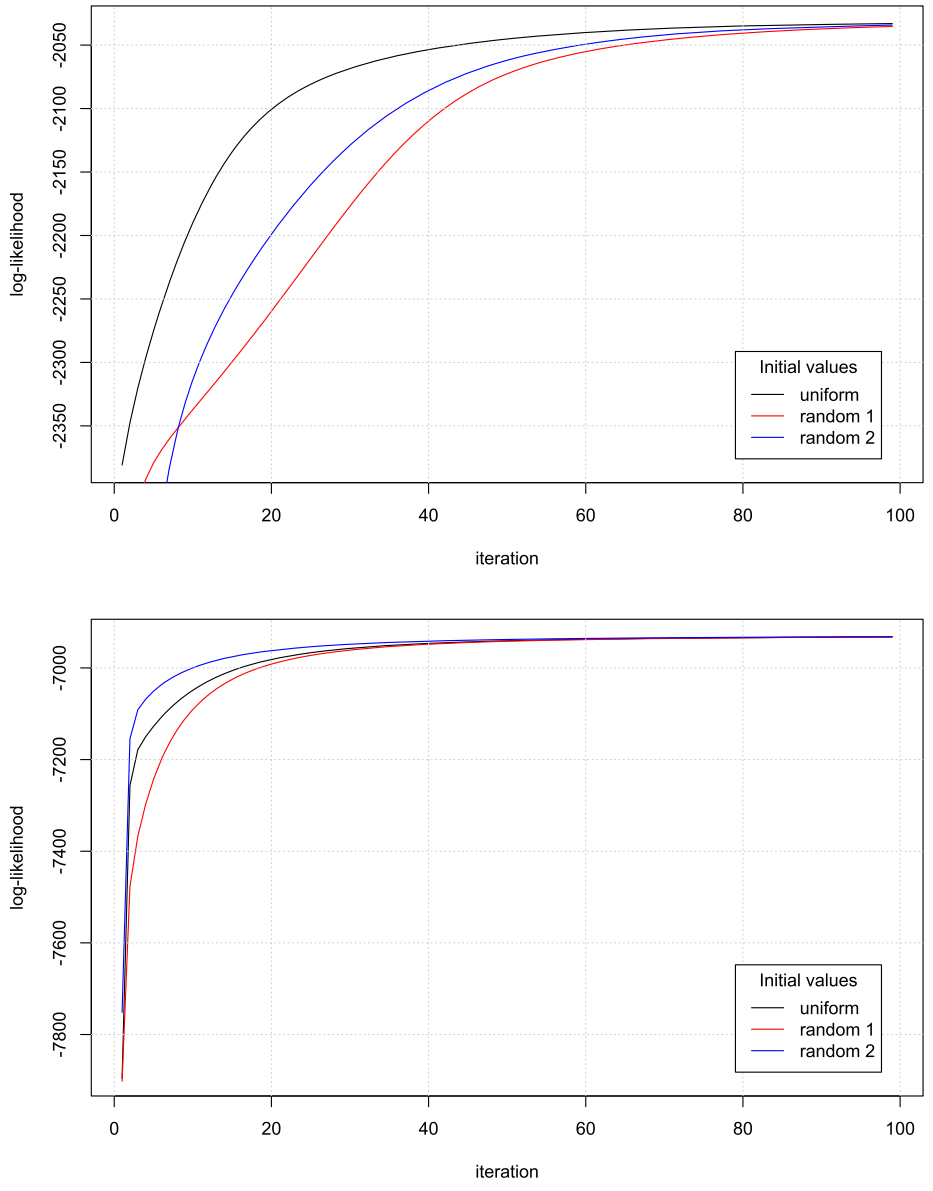

Fig. 2. The development of values of the log-likelihood during the EM learning of Noisy-OR from a dataset (with $N = 10000$) generated from (1) a Noisy-OR model (top) and (2) randomly with uniform probability for states 0 and 1 (bottom). (For interpretation of the colors in the figure, the reader is referred to the web version of this article.)

the gradient projection method performed best. In the experiments with synthetic and real data reported in Section 4 and Section 5, the computational time and the obtained optimal values of the gradient projection method were comparable to our implementation of the EM algorithm.

So far, we have been addressing the problem of learning parameters that maximize log-likelihood for a given BN structure specified by an acyclic directed graph. The next task is to learn a BN structure that best describes a given data **D**. It is well-known that the mere maximization of log-likelihood leads to dense graphs that are typical examples of overfitting training data. The Bayesian network model with the structure represented by the complete graph always has the highest log-likelihood value. Therefore, Bayesian network scoring functions that penalize networks with complex graphs are used. In this work, we will use the Bayesian Information Criterion (BIC) [23], although the presented approach can also be adapted for other scoring functions.

The BIC score is defined as the log-likelihood $LL(P|\mathbf{D})$ penalized by a penalty proportional to the number of parameters $C(P)$ of the Bayesian network representing probability distribution P:

$$BIC(P|\mathbf{D}) = LL(P|\mathbf{D}) \; - \; \frac{\log|\mathbf{D}|}{2} \cdot C(P) \; . \tag{11}$$

The penalty $C(P)$ is the total sum of the number of parameters of the individual conditional probability tables of the Bayesian network:

$$C(P) = \sum_{v \in V} C_v(P(X_v|X_{pa(v)}) \ . \tag{12}$$

In the case of binary variables, the penalty of a general CPT is

$$C_v(P(X_v|X_{pa(v)}) = (|\mathcal{X}_v| - 1) \prod_{j \in pa(v)} |\mathcal{X}_j| \ = \ 2^{|pa(v)|} \ . \tag{13}$$

In the case of a Noisy-OR model, the penalty is

$$C^{\diamond}_v(P(X_v|X_{pa(v)}) = |pa(v)| + 1 \ . \tag{14}$$

Note the significant difference between the penalty of a general conditional probability table and the penalty of a Noisy-OR. The former is exponential with respect to the number of parents, while the latter is only linear with respect to their number. This implies that if a Noisy-OR can replace a general table, more parents can be included in the model.

3. Structural learning of mixed noisy-OR Bayesian networks

It follows from the discussion in the previous section that learning a Noisy-OR Bayesian Network using methods based on standard penalty would typically lead to models with a substantially lower number of parents than are appropriate for the Noisy-OR models. Therefore structural learning of a Noisy-OR Bayesian Network should be based on a modified score function. In some practical applications of Bayesian networks (see Section 5 for such an application), some conditional probability tables have local structure, e.g., Noisy-OR, while others are better represented by general conditional probability tables.

Motivated by this observation, we propose a structure learning algorithm to decide which type of CPTs will be used for each node. Since the BIC scoring function is decomposable, this decision can be made locally for each node. The proposed algorithm decides between a general conditional probability table and a Noisy-OR. BNs whose CPTs are either general conditional probability tables or Noisy-OR models will be referred to as *Mixed Noisy-OR Bayesian Networks*. This approach could be easily extended to other local structure models of conditional probability tables for which the parameters maximizing the log-likelihood can be found. We present the algorithm for learning Mixed Noisy-OR Bayesian Networks in Algorithm 2. In the first stage, the algorithm computes maximum likelihood estimates for all nodes v, and their parent sets U. This is computed for both general CPTs and Noisy-OR models. Next, to learn maximum log-likelihood (MLL) estimates of Noisy-OR parameters, we use the EM algorithm discussed and presented as Algorithm 1 in Section 2. By adding the penalty terms, we get the BIC values of general CPTs and Noisy-OR models. These two BIC values are compared, and the values of v, U, and the higher BIC value are stored in the list of parent evaluations $\mathcal{L}$.

input : $\mathbf{D}$ – training dataset consisting of n complete data vectors
output: G – the structure of Bayesian network with CPTs being either standard CPTs or Noisy-OR models maximizing BIC score

$\mathcal{L} = \{\}$;
$w = \frac{\log|\mathbf{D}|}{2}$; /* the penalty weight */
for $v \in V$ **do**
  for $U \subseteq V \setminus \{v\}$ **do**
    $P(v|U) = \frac{N(\mathbf{x}_{v,U})}{N(\mathbf{x}_U)}$; /* MLL estimate of general CPT */
    $s_1 = LL_v(P(v|U))$; /* MLL score of general CPT */
    $c_1 = w \cdot 2^{|U|}$; /* penalty of general CPT */
    $BIC_1 = s_1 - c_1$; /* BIC of general CPT */
    $\mathbf{p} =$ EM.Algorithm$(v, U, \mathbf{D})$; /* MLL estimate of Noisy-OR */
    $s_2 = LL^{\diamond}_v(\mathbf{p}, v, U)$; /* MLL score for Noisy-OR */
    $c_2 = w \cdot (|U| + 1)$; /* penalty of Noisy-OR */
    $BIC_2 = s_2 - c_2$; /* BIC of Noisy-OR */
    if $BIC_1 > BIC_2$ **then**
      $\mathcal{L} = \mathcal{L} \cup (v, U, BIC_1)$; /* general CPT is added */
    else
      $\mathcal{L} = \mathcal{L} \cup (v, U, BIC_2)$; /* Noisy-OR is added */
    end
  end
end
$G =$ GOBNILP$(\mathcal{L})$; /* apply GOBNILP with the list $\mathcal{L}$ */

Algorithm 2: Learning the structure of a Mixed Noisy-OR BN.

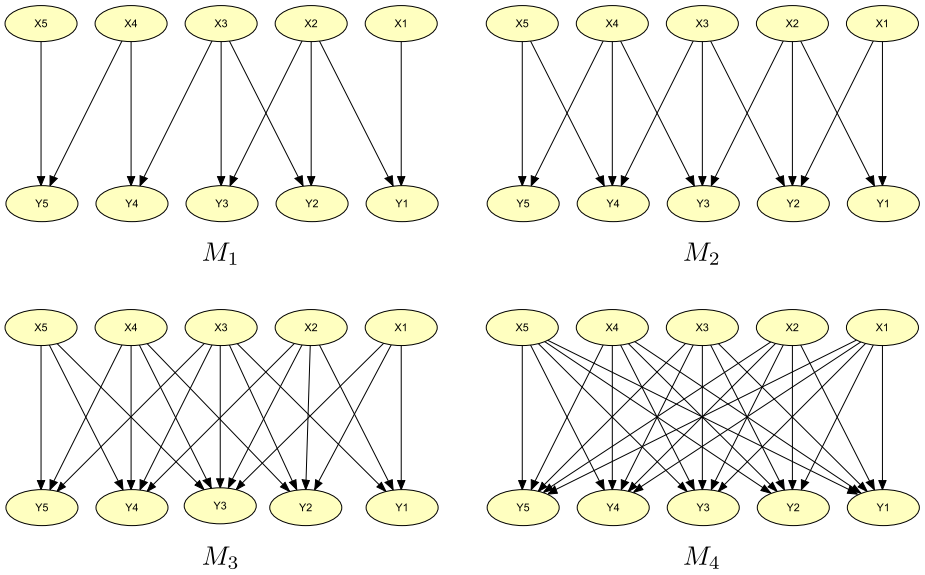

Fig. 3. Four BN2O network structures used in the experiments.

Before adding a triplet (v, U, BIC) to the list $\mathcal{L}$, a pruning strategy should be applied so that configurations of (v, U) that cannot be part of an optimal Bayesian network are omitted from the list $\mathcal{L}$. For example, Lemma 3 suggests a pruning method applicable to all CPTs, including Noisy-OR models. This is a well-known result, and it is presented, e.g., in [7] as Lemma 1.

Lemma 3. *A triplet (v, U, BIC) such that there is a triplet $(v, U', BIC') \in \mathcal{L}$ satisfying $U' \subset U$ and $BIC' > BIC$ can be safely excluded from the list $\mathcal{L}$.*

Several other pruning rules for general CPTs were suggested by [7], but they do not apply to Noisy-OR models. [25] proposed two pruning rules for Noisy-OR models. The first pruning rule from [25, Lemma 4] suggests eliminating all sets containing node u such that $X_v = 1$ implies $X_u = 0$ in the training data **D** from the search of candidate parent sets U of a node v. The second pruning rule from [25, Theorem 5] can be easily generalized as: Given a triplet (v, U', BIC'_2) all triplets (v, U, BIC_i), $i = 1, 2$ with $BIC_i = s_i + c_i$ such that $BIC'_2 > -c_i$ can be eliminated from the search. Note that if it holds for a triplet (v, U, BIC_i) then it also holds for all triplets (v, U'', BIC''_i) with $U'' \supset U$ since the penalty can only increase with larger parent sets. A discussion on the application of these pruning rules can be found in Section 5.

The algorithm's final step is applying the GOBNILP method developed by [5]. GOBNILP [6] is a program that learns optimal Bayesian networks from a list of local scores. It uses the SCIP framework for Constraint Integer Programming as its core routine.

4. Experiments with BN2O networks

This section describes experiments designed to determine the conditions under which the studied algorithms can learn the original models. In addition, we also compare the prediction performance of the learned models. To this end, we decided to use BN2O networks since these networks are standard in practical applications of BNs, e.g., in medicine [26] and educational assessment [1] and since these models can be systematically (as opposed to ad hoc) analyzed.

A BN2O network is a Bayesian network consisting of two layers of nodes. All edges are directed from the top layer to the bottom layer. There are no edges connecting nodes in the same layer. Nodes from the bottom layer usually share some parents but not necessarily all. The graphs of BN2O networks used in our experiments are presented in Fig. 3. All conditional probability tables are Noisy-OR models.

The CPTs of all nodes from the bottom layer (children) of model M_1 are the smallest possible Noisy-OR models connected with only two nodes from the top layer (parents). The higher the model index, the greater the number of parents. Actually, model M_4 is a complete BN2O model in the sense that all children have all nodes from the first layer as their parents. We expect it will be more challenging to learn models with children having more parents, especially in the case of the standard BNs. We discuss this in detail in this section.

First, we generated a dataset of 15810 data records from each model. Then we split this dataset into several smaller datasets so that each vector from the original dataset was used only once in the datasets of the same size. The dataset sizes are chosen to cover the range of our interests.

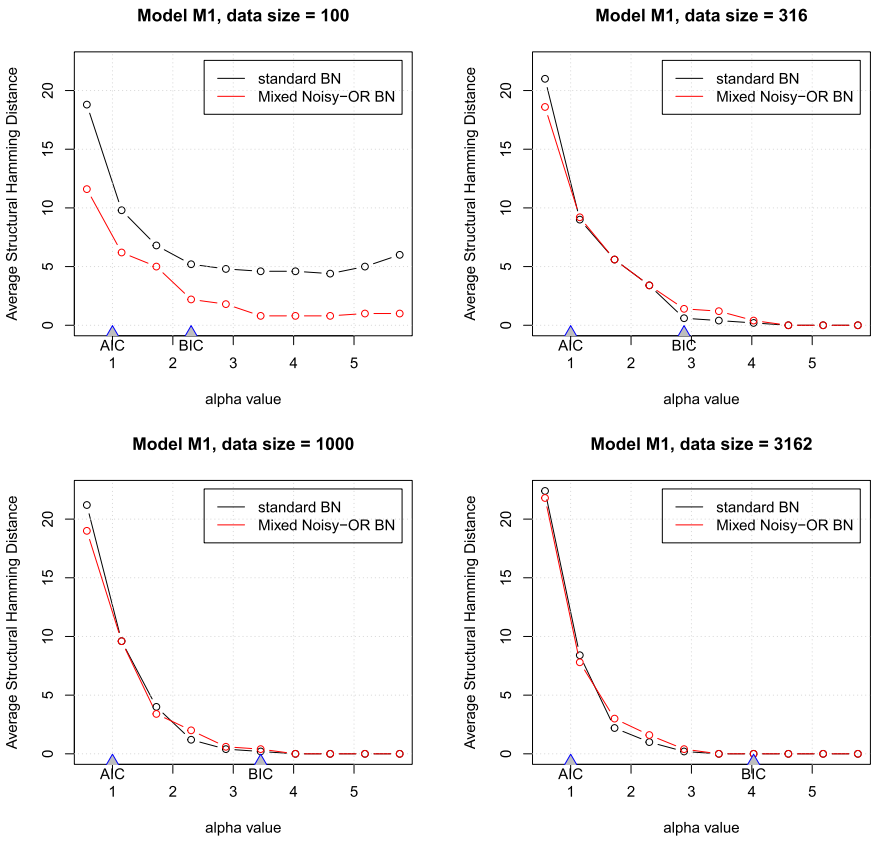

Fig. 4. Average SHD of the learned and the original BNs for model M_1. (For interpretation of the colors in the figure, the reader is referred to the web version of this article.)

In the experiments, we analyzed the dependence of the quality of learned models on training data size and the effect of different penalization with respect to the number of model parameters. For this purpose, we used parameter α, which was used as a multiplier for the number of parameters of each model. This means the maximized learning criteria were:

$$XIC_\alpha(P|\mathbf{D}) = LL(P|\mathbf{D}) \ - \ \alpha \cdot C(P) \ . \tag{15}$$

The penalty $C(P)$ is the total sum of the number of parameters of the Bayesian network's individual conditional probability tables as defined in Section 2 for a general CPT and for a Noisy-OR model. See formulas (13) and (14), respectively. The parameter α represents the strength of the penalization. Please, note that if $\alpha = \frac{1}{2}\log|\mathbf{D}|$ then XIC_α corresponds to the BIC criterion defined in Section 2 and if $\alpha = 1$ then XIC_α corresponds to the AIC criterion.

In Figs. 4–7, we present the dependence of the Structural Hamming Distance (SHD) of BNs that are optimal with respect to the XIC_α criterion. We have considered

$$\alpha \in \log(10) \cdot \left\{1 + \frac{i}{4}\right\}_{i \in \{-3,-2,\ldots,+5,+6\}} \ , \tag{16}$$

and four different sizes of training data. The data sizes correspond to four elements of a rounded geometric sequence $10^{(2+i/2)}$, $i = 0, 1, \ldots, 3$, which gives the following data sizes $|\mathbf{D}| = 100, 316, 1000, 3162$. These data sizes combine well with the set of values of α so that for each of these data sizes, we have an α value corresponding to $XIC_\alpha = BIC$ for that data. A standard BN and a Mixed Noisy-OR BN optimal with respect to XIC_α are learned from each dataset.

The data were generated from models M_1, M_2, M_3, and M_4, respectively (see Figs. 4–7 for the average SHD of the learned and the original BNs of these models). The reported values of SHD represent the average taken over five BNs learned from five datasets. The axes of the graphs show the values of the parameter α corresponding to the AIC and the BIC criteria. Next, we summarize the results.

For simpler models as are M_1 and M_2, we are able to learn the original structure if the training dataset is larger, i.e., if $|\mathbf{D}| \geq 1000$. For these data sizes α greater or equal to the value corresponding to BIC can guarantee recovering the original models. If the value of α meets this condition then the results are rather insensitive to its actual value. For smaller training datasets Mixed Noisy-OR BNs represent a substantially better fit to the original model than standard BNs and (with the exception of the smallest training dataset of M_2) the higher the value of α the better is their fit to the original model.

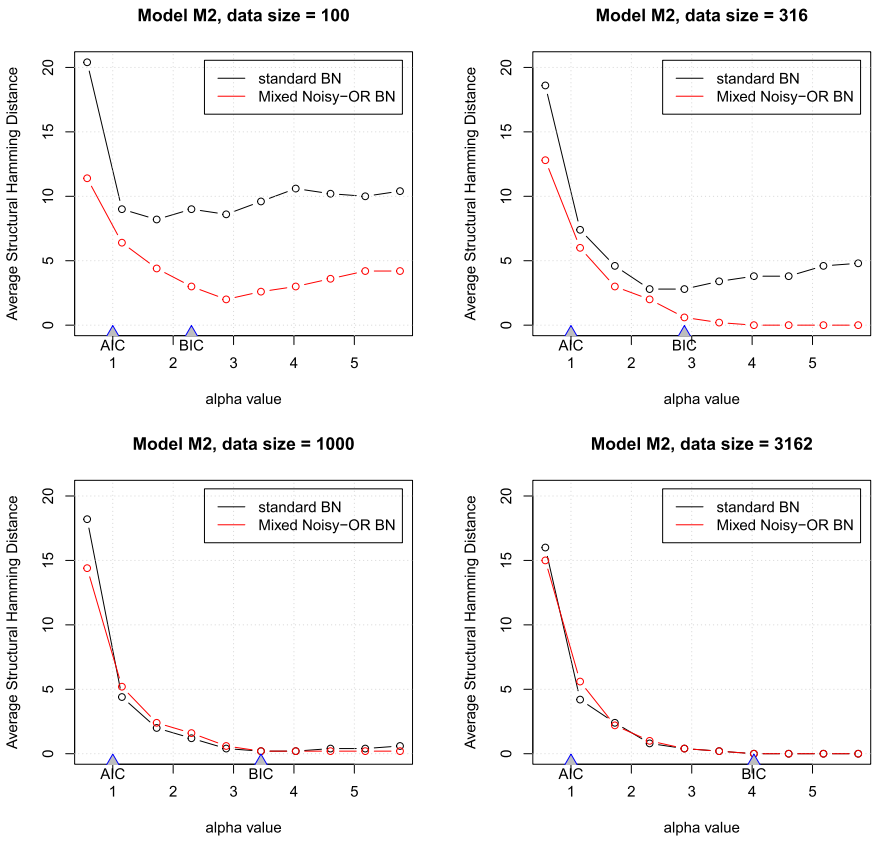

Fig. 5. Average SHD of the learned and the original BNs for model M_2. (For interpretation of the colors in the figure, the reader is referred to the web version of this article.)

The situation changes as the complexity of the considered model increases more than proportionally with the size of the training datasets. The correct original model can no longer be identified using standard BNs and it is tricky to find a value of α that leads to the best fit. Generally, the value of α should be rather low, corresponding to the value of the AIC criterion. The more complex the model the more data is needed also for Mixed Noisy-OR BNs. The higher the ratio of data size to the model size the lower is the influence of the actual value of α. If there is relatively very little data then the lower values of α lead to better fits of the original model even for Mixed Noisy-OR BNs. For example, in case of model M_4 and $|\mathbf{D}| = 100$, $\alpha = 1$, which corresponds to the value of the AIC criterion, seems to lead to the best result.

In summary, the experiments confirm that mixed Noisy-OR BNs, which consider both standard CPT and Noisy-OR models and use the BIC penalization, can identify the correct Bayesian network structure for much smaller training datasets than standard BNs. The quality of their fit is less dependent on the actual value of the penalization constant α.

In Fig. 8 we present examples of Mixed Noisy-OR BNs and standard BNs learned from data generated from model M_3. Red edges indicate redundant edges, orange edges have been learned reversed and blue dotted edges are missing. Model (a) is both the original model and the model learned by the mixed BIC optimization from 3162 and 10000 data records. Model (b) was learned using the standard BIC optimization from 10000 data records. Model (c) was learned using the mixed BIC optimization from a set of 316 data records. Model (d) was learned using the standard BIC optimization from the same 316 data records. We can clearly see that the learned Mixed Noisy-OR BNs is almost identical to the original model already for 316 data records, while under the same conditions, the standard BIC optimization fails.

One of the tasks for which Bayesian networks are used is the prediction of the most probable states of certain variables given observations of some other variables in the model. It can be expected that models with a structure similar to the original model can perform better; however, sometimes simple models perform comparably well. To see if Mixed Noisy-OR Bayesian Networks have better performance than standard Bayesian Networks learned from the same data, we performed experiments in which we studied the prediction ability of the models as a function of training data size. Since probability distributions of our BN2O models are typically imbalanced (one state is significantly more probable than the other state), we decided to use balanced accuracy[5] as our evaluation criterion. In Fig. 9, we present average results for model M_3 and for the scenario when evidence was inserted for five randomly selected variables and the states of the other five variables

[5] Balanced accuracy is defined as the arithmetic mean of sensitivity and specificity.

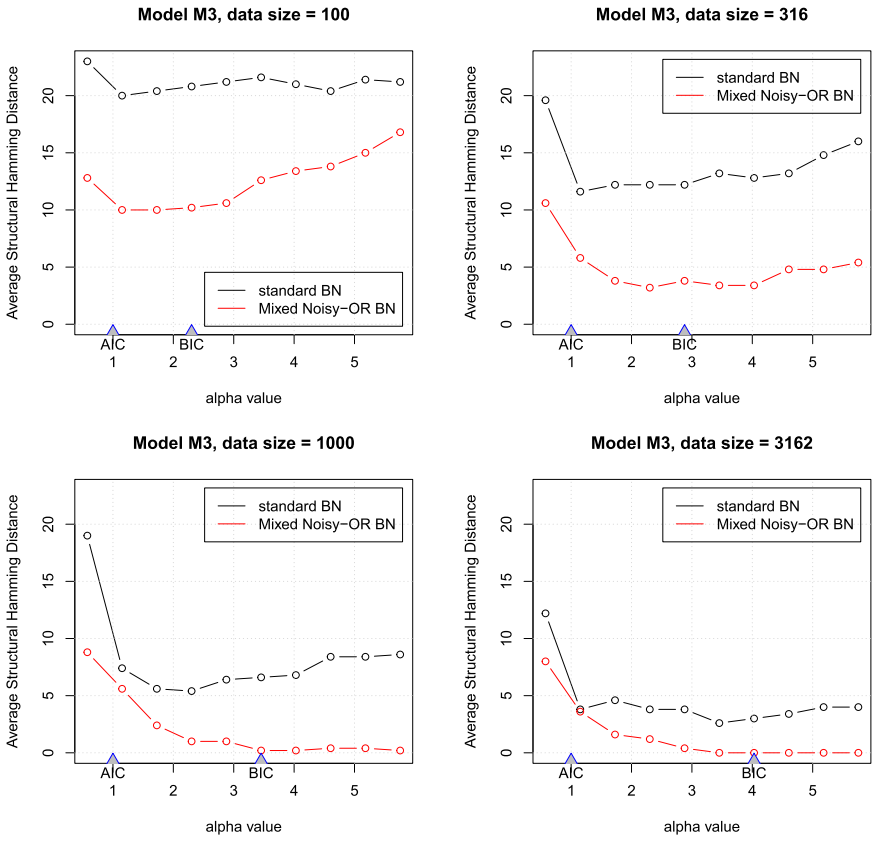

Fig. 6. Average SHD of the learned and the original BNs for model M_3. (For interpretation of the colors in the figure, the reader is referred to the web version of this article.)

were predicted. We present the balanced accuracy as a function of the training data size (please, note the log scale for the data size). The experiments confirm that Mixed Noisy-OR Bayesian Networks posses a better prediction ability than standard BNs,[6] especially for smaller training datasets.

5. Modeling the spread of loanwords

The topic of this paper – learning Mixed Noisy-OR Bayesian Networks – was originally motivated by our collaboration with linguists on a research project aimed at modeling language phenomena of borrowing relations in incomplete comparative databases characterized by uncertainty. In this paper we use the loanword inventories to model the spread of loanwords in the area of the Insular South-East Asia, which is characterized by a large number of languages distributed about the archipelago's many islands.

A loanword is a word permanently adopted from one language and incorporated into another without translation. While written records and archaeological evidence are mostly missing in this region, the distribution of loanwords offers preliminary insight into past human migrations, contacts, and trade. The mapping of the loanword distribution offers an opportunity to capture large patterns of human contact that are not as readily detectable by other means.

Our primary resource is a database of loanwords collected from several sources [e.g. 2]. The database is available at http://gogo.utia.cas.cz/loanwords/. All the loanwords we consider originate from one of twelve donor languages in Table 2. In our experiments reported in this paper, we use a dataset providing information about the presence/absence of 461 loanwords in 23 recipient languages presented in Table 3. A detailed description of the problem and the results of an experimental analysis using a heuristic BN learning method were presented in [14].

The task is to learn a Bayesian network having recipient languages from the studied region as its variables. Variables are binary, with states 0 and 1 representing the absence and presence of a loanword in the corresponding recipient language. Noisy-OR models seem to be a natural model for this problem. Particularly, it means that the presence of a loanword in related languages (represented by parent variables in the Bayesian network) increases the probability of that loanword being

[6] Both types of BNs were learned by maximizing the corresponding BIC criterion.

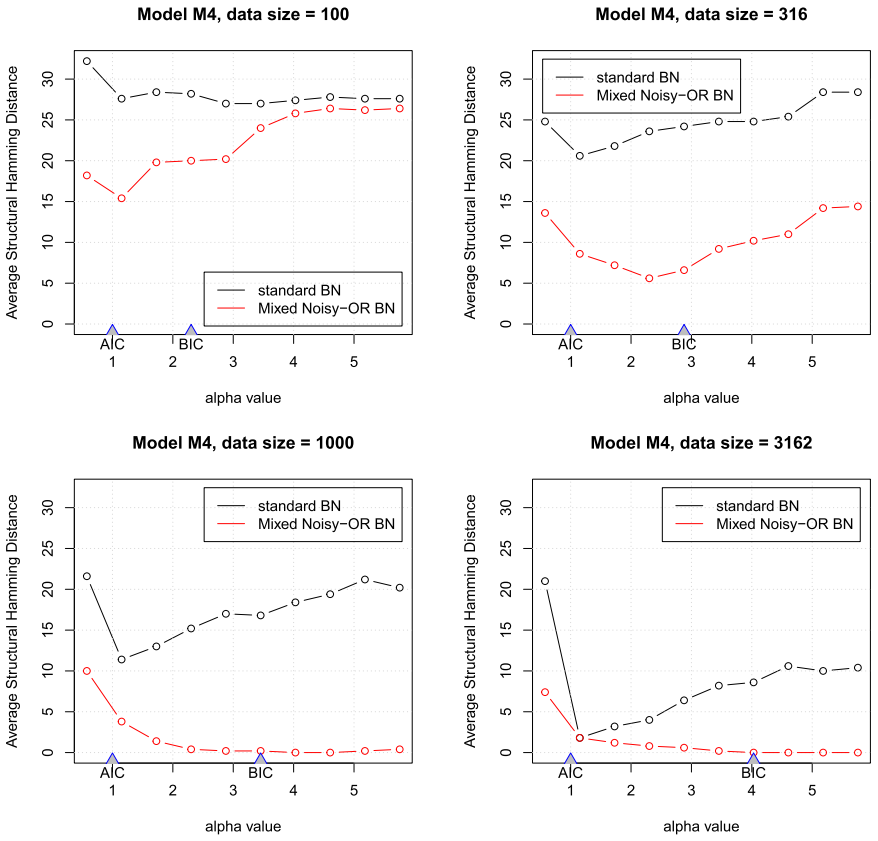

Fig. 7. Average SHD of the learned and the original BNs for model M_4. (For interpretation of the colors in the figure, the reader is referred to the web version of this article.)

Table 2
Donor languages.

Arabic	Chinese	Malagasy
Malay	Old Javanese	Persian
Proto-Malay	Prakrit	Roman languages
Sanskrit	Tagalog	Tamil

Table 3
Recipient languages.

Acehnese	Aklanon	Balinese
Bikol	Casiguran Dumagat	Cebuano
Iban	Javanese	Kapampangan
Karo Batak	Makasarese	Malay
Manggarai	Maranao	Melanau
Mongondow	Rejang	Rembong
Sangir	Simalur	Tagalog
Tiruray	Wolio	

present in the language corresponding to the child variable. Moreover, noisy-OR seems more appropriate than deterministic OR since some languages, such as Malay or Javanese, have rich resources, and the loanword inventories are reliable. In contrast, other languages are low-resource and the loanword inventories are likely incomplete. Also, other types of Noisy-OR models may be considered appropriate for this task, e.g., the deterministic OR model with the slip and guessing probabilities (used in psychometrics) may also be used. However, in the sequel text, we will discuss the application of Mixed Noisy-OR BNs where the CPTs are either standard CPTs or leaky Noisy-OR models. Mixed Noisy-OR BNs seem to represent the modeled domain better than BNs with all CPTs being Noisy-OR models since our preliminary results revealed that the assumption of all conditional probability tables being represented by Noisy-OR models worsened the performance. It seems better to let the learning algorithm decide whether Noisy-OR or the general CPT is more appropriate for each CPT.

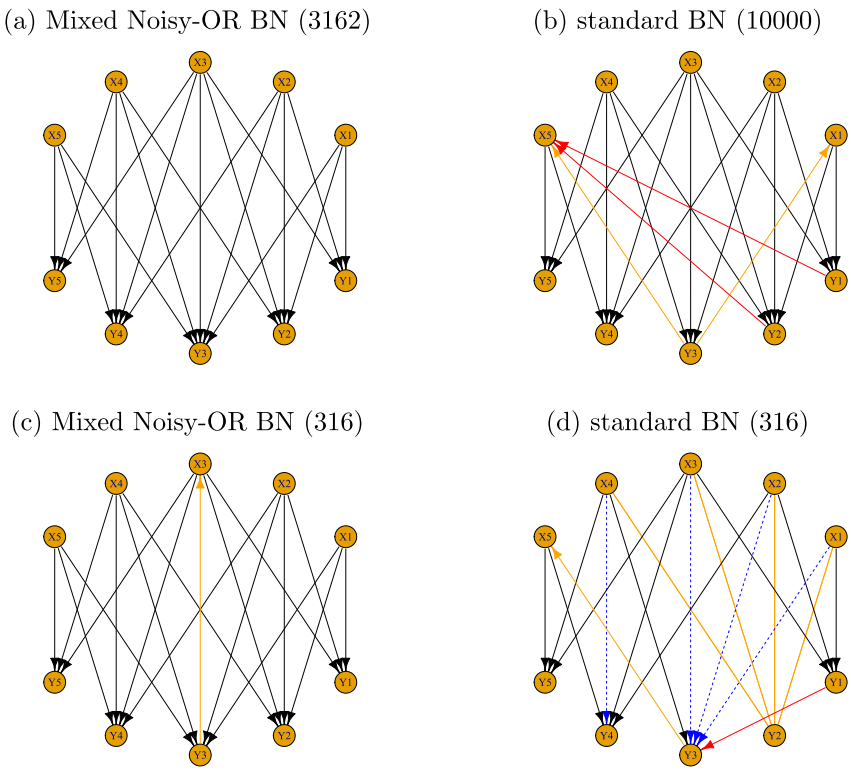

Fig. 8. Learned models with the training data size given in parentheses. (For interpretation of the colors in the figure, the reader is referred to the web version of this article.)

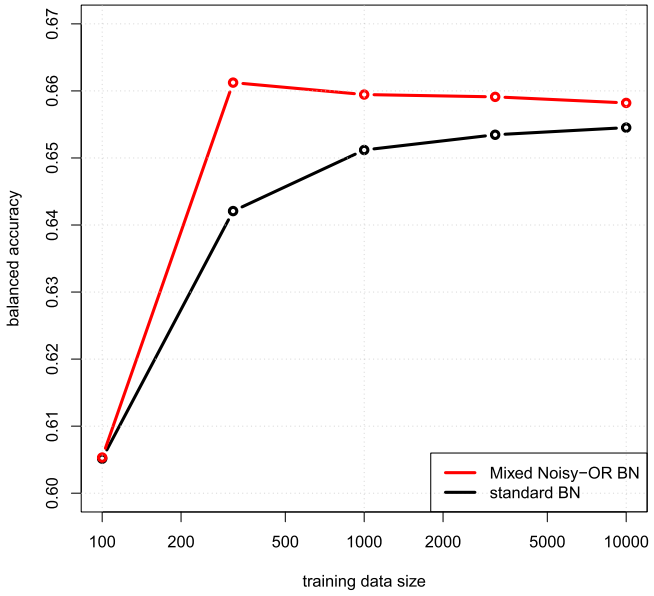

Fig. 9. The balanced accuracy of model M_3. (For interpretation of the colors in the figure, the reader is referred to the web version of this article.)

Parent sets pruning

In the case of a Noisy-OR model, the penalty in the BIC score is significantly lower than for general CPTs. To see this, compare (13) with (14). This implies that the BIC values of large parent sets have to be computed for Noisy-OR models. For this reason, it is important to apply efficient pruning to avoid huge lists of evaluated parent sets.[7] Unfortunately, the

[7] Please, note that here we consider only pruning methods that guarantee optimal Mixed Noisy-OR BNs.

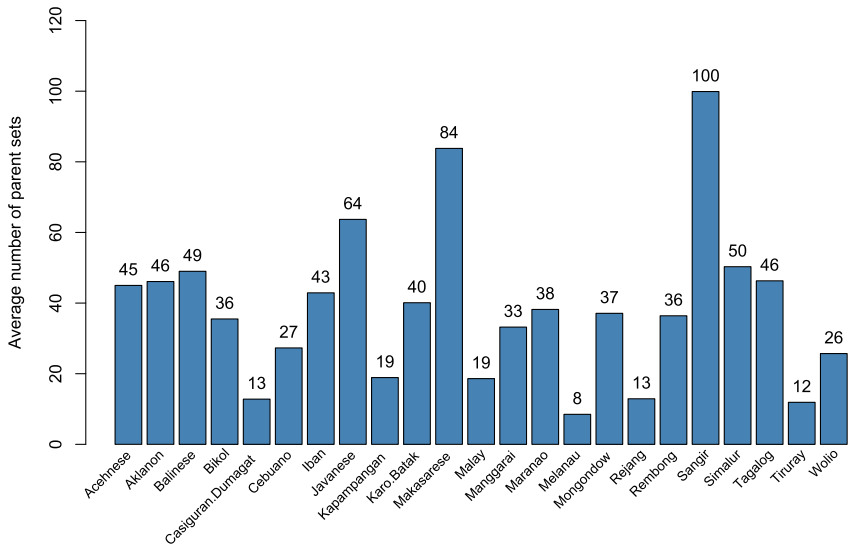

Fig. 10. The average[9] number of parent sets after pruning.

pruning rules from Sharma et al. [25] were inefficient in this case. The first pruning rule of Sharma et al. [25, Lemma 4] suggests eliminating from the search of candidate parent sets U of a node v all sets containing node u such that $X_v = 1$ implies $X_u = 0$ in the training data $\mathbf{D}$. This is very rare in our data. The average number[8] of such node-parent pairs was only three (out of possible 506). The second pruning rule of [25] did not exclude any parent sets.

Instead, we applied the essential pruning based on Lemma 3. This pruning method reduces the number of parent sets significantly; see Fig. 10 where the average number of the parent sets for each of 23 variables is presented[8]. Without pruning, the number of parent sets would be $2^{22} = 4,194,304$ for all variables from the BN model. GOBNILP computations with the pruned list of parent sets were several orders of magnitude faster than GOBNILP computations with complete lists of parent sets of cardinality at most six (a heuristic constraint that does not guarantee the optimality of the resulting model). Specifically, the GOBNILP computations with pruned lists took a few seconds instead of several hours. A disadvantage is that the pruning step is computationally demanding since the number of sets whose BIC is computed in the pruning step is exponential with respect to the number of variables in the BN model.

It would be advantageous if we could find new pruning rules that would effectively reduce the number of parents for Noisy-OR models without evaluating them all. We have already made some attempts in this direction and reduced the number of evaluated parents significantly. Still, we do not have any formal proof they guarantee optimal solutions (although we have not found any counterexamples).

We also implemented and tested a fast suboptimal method in which we pruned all supersets of a parent set that had lower BIC scores than their subsets, i.e., we pruned out all triplets (v, U'', BIC'') if $\exists (v, U, BIC) \in \mathcal{L}$, $U \subset U''$ which was pruned since there is a $(v, U', BIC') \in \mathcal{L}$ satisfying $U' \subset U$ and $BIC' > BIC$. Of course, this approach does not guarantee optimality, but it helps quickly reduce the list of triplets $\mathcal{L}$.

During the experiments, we observed that parent sets such that none of their subsets of cardinality one less had a better score than all its subsets also did not have a better score than all its subsets. This leads us to conjecture that a triplet (v, U', BIC') can be pruned if for all $U \subset U'$ such that $|U| = |U'| - 1$ holds $(v, U, BIC) \notin \mathcal{L}$. However, since we could not prove this conjecture, we tried to find a counterexample. Using extensive computations, we found a counterexample in our data about loanwords. It was quite challenging to find a counterexample because this behavior only occurred in 27 sets out of about 42 million candidate sets. In Table 4 we present simplified data that can be used to construct a counterexample in a much simpler way.

From Table 4 we computed log-likelihood and penalty values for parent sets listed in Table 5. This table shows that the BIC score for parent set $\{X_1, X_3\}$ is better than the score of any parent set of cardinality 3. Nevertheless, the BIC score of $\{X_1, X_2, X_3, X_4\}$ achieves even better value than the BIC score of $\{X_1, X_3\}$. Note that in the case of parent sets of cardinality 3, the log-likelihood function increases, but the penalty cancels this improvement. Only in the case of the parent set of cardinality, 4 is the improvement in the log-likelihood greater than the penalty for the increase in the number of parameters.

It remains an open problem whether there are pruning rules that efficiently prune the space of parent sets for mixed Noisy-OR BNs to a similar extent as for standard BNs.

[8] The average is computed over ten training datasets.

Table 4
Data for the counterexample to the pruning rule based on cardinality.

Y	X_1	X_2	X_3	X_4	Frequency
0	0	0	0	0	179
0	0	0	0	1	68
0	0	1	0	1	34
0	0	1	0	0	47
1	0	1	0	1	2
1	0	0	1	0	2
1	0	1	0	0	4
1	0	0	0	1	4
1	1	1	0	0	2

Table 5
The scores of Noisy-OR models for $P(Y|U)$.

U	LL	penalty	BIC	Noisy-OR parameters
$\{X_1,X_2\}$	-50.211	8.752	-58.963	(0.976, 0, 0.954)
$\{X_1,X_3\}$	-45.055	8.752	-53.807	(0.97, 0, 0)
$\{X_1,X_4\}$	-51.024	8.752	-59.776	(0.974, 0, 0.97)
$\{X_2,X_3\}$	-47.427	8.752	-56.180	(0.984, 0.925, 0)
$\{X_2,X_4\}$	-53.997	8.752	-62.749	(0.988, 0.927, 0.968)
$\{X_3,X_4\}$	-51.024	8.752	-59.776	(0.974, 0, 0.97)
$\{X_1,X_2,X_3\}$	-42.358	11.670	-54.027	(0.984, 0, 0.946, 0)
$\{X_1,X_2,X_4\}$	-48.745	11.670	-60.415	(0.988, 0, 0.951, 0.965)
$\{X_1,X_3,X_4\}$	-43.345	11.670	-55.014	(0.983, 0, 0, 0.961)
$\{X_2,X_3,X_4\}$	-43.227	11.670	-54.896	(1, 0.917, 0, 0.956)
$\{X_1,X_2,X_3,X_4\}$	-37.930	14.587	-52.517	(1, 0, 0.941, 0, 0.955)

Experimental results

Fig. 11 shows the optimal structure of the Noisy-OR mixed Bayesian network for one of the ten training datasets. The positions of the nodes correspond to the geographical coordinates of the languages under study. Interestingly, edges often connect neighboring languages, but there are also several edges between distant locations. This could be potentially explained by historical trade routes, but this hypothesis should be further investigated by linguists and historians of the region. It is interesting to note that Noisy-OR models represent in average 52.17% of CPTs in the optimal Mixed Noisy-OR Bayesian Networks.

We used the ten-fold cross-validation method to compare the learned models. The BIC values of models learned in each of the ten folds are compared for all three methods in Fig. 12. In Fig. 13, we present results of the balanced accuracy as a function of the number of variables with evidence.[10] As expected, the more variables are observed, the better the prediction quality.

Mixed Noisy-OR Bayesian Networks performed consistently better than optimal Bayesian Networks with standard CPTs. Despite the differences in the BIC values being large, the differences in prediction measured by balanced accuracy are relatively small. This can be explained by the general difficulty of correctly predicting the presence of a loanword in another language using models learned from a relatively small training set, as well as the inevitable partially stochastic nature of loanword presence.

6. Conclusions and open problems

We studied learning of Mixed Noisy-OR Bayesian Networks. We proved the log-likelihood function of a Noisy-OR model has a unique global maximum and adapted the EM-learning method of [28] for learning leaky Noisy-OR models.

We evaluated the proposed approach on synthetic data generated from BN2O models of different complexity. We find that in most cases the results of Mixed Noisy-OR Bayesian Networks are significantly better than those of standard Bayesian networks and BNs perform similarly only when the training datasets are large enough given the complexity of the model.

Our study of different penalizations as a function of the number of model parameters showed that when learning Mixed Noisy-OR BNs, the BIC score is a good choice for most models and data sizes, and the sensitivity to the penalization coefficient is relatively low for large datasets.

We applied the method to the problem of modeling the spread of loanwords in the area of the South-East Asia Archipelago. The learned Bayesian network models represent a valuable source of information for linguists and historians studying the considered region.

[9] The average is computed over ten different training sets.
[10] For each vector from the testing dataset, the evidence nodes were chosen randomly.

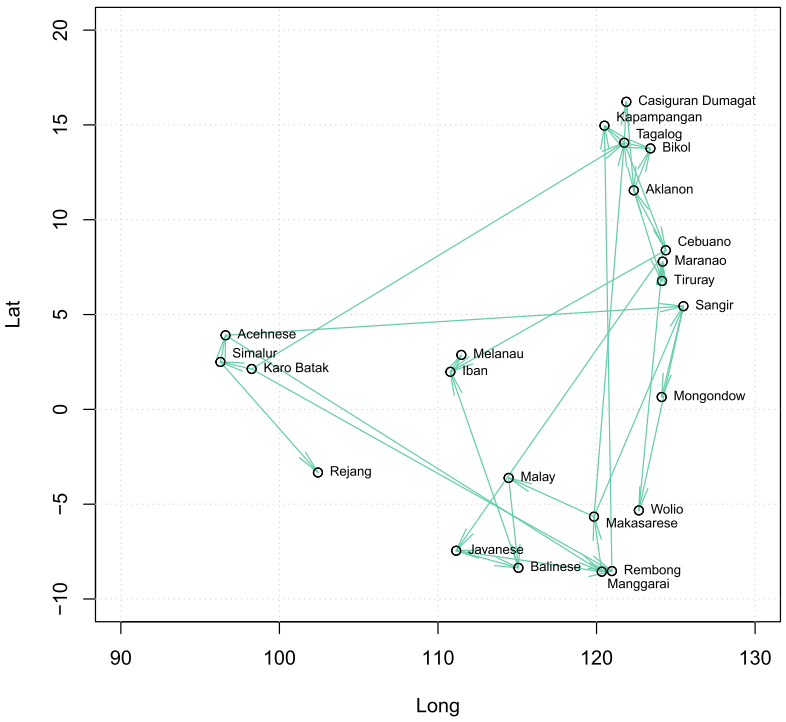

Fig. 11. Mixed Noisy-OR Bayesian Network modeling the spread of loanwords. The position of the nodes corresponds to the geographical location of the respective languages.

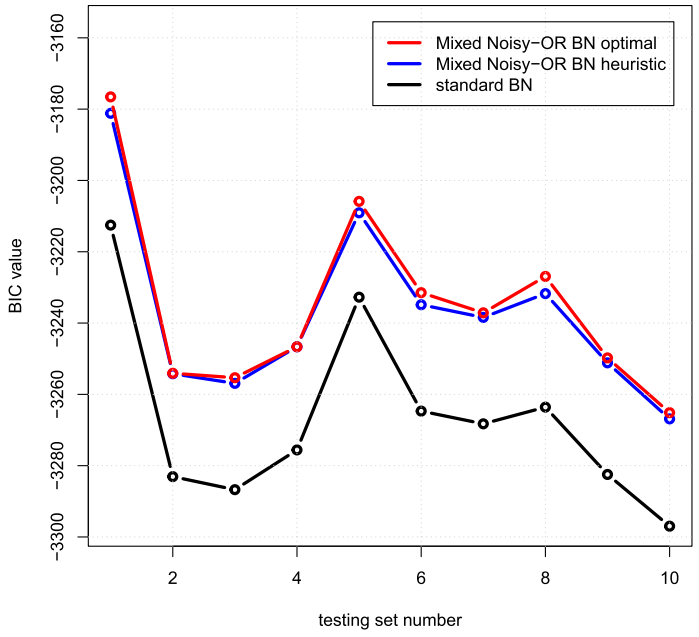

Fig. 12. BIC values of models learned in each of the ten folds. (For interpretation of the colors in the figure, the reader is referred to the web version of this article.)

From a theoretical point of view, we have left open the problem of computationally efficient pruning for Noisy-OR models. The extremely low numbers of parent sets needed for the structural search for the optimal model (shown in Fig. 10) suggest that it should be possible to find such pruning rules.

The discussed learning method can be extended to other models of the local structure of CPTs if their maximum likelihood estimates can be efficiently found. Thus, another topic for future research may be the structural learning of mixed BNs,

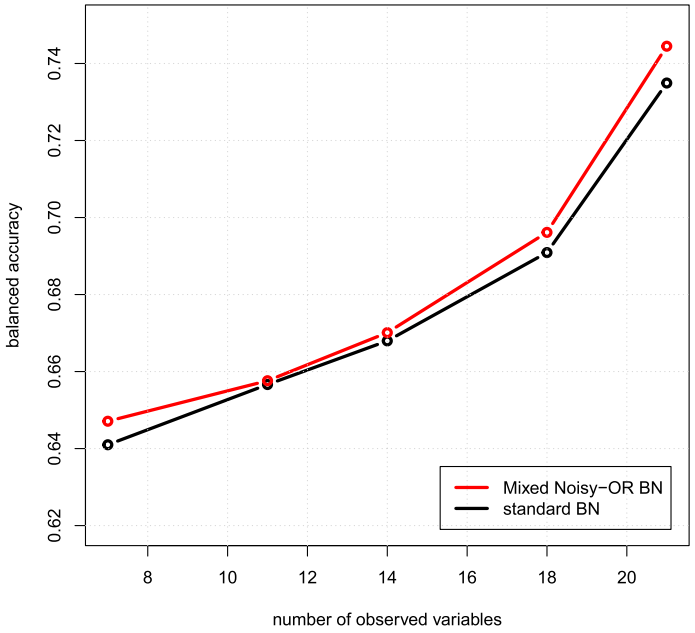

Fig. 13. Balanced accuracy of loanwords prediction. (For interpretation of the colors in the figure, the reader is referred to the web version of this article.)

where CPTs can be either standard CPTs or represented by logistic regression models as an extension of the work of [21] and [16]. The main difference with Noisy-OR models is that in the logistic regression models, parent variables can have both positive and negative influences on the child variable - as already mentioned in Appendix A of [22]. This model would be useful for many practical applications of BNs and could represent an alternative approach to BNs with CPTs modeled using polynomial regression models represented as piecewise spline functions proposed by [24].

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jiří Vomlel reports financial support was provided by Czech Science Foundation.

Data availability

Data will be made available on request.

References

[1] R.G. Almond, R.J. Mislevy, L.S. Steinberg, D. Yan, D.M. Williamson, Bayesian Networks in Educational Assessment, Springer, New York, 2015.
[2] R. Blust, S. Trussel, A.D. Smith, CLDF dataset derived from Blust's "Austronesian Comparative Dictionary", (v1.2) [Data set], Zenodo, URL https://doi.org/10.5281/zenodo.7741197, 2023.
[3] D.M. Chickering, Optimal structure identification with greedy search, J. Mach. Learn. Res. 3 (2002) 507–554.
[4] D.M. Chickering, D. Heckerman, C. Meek, Large-sample learning of Bayesian networks is NP-hard, J. Mach. Learn. Res. 5 (2004) 1287–1330.
[5] J. Cussens, M. Bartlett, GOBNILP, Version 1.6.3 https://www.cs.york.ac.uk/aig/sw/gobnilp/, 2018.
[6] J. Cussens, M. Järvisalo, J.H. Korhonen, M. Bartlett, Bayesian network structure learning with integer programming: polytopes, facets and complexity, J. Artif. Intell. Res. 58 (2017) 185–229.
[7] C.P. de Campos, M. Scanagatta, G. Corani, M. Zaffalon, Entropy-based pruning for learning Bayesian networks using BIC, Artif. Intell. 260 (2018) 42–50.
[8] F.J. Díez, M.J. Druzdzel, Canonical probabilistic models for knowledge engineering, Technical Report CISIAD-06-01, UNED, Madrid, Spain, 2006.
[9] F.J. Díez, S.F. Galán, An efficient factorization for the noisy MAX, Int. J. Intell. Syst. 18 (2003) 165–177.
[10] N. Friedman, M. Goldszmidt, Learning Bayesian networks with local structure, in: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, 1996, pp. 252–262.
[11] M. Henrion, Some practical issues in constructing belief networks, in: Proceedings of the Third Conference on Uncertainty in Artificial Intelligence, Elsevier Science Publishers B.V. (North Holland), 1987, pp. 161–173.
[12] F.V. Jensen, T.D. Nielsen, Bayesian Networks and Decision Graphs. Information Science and Statistics, 2 ed., Springer, New York, NY, 2007.
[13] R. Jurgelenaite, T. Heskes, Learning symmetric causal independence models, Mach. Learn. 71 (2008) 461–464, https://doi.org/10.1007/s10994-007-5041-7.
[14] F. Kratochvíl, V. Kratochvíl, G. Saad, J. Vomlel, Modeling the spread of loanwords in South-East Asia using sailing navigation software and Bayesian networks, in: Proceedings of the 12th Workshop on Uncertainty Processing (WUPES'22), MatfyzPress, 2022, pp. 135–146, http://wupes.utia.cas.cz/2022/Proceedings.pdf#page=144.

[15] F. Kratochvíl, V. Kratochvíl, J. Vomlel, Learning noisy-or networks with an application in linguistics, in: A. Salmerón, R. Rumí (Eds.), Proceedings of the 11th International Conference on Probabilistic Graphical Models, 2022, pp. 277–288.
[16] R.M. Neal, Connectionist learning of belief networks, Artif. Intell. 56 (1992) 71–113, https://doi.org/10.1016/0004-3702(92)90065-6.
[17] J. Nocedal, S.J. Wright, Numerical Optimization, second edition ed., Springer, New York, NY, 2006.
[18] K. Nowak, M.J. Druzdzel, Learning parameters in canonical models using weighted least squares, in: L.C. van der Gaag, A.J. Feelders (Eds.), Proceedings of Probabilistic Graphical Models PGM 2014, Springer, 2014, pp. 366–381.
[19] A. Onisko, M. Druzdzel, H. Wasyluk, Learning Bayesian network parameters from small data sets: application of Noisy-OR gates, Int. J. Approx. Reason. 27 (2001) 165–182, https://doi.org/10.1016/S0888-613X(01)00039-1.
[20] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.
[21] F. Rijmen, Bayesian networks with a logistic regression model for the conditional probabilities, Int. J. Approx. Reason. 48 (2008) 659–666, https://doi.org/10.1016/j.ijar.2008.01.001, In memory of Philippe Smets (1938–2005).
[22] L.K. Saul, T. Jaakkola, M.I. Jordan, Mean field theory for sigmoid belief networks, J. Artif. Intell. Res. 4 (1996) 61–76, https://doi.org/10.1613/jair.251.
[23] G. Schwarz, Estimating the dimension of a model, Ann. Stat. 6 (1978) 461–464.
[24] C. Sharma, P. van Beek, Scalable Bayesian network structure learning with splines, in: A. Salmerón, R. Rumí (Eds.), Proceedings of the 11th International Conference on Probabilistic Graphical Models, 2022, pp. 181–192.
[25] C. Sharma, Z.A. Liao, J. Cussens, P. van Beek, A score-and-search approach to learning Bayesian networks with noisy-or relations, in: Proceedings of the 10th International Conference on Probabilistic Graphical Models (PGM 2020), 2020, pp. 413–424, https://proceedings.mlr.press/v138/sharma20a.html.
[26] M. Shwe, B. Middleton, D. Heckerman, M. Henrion, E. Horvitz, H. Lehmann, G. Cooper, Probabilistic diagnosis using a reformulation of the INTERNIST–1/QMR knowledge base I. The probabilistic model and inference algorithms, Methods Inf. Med. 30 (1991) 241–255.
[27] P. Spirtes, C. Glymour, An algorithm for fast recovery of sparse causal graphs, Soc. Sci. Comput. Rev. 9 (1991) 62–72.
[28] J. Vomlel, Noisy-or classifier, Int. J. Intell. Syst. 21 (2006) 381–398, https://doi.org/10.1002/int.20141.
[29] A. Zagorecki, M.J. Druzdzel, Knowledge engineering for Bayesian networks: how common are Noisy-MAX distributions in practice?, IEEE Trans. Syst. Man Cybern. Syst. 43 (2013) 186–195, https://doi.org/10.1109/TSMCA.2012.2189880.