Learning Noisy-Or Networks with an Application in Linguistics

František Kratochvíl frantisek.kratochvil@upol.cz
Department of Asian Studies, Palacký University Olomouc

Václav Kratochvíl velorex@utia.cas.cz
Jiří Vomlel vomlel@utia.cas.cz
Institute of Information Theory and Automation, Czech Academy of Sciences

Abstract

In this paper we discuss the issue of learning Bayesian networks whose conditional probability tables (CPTs) are either noisy-or models or general CPTs. We refer to these models as Mixed Noisy-Or Bayesian Networks. In order to learn the structure of such Bayesian networks we modify the Bayesian Information Criteria (BIC) used for general Bayesian networks so that it reflects the number of parameters of a noisy-or model. We prove the log-likelihood function of a noisy-or model has a unique maximum and adapt the EM-learning method for leaky noisy-or models. We evaluate the proposed approach on synthetic data where it performs substantially better than general BNs. We apply this approach also to a problem from the domain of linguistics. We use Mixed Noisy-Or Bayesian Networks to model spread of loanwords in the South-East Asia Archipelago. We perform numerical experiments in which we compare prediction ability of general Bayesian Networks with Mixed Noisy-Or Bayesian Networks.

Keywords: Bayesian networks; Learning Bayesian networks; Noisy-or model; Applications of Bayesian networks; Linguistics; Loanwords.

1. Introduction

Bayesian networks (Pearl, 1988; Jensen, 2001) is a popular class of models for problems with uncertainty. The problem of learning the structure of Bayesian networks from data is well studied problem with many interesting results (Spirtes and Glymour, 1991; Chickering, 2002; Cussens et al., 2017). Since the general structure learning problem is known to be NP-hard optimal learning can be performed for smaller models only, although the tractability border for optimal learning keeps being shifted by sophisticated learning methods as (Cussens et al., 2017). Bayesian network models with certain local structure of its conditional probability tables (CPTs) (Díez and Druzdzel, 2006) represent a special subclass of Bayesian networks well applicable in many real problems. Much less attention was given to learning the structure of Bayesian network models with a local structure of its CPTs (Friedman and Goldszmidt, 1996). A commonly used model is the noisy-or model (Pearl, 1988; Díez and Galán, 2003; Vomlel, 2006). This model has found its way to several applications of Bayesian networks due to its natural interpretation and low number of its parameters, which is linear with respect to the number of variables in the corresponding conditional probability table. In this paper we study the problem of learning the structure of Bayesian network where the CPTs can be represented by general CPTs or noisy-or models depending on which lead to a better final 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 the method to the problem of modeling the spread of loanwords in the South-East Asia Archipelago.

Our work is closely related to Sharma et al. (2020) where the authors also consider structural learning of Mixed Noisy-Or Bayesian Networks. Our work differs in that we work with leaky noisy-or models (i.e., noisy-or models extended by a leaky probability), we use the EM learning method of Vomlel (2006) to learn parameters of noisy-or models, we prove the log-likelihood function of a noisy-or model has a unique maximum, and, finally, we use different datasets for experimental evaluations.

2. Bayesian Information Criteria for 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 *true* and *false* represented by numerical values 1 and 0, respectively. It means that $\mathcal{X}_v = \{0, 1\}$. 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. $P : \{0, 1\}^n \rightarrow [0, 1]$ 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 an edge $u \rightarrow v$ exists in graph G.

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 boldface small 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 $\mathbf{X}_{v,U}$. Then the probability of observing i.i.d. data $\mathbf{D}$ given a Bayesian network model P is:

$$\begin{aligned} L(P|\mathbf{D}) &= \prod_{\mathbf{x} \in \mathbf{D}} P(\mathbf{x}) && (1) \\ &= \prod_{\mathbf{x} \in \mathbf{D}} \prod_{v \in V} P(x_v | x_{pa(v)}) \ . && (2) \end{aligned}$$

It is referred to as likelihood of a model with respect to data $\mathbf{D}$. Assume $A \subseteq V$, then the function $N : \mathcal{X}_A \rightarrow \mathbb{N}$ 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:

$$\begin{aligned} LL(P|\mathbf{D}) &= \log \prod_{\mathbf{x} \in \mathbf{D}} P(\mathbf{x}) \quad = \quad \sum_{\mathbf{x} \in \mathbf{D}} \sum_{v \in V} \log P(x_v | \mathbf{x}_{pa(v)}) && (3) \\ &= \sum_{v \in V} \sum_{\mathbf{x} \in \{0,1\}^n} N(\mathbf{x}) \cdot \log P(x_v | \mathbf{x}_{pa(v)}) && (4) \\ &= \sum_{v \in V} LL_v(P|\mathbf{D}) \ , \text{ where} && (5) \\ LL_v(P|\mathbf{D}) &= \sum_{\mathbf{x}_{fa(v)} \in \{0,1\}^{|fa(v)|}} N(\mathbf{x}_{fa(v)}) \cdot \log P(x_v | \mathbf{x}_{pa(v)}) \ . && (6) \end{aligned}$$

This means that the log-likelihood of a Bayesian network can be computed locally, i.e. for each node v and its parents $pa(v)$ which together form family $fa(v)$.

For a CPT of a noisy-or model of a variable $X_v, v \in V$ it holds that for $\mathbf{x}_{v,pa(v)}$:

$$P(x_v|\mathbf{x}_{pa(v)}) \quad = \quad \left(p_{v,0} \cdot \prod_{j \in pa(v)} p_{v,j}^{x_j}\right)^{(1-x_v)} \cdot \left(1 - p_{v,0} \cdot \prod_{j \in pa(v)} p_{v,j}^{x_j}\right)^{(x_v)} \quad , \tag{7}$$

where $p_{v,j}$ represents the probability that the positive influence of parent X_j on its child X_v is inhibited. The parameter $p_{v,0}$ is called leaky probability and specifies the probability that node X_v takes value 1 despite all its parents have value 0. In case of noisy-or the local log-likelihood score[1] for node v can be written as

$$\begin{aligned} & LL_v^\diamond(P|\mathbf{D}) \\ & = \sum_{\mathbf{x}_{fa(v)} \in \{0,1\}^{|fa(v)|}} N(\mathbf{x}_{fa(v)}) \cdot \left(\begin{array}{l} (1-x_v) \cdot \left(\log p_{v,0} + \sum_{j \in pa(v)} x_j \log p_{v,j}\right) + \\ x_v \cdot \log\left(1 - p_{v,0} \cdot \prod_{j \in pa(v)} p_{v,j}^{x_j}\right) \end{array} \right) \quad . \end{aligned} \tag{8}$$

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 as relative frequencies from data $\mathbf{D}$, i.e. for $(x_v, \mathbf{x}_{pa(v)})$ it holds

$$P^*(x_v|pa(v)) \quad = \quad \frac{N(\mathbf{x}_{fa(v)})}{N(\mathbf{x}_{pa(v)})} \quad . \tag{9}$$

In case of noisy-or no closed form solution for the conditional probability distributions P^* that maximize the log-likelihood is known. However, due to the decomposability of the log-likelihood the estimates can still be computed locally for each node $v \in V$. In the next lemma we show that the local log-likelihood score of noisy-or is strictly concave.

Lemma 1 *The local log-likelihood score of 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 (8). 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. ■

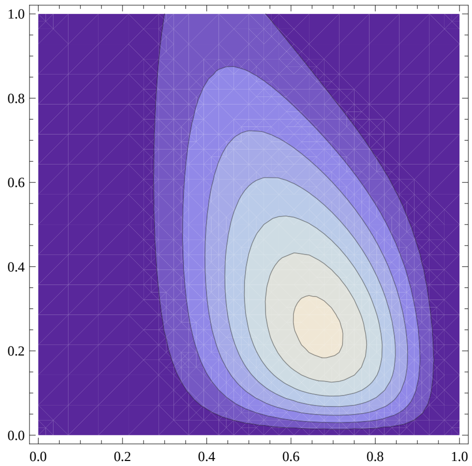
A strictly concave function has a unique maximum. See Figure 1 for a contour plot of likelihood function[2] $p_0^4 \cdot (p_0 p_1)^1 \cdot (1-p_0)^2 \cdot (1-p_0 p_1)^5$ as a function of p_0 and p_1. The horizontal axis corresponds to p_0 and the vertical axis to p_1. We plot the likelihood instead of the log-likelihood[3] since the contours are better spaced. The lighter the color the higher the value of the likelihood.

[1] We will use the diamond symbol $\diamond$ to denote the local scores of a noisy-or.

[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 of course, strictly concave as well.

Figure 1: The contour plot of a likelihood function of a noisy-or.



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. Now, we can state an important lemma about the log-likelihood function $LL^\diamond(P|\mathbf{D})$ of a Bayesian network with noisy-or models.

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

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

So far we have addressed the problem of learning parameters when the Bayesian network has its structure represented by a directed acyclic graph G. It is well-known that the mere maximization of log-likelihood leads to models that are dense and are typical examples of overfitting the training data. Actually, the Bayesian network model with the structure represented by a complete graph has always the highest value of log-likelihood. Therefore, Bayesian network scoring functions that penalize networks with complex graphs are used. In this work we will use the Bayesian Information Criterion (BIC) (Schwarz, 1978), although the presented approach can be adapted for other scoring functions as well.

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 P:

$$BIC(P|\mathbf{D}) \quad = \quad LL(P|\mathbf{D}) \quad - \quad \frac{\log|\mathbf{D}|}{2} \cdot C(P) \ . \tag{10}$$

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

$$C(P) \quad = \quad \sum_{v \in V} C_v(P(X_v|X_{pa(v)}) \ . \tag{11}$$

In case of binary variables the penalty of a general conditional probability table is

$$C_v(P(X_v|X_{pa(v)}) \quad = \quad (|\mathcal{X}_v|-1)\prod_{j\in pa(v)}|\mathcal{X}_j| \quad = \quad 2^{|pa(v)|} \ . \tag{12}$$

In case of a noisy-or model the penalty is

$$C_v^{\diamond}(P(X_v|X_{pa(v)}) \quad = \quad |pa(v)|+1 \ . \tag{13}$$

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

3. Learning Noisy-Or Bayesian Networks

It follows from the discussion presented in the previous section that learning a Noisy-Or Bayesian Network using methods based on standard penalty would typically lead to models that have a substantially lower number of parents than it is appropriate for the noisy-or models. Therefore structural learning of a Noisy-Or Bayesian Networks should be based on a modified score function. In practical applications of Bayesian networks some conditional probability tables have local structure, e.g. noisy-or, while other conditional probability tables are better represented by general conditional probability tables.

Motivated by this observation we propose a structure learning algorithm which can decide which type of conditional probability table (CPT) 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 general conditional probability table and noisy-or. We call such Bayesian networks *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 Mixed Noisy-Or Bayesian Networks in Algorithm 1. In its first phase the algorithm computes maximal likelihood estimates for all nodes v and their parent sets U. This is computed for both general CPTs and noisy-or models. To learn maximum likelihood (MLL) estimates of noisy-or parameters we use the computationally efficient version of the EM algorithm proposed by Vomlel (2006), which we adjusted for leaky noisy-or models[4]. The EM algorithm is presented in Algorithm 2 and discussed later in this section. The BIC values of general CPTs and noisy-or models are compared and the values of v, U, and the higher BIC value are stored in the list of all parent set evaluation.

Before adding a triplet (v, U, BIC) into 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 not included in the list $\mathcal{L}$. This can be safely done for a triple (v, U, BIC) such that there is a $(v, U', BIC') \in \mathcal{L}$ satisfying $U' \subset U$ and $BIC' > BIC$. In de Campos et al. (2018) several other pruning rules for general CPTs are presented. In Sharma et al. (2020) two

[4] We performed experiments also with other methods as Nedler-Mead, a box constrained BFGS, and gradient projection methods. The EM algorithm was by far the most efficient one, especially for large parent sets.

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

Algorithm 1: Learning the structure of a Mixed Noisy-Or Bayesian Network.

pruning rules for noisy-or models were proposed. The first pruning rule from Sharma et al. (2020)[Lemma 4] suggests to eliminate 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 $\mathbf{D}$. The second pruning rule from Sharma et al. (2020)[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 holds also for all triplets (v, U'', BIC_i'') with $U'' \supset U$ since the penalty can only increase with larger parent sets. The discussion on an application of these pruning rules can be found in Section 5.

The final step of the algorithm is the application of the GOBNILP method (Cussens and Bartlett, 2018). It is a program which can learn optimal Bayesian networks from local scores. It uses the SCIP framework for Constraint Integer Programming as its core routine(Cussens et al., 2017).

In Algorithm 2 we present the EM algorithm for learning maximum likelihood estimates of parameters of a noisy-or model. The algorithm is derived from the EM learning method presented in (Vomlel, 2006) and adpated for leaky noisy-or model. The algorithm alternates between $E - step$ and $M - step$ until the convergence criterion is met or a given maximum number of iteration is performed. We will use $\mathbf{p}_v$ as an abbreviation for vector $\left(p_{v,0}, (p_{v,j})_{j \in pa(v)}\right)$. The symbols $\oplus$, $\ominus$, $\otimes$, and $\oslash$ will denote pointwise addition, subtraction, multiplication, and division of two vectors, respectively. The symbol $\mathbf{p}^{\mathbf{a}}$ will denote a vector $\left(\mathbf{p}_j^{\mathbf{a}_j}\right)_{j=1}^{|pa(v)|+1}$, i.e. the pointwise exponentiation.

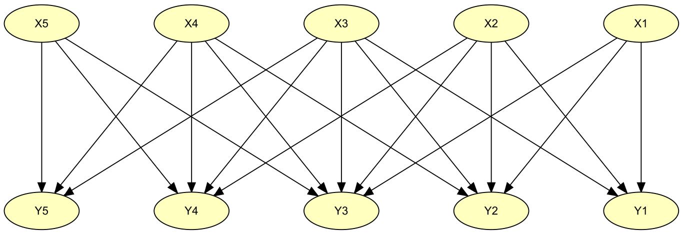
```
input  : $v$ – the child node
         $U$ – parents of node $v$
         $\mathbf{D} = \{\mathbf{x}_n\}_{n=1}^{N}$ – dataset of $N$ complete data vectors $\mathbf{x}_n$
output: $\mathbf{p}_v$ – estimated MLL parameters of noisy-or
$\ell = |U| + 1$ ;                          /* the length of considered 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 parameter values of length $\ell$ */
while $(m < M) \wedge (\delta > \Delta)$ do
    $m = m + 1$ ;                                   /* increase iteration counter */
    $\mathbf{p}'_v = \mathbf{p}_v$ ;
    $\mathbf{n}_0 = \mathbf{0}_\ell$ ; $\mathbf{n}_1 = \mathbf{0}_\ell$ ;     /* initialize with vectors of $0$ of length $\ell$ */
    for $n = 1, \ldots, N$ ;     /* E-step: for all data 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 vector of $1$ of length $\ell$ */
            $\mathbf{r}_1 = \mathbf{0}_\ell$ ;                            /* a vector of $0$ of length $\ell$ */
        end
        else
            $q = \prod_{j=1}^{\ell} \mathbf{p}_j^{\mathbf{a}_j}$ ;                   /* the product of values in $\mathbf{p}^{\mathbf{a}}$ */
            $\mathbf{r}_0 = \mathbf{p}^{\mathbf{a}} \ominus \mathbf{q}_\ell$ ;    /* $\mathbf{q}_\ell$ is the vector of length $\ell$ 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 $\mathbf{r}_0$, define $0/0 = 0$ */
            $\mathbf{r}_1 = \mathbf{r}_1 \oslash \mathbf{r}$ ; /* pointwise normalization of $\mathbf{r}_1$, define $0/0 = 0$ */
        end
        $\mathbf{n}_0 = \mathbf{n}_0 \oplus (\mathbf{a} \odot \mathbf{r}_0)$ ; /* pointwise addition of a pointwise product */
        $\mathbf{n}_1 = \mathbf{n}_1 \oplus (\mathbf{a} \odot \mathbf{r}_1)$ ; /* pointwise addition of a pointwise 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 2: The EM-algorithm for the leaky noisy-or model.

4. Experiments with a Synthetic Data

In this section we will describe an experiment we have used to verify that the proposed algorithm can identify noisy-or models correctly. We will also compare the predictive performance of the learned model with the model learned by maximizing BIC score without considering noisy-or models. We created a BN2O network[5], which is a Bayesian network consisting of two layers of nodes. All edges are directed from the top layer to the bottom layer. No edges connecting nodes from the same layer exist. The nodes from the second layer share some parents but not all of them. The structure used in the experiments is presented in Figure 2. All conditional probability tables are noisy-or models.

Figure 2: BN2O network structure.



The experiments confirmed our expectation that the mixed BIC optimization, which considers both the standard CPTs and noisy-or models in its search and penalizes likelihood accordingly, is able to identify the correct Bayesian network structure for much smaller training datasets. See left hand side of Figure 3 for results on training datasets of different size[6] (please, note the log scale of axis x).

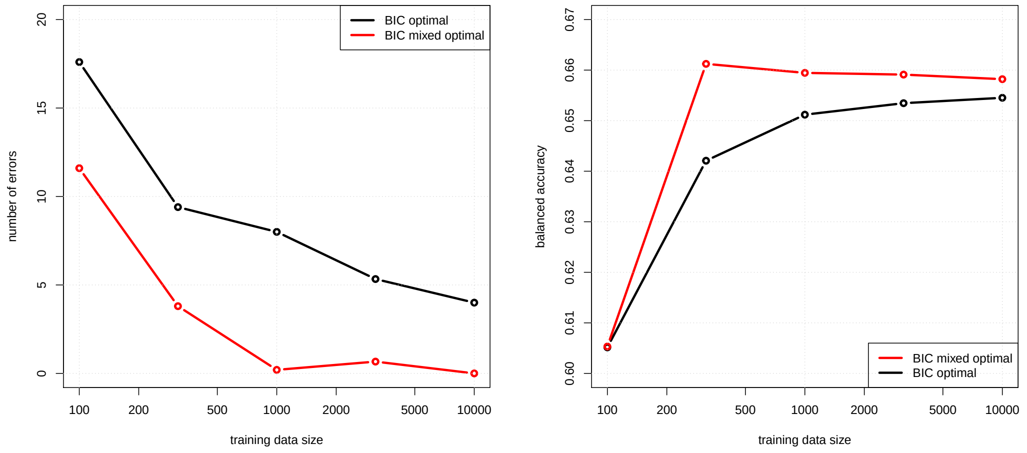
One of the tasks for which Bayesian networks are used is the prediction of states of certain variables given observations of some other variables in the model. It can be expected that models having the structure similar to the structure of the original model can perform better, however, sometimes simple model 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 variables of our BN2O models are typically imbalanced (a state is significantly more probable than another) we decided to use balanced accuracy[7] as our evaluation criteria. On the right hand side of Figure 3 we present average results for the task when evidence was inserted into the model for five randomly selected variables and states of other five variables were predicted for the BIC optimal and the BIC mixed optimal methods as a function of the training data size (please, note the log scale). The experiments confirm that especially for smaller training datasets Mixed Noisy-OR Bayesian Networks have better prediction ability.

[5] These networks are common in practical applications of BNs, e.g., in medical and educational domains.

[6] First, we generated a dataset consisting of 10000 data records. Then we split this dataset to smaller datasets so that each vector from the original dataset was used only once in the datasets of the same size. The datasizes are chosen so that they cover well the interesting cases, namely, they correspond to the rounded geometric sequence $10^{(2+i/2)}, i = 0, 1, \ldots, 4$.

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

Figure 3: The number of wrong edges in learned models (missing, reversed, or additional) (on the left) and the balanced accuracy (on the right).

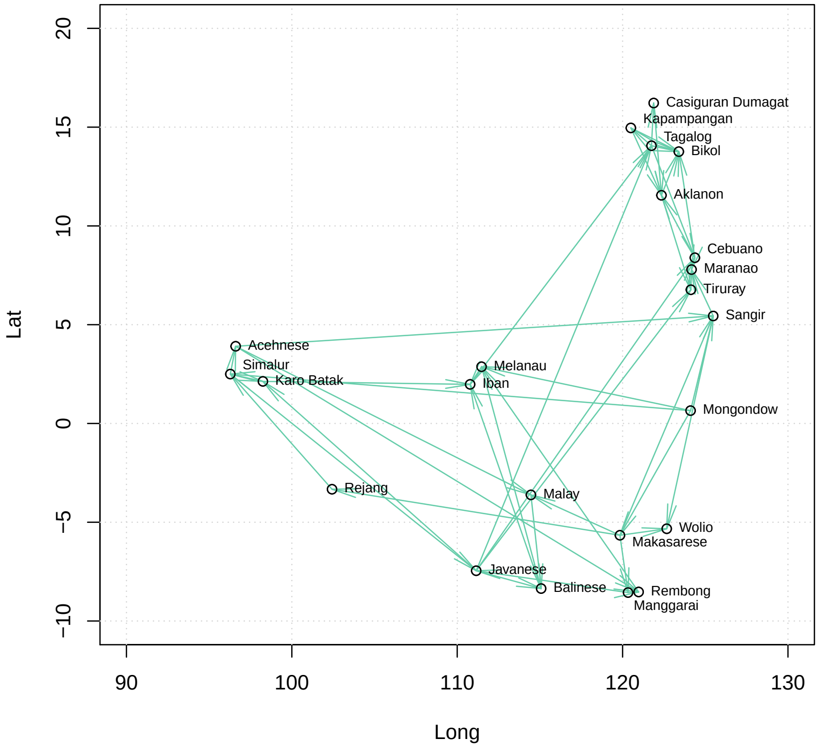


5. Application to Modeling the Spread of Loanwords

The main idea of this paper – learning Mixed Noisy-Or Bayesian Networks – was motivated by our collaboration on a research project from the area of linguistics on modeling the spread of loanwords in the area of the South-East Asia Archipelago. This is a region specific for a large number of languages, which is caused by its character of thousands of islands. A loanword is a word permanently adopted from one language and incorporated into another language without translation. Since written records and archaeological evidence are missing in this region, the distribution of loanwords offers an insight into past human migrations, contacts, and trade. Our primary resource is a large database of loanwords collected from several sources. The database is available at `http://gogo.utia.cas.cz/loanwords/`. In our experiments reported in this paper we have used a dataset providing information about presence/absence of 461 loanwords in 23 languages. All studied loanwords originated from one of eleven donor languages that differ from the studied 23 recipient languages. A detailed description of the studied problem together with other results of our experimental analysis was presented recently in (Kratochvíl et al., 2022).

The task is to learn a Bayesian network having languages from the studied region as its variables. We studied the problem with 23 languages. All variables are binary with states 0 and 1 representing absence and presence of a loanword in the corresponding language, respectively. Noisy-or models seems to be a natural model for this problem. Particularly, it means that the presence of a loanword in related languages represented by parent variables increases probability of that loanword being present in the language of the child variable. However, preliminary results revealed that the assumption of all conditional probability tables being represented by noisy-or models worsened the performance. This lead to the idea to let the learning algorithm decide for each CPT whether the noisy-or or general CPT represents a better fit. Unfortunately, we faced the problem of very large number of parent

Figure 4: Mixed Noisy-Or Bayesian Networks modeling the spread of loanwords.



sets since the pruning rules from (Sharma et al., 2020) were not efficient in this application. For example, the first pruning rule from Sharma et al. (2020)[Lemma 4] suggests to eliminate 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). A natural next step seems to be the design of new pruning rules for noisy-or models.

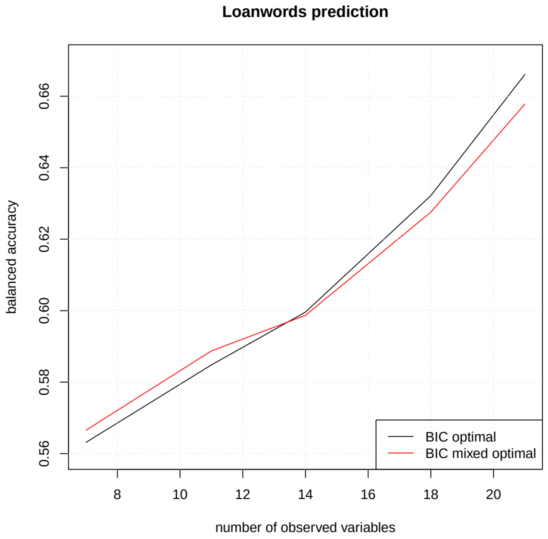
We decided to proceed using a heuristic pruning method that discarded all supersets of a parent set that had a lower BIC score than its subset, i.e. we pruned out all triplets (v, U'', BIC'') if $\exists(v, U, BIC) \in \mathcal{L}, U \subset U''$ which was pruned since there is a $(v, U', BIC') \in \mathcal{L}$ satisfying $U' \subset U$ and $BIC' > BIC$. This approach does not guarantee optimality but helped us to reduce significantly the list of triplets $\mathcal{L}$. In the learned BNs [8] 57% of CPTs were represented by noisy-or models.

In Figure 4 we present the structure of one of the learned Mixed Noisy-Or Bayesian Networks. The positions of the nodes correspond to the geographical coordinates of the studied languages. It is interesting to see that the edges most often connect neighbor languages but there are also few edges between remote places. This could be potentially explained by historical trade routes but this is a hypothesis to be further studied by linguist and historians of this region.

In Figure 5 we present results of our experiments with the database of loanwords. The ten-fold cross-validation was used to evaluate the models. The balanced accuracy is dis-

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

Figure 5: Balanced accuracy for the BIC optimal and the BIC mixed optimal methods as a function of the number of nodes with evidence.



played as a function of the number of variables with evidence[9]. The more variables are observed the better is the prediction quality. Both methods perform comparably and none of them is a clean winner. We conjecture that the Mixed Noisy-Or Bayesian Networks better describes the studied problem but this requires further verification. Also, optimal Mixed Noisy-Or Bayesian Networks may posses better prediction quality than the suboptimal ones.

6. Conclusions and Open Problems

We studied learning of Mixed Noisy-Or Bayesian Networks. The discussed learning method can be extended to other models of the local structure of CPTs if their maximum likelihood estimates can be found efficiently. We proved the log-likelihood function of a noisy-or model has a unique maximum and adapted the EM-learning method of Vomlel (2006) for learning leaky noisy-or models. We evaluated the proposed approach on synthetic data where it performed substantially better than general BNs. We applied the method to the problem of modeling of 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. From the theoretical point of view we have left open the problem of efficient pruning rules for noisy-or models.

Acknowledgments

This work was supported by the Czech Science Foundation Project Nr. 20-18407S.

[9] For each vector from the testing dataset the evidence nodes were chosen randomly.

References

D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

J. Cussens and M. Bartlett. GOBNILP, 2018. Version 1.6.3, `https://www.cs.york.ac.uk/aig/sw/gobnilp/`.

J. Cussens, M. Järvisalo, J. H. Korhonen, and M. Bartlett. Bayesian network structure learning with integer programming: Polytopes, facets and complexity. *Journal of Artificial Intelligence Research*, 58:185–229, 2017.

C. P. de Campos, M. Scanagatta, G. Corani, and M. Zaffalon. Entropy-based pruning for learning Bayesian networks using BIC. *Artificial Intelligence*, 260:42–50, 2018.

F. J. Díez and M. J. Druzdzel. Canonical probabilistic models for knowledge engineering. Technical Report CISIAD-06-01, UNED, Madrid, Spain, 2006.

F. J. Díez and S. F. Galán. An efficient factorization for the noisy MAX. *International Journal of Intelligent Systems*, 18:165–177, 2003.

N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 252–262, 1996.

F. V. Jensen. *Bayesian Networks and Decision Graphs.* Springer-Verlag, New York, 2001.

F. Kratochvíl, V. Kratochvíl, G. Saad, and 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)*, pages 135–146. MatfyzPress, 2022. URL `http://wupes.utia.cas.cz/2022/Proceedings.pdf#page=144`.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

C. Sharma, Z. A. Liao, J. Cussens, and 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)*, volume 138 of *Proceedings of Machine Learning Research*, pages 413–424, 2020. URL `https://proceedings.mlr.press/v138/sharma20a.html`.

P. Spirtes and C. Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9:62–72, 1991.

J. Vomlel. Noisy-or classifier. *International Journal of Intelligent Systems*, 21:381–398, 2006. URL `https://doi.org/10.1002/int.20141`.