Learning Sets of Bayesian Networks

Andrés Cano, Manuel Gómez-Olmedo, and Serafín Moral

Departamento de Ciencias de la Computación e Inteligencia Artificial,
18071 Granada, Spain
{acu,mgomez,smc}@decsai.ugr.es

Abstract. This paper considers the problem of learning a generalized credal network (a set of Bayesian networks) from a dataset. It is based on using the BDEu score and computes all the networks with score above a predetermined factor of the optimal one. To avoid the problem of determining the equivalent sample size (ESS), the approach also considers the possibility of an undetermined ESS. Even if the final result is a set of Bayesian networks, the paper also studies the problem of selecting a single network with some alternative procedures. Finally, some preliminary experiments are carried out with three small networks.

Keywords: Generalized credal networks · Learning · Likelihood regions · Probabilistic graphical models

1 Introduction

Probabilistic graphical models [17] and in particular Bayesian networks have been very successful for representing and reasoning in problems with several uncertain variables. The development of procedures to learn a Bayesian network from a dataset of observations [16] is one the most important reasons of this success. Usually, learning is carried out by selecting a score measuring the adequacy of a model given the data and optimizing it in the space of models. However, in most of the situations the selection of a single Bayesian network is not justified as there are many models that explain the data with a similar degree, being the selection of an optimal network a somewhat arbitrary choice [7]. For this reason, recently, there has been some effort in computing a set of possible models instead of selecting a single one [12]. The idea is to compute all the models with a score within a given factor of the optimal one. In this paper we will follow this line, but interpreting the result as a generalized credal network: a set of Bayesian networks which do not necessarily share the same graph [13]. The term credal network was introduced [6] for a set of Bayesian networks over a single graph (there is imprecision only in the parameters). The overall procedure is based on the general framework introduced in [15], where it is proposed a justification

This research was supported by the Spanish Ministry of Education and Science under project TIN2016-77902-C3-2-P, and the European Regional Development Fund (FEDER).

M.-J. Lesot et al. (Eds.): IPMU 2020, CCIS 1238, pp. 151–164, 2020.
https://doi.org/10.1007/978-3-030-50143-3_12

based on sets of desirable gambles [5,18,22] for the selection of a set of models instead of a single one, following the lines of Gärdenfors and Shalin [8] and α-cut conditioning by Cattaneo [2].

The basic criterion used for learning is the so called BDEu score [9]. This score needs a parameter, the equivalent sample size (ESS), which is usually arbitrarily selected in practice with a value between 1 and 10. However, there are results showing that the final network can have a dependence on the ESS, producing more dense networks when it is higher [4,14,21]. For this reason, our approach will also consider the possibility of imprecision due to an undetermined ESS.

The paper is organized as follows. Section 2 provides the basic theoretical framework for our problem. Section 3 describes the algorithms used in the computation. Section 4 is devoted to the experiments. Finally, the conclusions and future work are in Sect. 5.

2 Learning Imprecise Models

Given a set of variables, $\mathbf{X} = (X_1, \ldots, X_m)$, a Bayesian network [17] is a pair (G, β), where G is a directed acyclic graph such that each node represents a variable, and β is the set of parameters: a conditional probability distribution for each variable X_i given its parents in the graph, Pa_i, denoted as $P(X_i|Pa_i)$ or as $P_{(G,\beta)}(X_i|Pa_i)$ when we want to make reference to the associated model. It will be assumed that each variable X_i takes values on a finite set with K_i possible values. A generic value for variable X_i is denoted by x_i and a generic value for all the variables $\mathbf{X}$ is denoted as $\mathbf{x}$. An assignation of a concrete value to each variable in Pa_i is called a configuration and denoted as pa_i. The number of possible configurations of Pa_i is denoted by R_i. There is a joint probability distribution for variables $\mathbf{X}$ associated with a Bayesian network (G, β) that will be denoted as $P_{(G,\beta)}$ and that is equal to the product $\prod_{i=1}^{m} P_{(G,\beta)}(X_i|Pa_i)$.

We will consider that we have a set of full observations $\mathcal{D}$ for all the variables in $\mathbf{X}$. Given a graph G, n_{ijk} will denote the number of observations in $\mathcal{D}$ where $X_i = x_k$ and its parents Pa_i take the jth configuration, $n_{ij} = \sum_{k=1}^{K_i} n_{ijk}$, whereas n will be the total sample size. In the framework for learning proposed in [15], it is assumed that we have the following elements:

- A set of parameters Θ that corresponds to the space of possible decisions. In our case, Θ is the set of pairs (G, s), where G is a direct acyclic graph, and s is a possible ESS belonging to a finite set of values, S. For example, in our experiments we have considered $S = \{0.1, 0.5, 1.0, 2.0, 4.0, 6.0, 8.0, 10.0\}$. We assume a finite set instead of a continuous interval for computational reasons.
- A set of parameters B, and a conditional probability distribution $P(\beta|\theta)$ specifying the probability on B for each value of the parameter $\theta \in \Theta$. In our case the set B is the list of conditional probability distributions $P(X_i|Pa_i)$, where the probability values of the conditional distribution of X_i given the jth configuration of the parents are denoted by $\beta_{ij} = (\beta_{ij1}, \ldots, \beta_{ijK_i})$ (i.e. $\beta_{ijk} = P(X_i = x_k|Pa_i = pa_i^j)$). It is assumed that each β_{ij} follows an independent

Dirichlet distribution $D(s/(R_iK_i), \ldots, s/(R_iK_i))$. The set of all parameters β_{ij} will be denoted by β.

- A conditional distribution for an observation of the variables $(X_1, \ldots, X_m)$ given a pair $(\theta, \beta) \in \Theta \times B$ (in our case, given G, s and β). The probability of observing $X_1 = x_{k_1}, \ldots, X_m = x_{k_m}$ is the product $\prod_{i=1}^m \beta_{ijk_i} = \prod_{i=1}^M P(x_{k_i}|pa_i^j)$, where pa_i^j is the configuration of the parents compatible with the observation, and k_i represents the subscript of the observed value for $X_i = x_{k_i}$.

In this setting, a set of observations $\mathcal{O}$, defines a likelihood function L in Θ, which is given at the general case by,

$$L(\theta) = \int_B P(\beta|\theta)P(\mathcal{D}|\beta, \theta)d\beta. \tag{1}$$

In the particular case of learning generalized credal networks, we have that this likelihood is identical to the well known BDEu score [9] for learning Bayesian networks:

$$L(G, s) = BDEu(G, s) = P(\mathcal{D}|G) = \prod_{i=1}^m \prod_{j=1}^{R_i} \frac{\Gamma(s/R_i)}{\Gamma(n_{ij} + s/R_i)} \prod_{k=1}^{K_i} \frac{\Gamma(n_{ijk} + s/(R_iK_i))}{\Gamma(s/(R_iK_i))}. \tag{2}$$

The score for Pa_i as set of parents of X_i is the value:

$$\log(BDEu(Pa_i, s, X_i)) = \log \left(\prod_{j=1}^{R_i} \frac{\Gamma(s/R_i)}{\Gamma(n_{ij} + s/R_i)} \prod_{k=1}^{K_i} \frac{\Gamma(n_{ijk} + s/(R_iK_i))}{\Gamma(s/(R_iK_i))} \right). \tag{3}$$

It is immediate that $\log(BDEu(G, s)) = \sum_{i=1}^m \log(BDEu(Pa_i, s, X_i))$. Finally, a generalized uniform distribution on Θ is considered given in terms of a coherent set of desirable gambles [15]. When Θ is finite as in this case, associated credal set only contains the uniform probability, but when Θ is a continuous interval is quite different from the usual uniform density. Then a discounting is considered of this prior information on Θ given by a value $\epsilon \in [0, 1]$. This discounting is a generalization of the ϵ discounting of a belief function [20]. After the observations are obtained, the model is conditioned to them, obtaining a posterior information on Θ. It is assumed that the set of decisions is equivalent to the set of parameters Θ and the problem is solved by computing all un-dominated decisions under a 0–1 loss (details in [15]). Finally, in our case the set of un-dominated decisions is the set of parameters (G, s) such that:

$$L(G, s) = BDEu(G, s) \geq \alpha BDEu(\hat{G}, \hat{s}),$$

where $\alpha = \frac{1-\epsilon}{1+\epsilon} \in [0, 1]$ and $(\hat{G}, \hat{s})$ is the pair maximizing the likelihood $L(G, s)$ for $(G, s) \in \Theta$. The set of parameters satisfying the above inequality is denoted by H_L^α and defines what we shall call the set of possible models.

In the following, we will use the value $\alpha = \frac{1-\epsilon}{1+\epsilon} \in [0,1]$ which is computed as a continuous decreasing function from $[0,1]$ into $[0,1]$ and that determines the factor of the maximum entropy model which makes (G,S) non-dominated.

Given a parameter (G,s), the model for $\mathbf{X}$ is given by the Bayesian network $(G,\hat{\beta})$, where $\hat{\beta}$ is the Bayesian estimation of β (expected value of β given (G,s) and $\mathcal{D}$), and which can be computed in closed form by the well known expression:

$$\hat{\beta}_{ijk} = \frac{n_{ijk} + s/(R_i K_i)}{n_{ij} + s/(R_i)}. \tag{4}$$

The probability distribution associated with $(G,\hat{\beta})$ will be also denoted as $P_{(G,s)}(\mathbf{x})$[1]. Finally the set of possible models (a generalized credal network) is the set of Bayesian networks:

$$M_{\mathcal{D}}^{\alpha} = \{(G,\hat{\beta}) \mid (G,s) \in H_L^{\alpha}\}, \tag{5}$$

and where $\hat{\beta}$ is the set of parameters given by Eq. (4).

Though, in our opinion, the result of learning should be the set $M_{\mathcal{D}}^{\alpha}$, in some cases, it is interesting to select a single model. For example, we have carried out experiments in which we want to compare this approach to learning with a Bayesian procedure that always selects a single network. For this aim we have considered two approaches:

- *Maximum Entropy*: We select the pair $(G,\hat{\beta}) \in M_{\mathcal{D}}^{\alpha}$ maximizing the entropy, where the entropy of a model (G,β) is given by:

$$E(G,\beta) = -\sum_{\mathbf{x}} P_{(G,\beta)}(\mathbf{x}) \log P_{(G,\beta)}(\mathbf{x}). \tag{6}$$

- *Minimum of Maximum Kullback-Leibler Divergence*: If (G,β) and (G',β') are two models, then the Kullback-Leibler divergence of (G,β) to (G',β') is given by the expression:

$$KL((G',\beta'),(G,\beta)) = \sum_{\mathbf{x}} P_{(G',\beta')}(\mathbf{x}) \log\left(\frac{P_{(G',\beta')}(\mathbf{x})}{P_{(G,\beta)}(\mathbf{x})}\right). \tag{7}$$

  Then, for each model $(G,\hat{\beta}) \in M_{\mathcal{D}}^{\alpha}$, the following value is computed:

$$MKL(G,\hat{\beta}) = \max\{KL((G',\hat{\beta}'),(G,\hat{\beta})) \mid (G',\hat{\beta}') \in M_{\mathcal{D}}^{\alpha}\}.$$

  Finally, the model $(G,\hat{\beta}) \in M_{\mathcal{D}}^{\alpha}$ minimizing $MKL(G,\hat{\beta})$ is selected.

[1] In fact, this probability also depends on $\mathcal{D}$, but we do not include it to simplify the notation.

3 Algorithms

Given a set of observations $\mathcal{D}$ and a value of ϵ, our aim is to compute the set of Bayesian networks given by Eq. (5), where $\alpha = \frac{1-\epsilon}{1+\epsilon}$. For this we have taken as basis the `pgmpy` package in which basic procedures for inference and learning with Bayesian networks are implemented [1].

Our first algorithm `AllScores`(ESS, α) computes the set of possible parents as well as the logarithm of their scores for each variable X_i, for each sample size $s \in S$ and for a given value of α, being denoted this set as $Par(X_i, s, \alpha)$.

To do it, we compute the value of $\log(BDEu(Pa_i, s, X_i))$ following Eq. (3) for each set $Pa_i \subseteq \mathbf{X} \setminus \{X_i\}$, storing the pair $(Pa_i, \log(BDEu(Pa_i, s, X_i))$, but taking into account the following pruning rules as in [12]:

- If $Pa_i \subset Pa_i'$ and $\log(BDEu(Pa_i, s, X_i)) > \log(BDEu(Pa_i', s, X_i)) - log(\alpha)$, then Pa_i' is not added to $Par(X_i, s, \alpha)$ as there can not be a model in $M_{\mathcal{D}}^{\alpha}$ with this set of parents.
- If $Pa_i \subset Pa_i'$ and $\log(BDEu(Pa_i, s, X_i)) > \log(K_i) + R_i^+(Pa_i') - \log(\alpha)$, where $R_i^+(Pa_i')$ is the number of configurations of Pa_i' with $n_{ij} > 0$, then Pa_i' is not added to $Par(X_i, s, \alpha)$ and none of the supersets of Pa_i' is considered as possible set of parents for X_i.

Once `AllScores`(S, α) computes $Par(X_i, s, \alpha)$ for any $s \in S$ and any variable X_i, then an A* algorithm is applied to compute all the order relationships σ in $\{1, \ldots, m\}$ and values $s \in S$ such that there is a pair (G, s) in H_s^{α} with $Pa_{\sigma(i)} \subseteq \{X_{\sigma(1)}, \ldots, X_{\sigma(i-1)}\}$. For this we introduce two modifications of the algorithm proposed in [10]: a value α has been considered and we compute not only the order with the optimal value but also all the orders within a factor α of the optimal one.

A partial order σ_k is given by the values $(\sigma_k(1), \ldots, \sigma_k(k))$ for $k \leq m$ (only the first k values are specified). A partial order can be defined for values $k = 0, \ldots, m$. For $k = m$ we have a full order and for $k = 0$ the empty order. A graph G is compatible with a partial order σ_k when $Pa_{\sigma_k(i)} \subseteq \{X_{\sigma_k(1)}, \ldots, X_{\sigma_k(i-1)}\}$ for $i = 1, \ldots, k$. Given an ESS s and σ_k, it is possible to give an upper bound for the logarithm of the score of all the orders σ that are extensions of σ_k and which is given by,

$$\begin{aligned} Bound(\sigma_k, s) = & \sum_{i=1}^{k} Best(X_{\sigma(i)}, \{X_{\sigma(1)}, \ldots, X_{\sigma(i-1)}\}, s) + \\ & \sum_{i=k+1}^{m} Best(X_{\sigma(i)}, \mathbf{X} \setminus \{X_{\sigma(i)}\}, s), \end{aligned} \tag{8}$$

where $Best(X_i, A, s)$ is the best score stored in $Par(X_i, s, \alpha)$, between those set of parents $Pa_i \subseteq A$, i.e. we select the parents compatible with partial order σ_k for variables $X_{\sigma_k(1)}, \ldots, X_{\sigma_k(k)}$ and the rest of the set of parents are chosen in an arbitrary way.

Our algorithm is applied to nodes $N(A_k, s, score, up)$, where A_k is a set $\{\sigma_k(1), \ldots, \sigma_k(k)\}$ for a partial order σ_k, s is a value in S, $score$ is the value of $Bound(\sigma_k, s)$, and up is a reference to the node $N(A_{k-1}, s, score', up')$ such that σ_k is obtained by extending partial order σ_{k-1} with the value $\sigma_k(k)$.

The A* algorithm is initiated with a priority queue with a node for each possible value of $s \in S$, $N(\emptyset, s, score, NULL)$, where $score$ is obtained by applying Eq. (8) to partial order σ_0 (empty partial order). The algorithm stores a value B which is the best score obtained so far for a complete order (which is equal to the score of the first complete order selected from the priority queue) and $H(A, s)$ which is the best score obtained so far for a node $N(A_k, s, score, up)$ where $A_k = A$. Let us note that for a node $N(A_k, s, score, up)$ it is always possible to recover its corresponding partial order σ_k as we have that $\sigma_k(k) = A_k \setminus A_{k-1}$ where A_{k-1} is the set appearing in node $N(A_{k-1}, s, score', up')$ referenced by up, and the rest of values can be recursively found by applying the same operation of the node $N(A_{k-1}, s, score', up')$.

The algorithm proceeds selecting the node with highest $score$ from the priority queue while the priority queue is not empty and $score \geq B + \log(\alpha)$. If this node is $N(A_k, s, score, up)$, then if it is complete ($A_k = \mathbf{X}$), the node is added to the set of solution nodes. In the case it is not complete then all the nodes $N(A_{k+1}, s, score', up')$ obtained by adding one variable X_l, in $\mathbf{X} \backslash A_k$ to A_k are computed, where up' points to former node $N(A_k, s, score, up)$ and the value of $score'$ is calculated taking into account that

$$score' = score + Best(X_l, A_k, s) - Best(X_l, \mathbf{X} \setminus \{X_{\sigma(k+1)}\}, s). \tag{9}$$

The new node is added to the priority queue if and only if $score' \geq H(A_{k+1}, s) + \log(\alpha)$. In any case the value of $H(A_{k+1}, s)$ is updated if $score' > H(A_{k+1}, s)$.

Once A* is finished, we have a set of solution nodes $N(\mathbf{X}, score, s, up)$. For each one of these nodes we compute their associated order σ and then the order is expanded in a set of networks. Details are given in Algorithm 1. In that algorithm, $ParOrder(X_{\sigma(k)}, s, \alpha, \sigma)$ is the set of pairs $(Pa_{\sigma(k)}, t) \in Par(X_{\sigma(k)}, s, \alpha)$ such that $Pa_{\sigma(k)} \subseteq \{X_{\sigma(1)}, \ldots, X_{\sigma(k-1)}\}$, and t is $\log(BDEu(Pa_{\sigma(k)}, s, X_{\sigma(k)}))$.

The algorithm is initially called with a list L with a pair (G, u) where G is the empty graph, u is the value of $score$ in the solution node $N(\mathbf{X}, score, s, up)$ and with $k = 1$. It works in the following way: it considers pairs (G, u), where G is a partial graph (parents for variables $X_{\sigma(1)}, \ldots, X_{\sigma(k-1)}$ have been selected, but not for the rest of variables) and u is the best score that could be achieved if the optimal selection of parents is done for the variables $X_{\sigma(k)}, \ldots, X_{\sigma(m)}$. Then, the possible candidates for parents of variable $X_{\sigma(k)}$ are considered. If $Pa_{\sigma(k)}$ is a possible candidate set with a score of t and the optimal set of parents for this variable is T, then if this parent set is chosen, then $T - t$ is lost with respect to the optimal selection. If u was the previous optimal value, now it is $u' = u - T + t$. This set of parents can be selected only if $u' \geq B + \log(\alpha)$; in that case, the new graph G' obtained from G by adding links from $Pa_{\sigma(k)}$ to $X_{\sigma(k)}$ is considered

Algorithm 1. Computing the networks associated to an order and ESS s

```
Require: $\sigma$, an order of the variables
Require: $\alpha$, the factor of the optimal score
Require: $B$, the best score of any network
Require: $s$, the ESS
Require: $L$ a list of pairs $(G, u)$ where $G$ is a partial graph and $u$ is the best score of
    a completion of $G$
Require: $k$ the node to expand
Ensure: $LR = \{(G_1, \ldots, G_k\}$, a list of graphs with an admissible score compatible
    with $\sigma$
 1: procedure Expand($\sigma$,$\alpha$,$B$,$s$,$L$,$k$)
 2:     if $k > m$ then
 3:         Let $LR$ the list of graphs $G$ such that $(G, u) \in L$
 4:         return $LR$
 5:     end if
 6:     Let $L'$ equal to $\emptyset$
 7:     Let $T = \max\{t : (Pa_{\sigma(k)}, t) \in ParOrder(X_{\sigma(k)}, s, \alpha, \sigma)\}$
 8:     for $(G, u) \in L$ do
 9:         Let $Q$ be the set of pairs $(Pa_{\sigma(k)}, t) \in ParOrder(X_{\sigma(k)}, s, \alpha, \sigma)$ with
10:                 $u - T + t \geq B + \log(\alpha)$
11:         for $(Pa_{\sigma(k)}, t) \in Q$ do
12:             Let $G'$ the graph $G$ expanded with links from $Pa_{\sigma(k)}$ to $X_{\sigma(k)}$
13:             Let $u' = u - T + t$
14:             Add $(G', u')$ to $L'$
15:         end for
16:     end for
17:     return Expand($\sigma$,$score$,$B$,$s$,$L'$,$k + 1$)
18: end procedure
```

with optimal value u'. The algorithm proceeds by expanding all the new partial graphs obtained this way, by assigning parents to the next variable, $X_{\sigma(k+1)}$.

Finally we compute the list of all the graphs associated to the result of the algorithm for any solution node $N(\mathbf{X}, score, s, up)$ with the corresponding value s. In this list, it is possible that the same graph is repeated with identical value of s (the same graph can be obtained with two different order of variables). To avoid repetitions a cleaning step is carried out in order to remove the repetitions of identical pairs (G, s). This is the final set of non-dominated set of parameters H_L^α. Finally the set of possible models $M_\mathcal{D}^\alpha$ is the set of Bayesian networks $(G, \hat{\beta})$ that are computed for each pair $(G, s) \in H_L^\alpha$ where $\hat{\beta}$ has been obtained by applying Eq. (4).

The number of graphs compatible with an order computed by this algorithm can be very large. The size of L is initially equal to 1, and for each variable $X_{\sigma(k)}$ in $1, \ldots, m$, this number is increased in the different calls to Expand(σ,α,B,s,L,k). The increasing depends of the number of set of parents in Q (computed in lines 9–10 of the algorithm). If for each, (G, u), we denote by $NU(G, u)$ the cardinality of Q, then the new cardinality of L' is given by:

$$\sum_{(G,u)\in L} NU(G,u)$$

Observe that if $NU(G,u)$ is always equal to k, then the final number of networks is $O(k^m)$, and the complexity is exponential. However, in the experiments we have observed that this number is not very large (in the low size networks we have considered) as the cardinality of sets Q is decreasing for most of the pairs (G,u) when k increases, as the values u' associated with the new pairs (G',u') in L' are always lower than the value u in the pair (G,u) giving rise to them (see line 13 in the algorithm).

Above this, we have implemented some basic methods for computing the entropy of the probability distribution associated with a Bayesian network $E(G,\beta)$ and the Kullback-Leibler divergence from a model (G,β) to another one (G',β') given by $KL((G',\beta'),(G,\beta))$. For that, following [11, Theorem 8.5], we have implemented a function computing $ELL((G',\beta'),(G,\beta))$ given by:

$$ELL((G',\beta'),(G,\beta)) = \sum_{\mathbf{x}} P_{(G',\beta')}(\mathbf{x}) \log(P_{(G,\beta)}(\mathbf{x})).$$

For this computation, we take into account that $P_{(G,\beta)}(\mathbf{x}) = \prod_{i=1}^m P_{(G,\beta)}(x_i|pa_i)$, where pa_i is a generic configuration of the parents Pa_i of X_i in G, obtaining the following expression:

$$ELL((G',\beta'),(G,\beta)) = \sum_{i=1}^{m} \sum_{pa_i} P_{(G',\beta')}(x_i,pa_i) \log(P_{(G,\beta)}(x_i|pa_i)).$$

In this expression, $P_{(G,\beta)}(x_i|pa_i)$ is directly available in Bayesian network (G,β), but $P_{(G',\beta')}(x_i,pa_i)$ is not and have to be computed by means of propagation algorithms in Bayesian network (G',β'). This is done with a variable elimination algorithm for each configuration of the parents $Pa_i = pa_i$, entering it as evidence and computing the result for variable X_i without normalization. This provides the desired value $P_{(G',\beta')}(x_i,pa_i)$.

Finally, the values of entropy and Kullback-Leibler divergence are computed as follows:

$$E(G,\beta) = -ELL((G',\beta'),(G,\beta))$$
$$KL((G',\beta'),(G,\beta)) = ELL((G',\beta'),(G',\beta)) - ELL((G',\beta'),(G,\beta))$$

4 Experiments

To test the methods proposed in this paper we have carried out a series of experiments with 3 small networks obtained from the Bayesian networks repository in `bnlearn` web page [19]. The networks are: Cancer (5 nodes, 10 parameters), Earthquake (5 nodes, 10 parameters), and Survey (6 nodes, 21 parameters). The main reason for not using larger networks was the complexity associated to compute the Kullback-Leibler divergence for all the pairs of possible models. This is a really challenging problem, as if the number of networks is T, then $T(T-1)$ divergences must be computed, and each one of them, involves a significant number of propagation algorithms computing joint probability distributions. So, at this stage the use of large networks is not feasible to select the network with minimum maximum KL divergence to the rest of possible networks.

4.1 Experiment 1

In this case, we have considered a set of possible values for ESS, $S = \{0.1, 0.5,\ 1.0, 2.0,\ 4.0, 6.0, 8.0, 10.0\}$, and we have repeated 200 times the following sequence:

- A dataset of size 500 is simulated from the original network.
- The set of possible networks is computed with a value of $\alpha = e^{-0.6}$.
- The maximum entropy network (MEntropy), the minimum of maximum Kullback-Leibler divergence (MinMaxKL), and the maximum score network for all the sample sizes (Bayesian) are computed. For all of them the Kullback-Leibler divergence with the original one are also computed, as well as the maximum (MaxKL) and minimum divergence (MinKL) of all the possible models with the original one.

The means of the divergences of the estimated models can be seen in Table 1. We can observe as the usual method for learning Bayesian networks (considering the graph with highest score) gives rise to a network with a divergence between the maximum and minimum of the divergences of all the possible networks, and that the average is higher than the middle of the interval determined by the averages of the minimum and the maximum. This supports the idea that the Bayesian procedure somewhat makes an arbitrary selection among a set of networks that are all plausible given the data. This idea is also supported by Fig. 1 in which the density of the Bayesian, MinKL, and MaxKL divergences are depicted for each one of the networks[2]. On it we can see the similarities between the densities of these three values: of course the MinKL density is a bit biased to the left and MaxKL density to the right, being the Bayesian density in the middle, but with very small differences. This again supports the idea that all the computed models should be considered as result of the learning process.

When selecting a single model, we also show that our alternative methods based on considering a family of possible models and then selecting the one with maximum entropy or minimum of maximum of Kullback-Leibler divergence produce networks with a lower divergence on average to the original one than the usual Bayesian procedure. We have carried out a Friedman non-parametric test and in the three networks the differences are significant (p-values: 0.000006, 0.0159, 0.0023, for Cancer, Earthquake, and Survey networks, respectively). In a posthoc Friedman-Nemenyi test, the differences between MinMaxKL and

[2] Plotted with Python `seaborn` package.

Table 1. Means of divergences of estimated models and the original one

Network	Bayesian	MinKL	MaxKL	MEntroKL	MinMaxKL
Cancer	0.013026	0.011225	0.014126	0.012712	0.012270
Earthquake	0.017203	0.013132	0.019769	0.016784	0.016072
Survey	0.031459	0.028498	0.033899	0.031257	0.030932

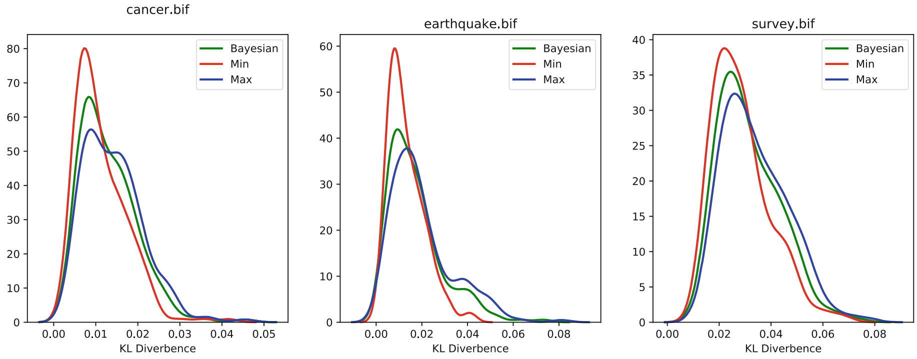

Fig. 1. Density for the Bayesian, minimum, and maximum Kullback-Leibler divergences.

Bayesian are significant in Cancer and Survey networks (p-values: 0.027, 0.017) but not in Cancer (p-value: 0.1187). The differences of MaxEntropy and the Bayesian procedure are not significant.

4.2 Experiment 2

In this case, we have a similar setting than in Experiment 1, but what we have measured is the number of networks that are obtained by our procedure (number of elements in $M_{\mathcal{D}}^{\alpha}$) and the distribution of the number of networks by each ESS $s \in S$. In Fig. 2 we can see the densities of the number of networks (left) and the figure with the averages of the networks by each $s \in S$. First, we can observe that the number of possible networks is low in average (below 5) for our selection of networks, α, and sample size, but that the right queue of the densities is somewhat large, existing cases in which the number of possible networks is 20 or more. With respect to the number of networks by value of ESS s, the most important fact is that the distribution of networks by ESS is highly dependent of the network, being the networks for Survey obtained with much higher values of s than in the case of Cancer or Earthquake. This result puts in doubt the usual practice of selecting a value of s when learning a Bayesian network without thinking that this does not have an effect in the final result.

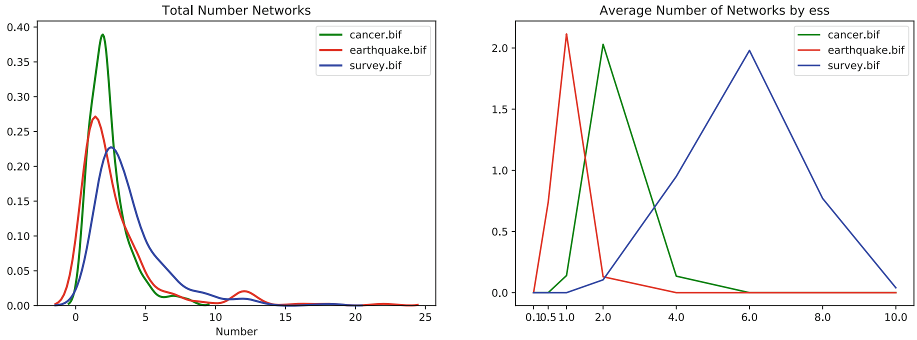

Fig. 2. Densities of the number of networks (left) and average number of networks by ESS (right).

4.3 Experiment 3

In this case, we compare the results of selecting a unique network by fixing a value of s (the optimal one for this value) with the result of selecting the parameter s and the network G optimizing the score. We again repeat a similar experiment to the other two cases, but we compute the networks: the Bayesian network, given by the pair (G, s) with highest score (the Bayesian approach in Experiment 1), and the best graph for each one of the values $s \in S$. For each one of the networks we compute the Kullback-Leibler divergence with the original one. The results of the averages of these divergences are depicted in Fig. 3 for each one of the networks. The dashed line represents the average of the divergence pair (G, s) with best score. On it, we can see that selecting the pair with best score is a good idea in Cancer network, as it produces an average divergence approximately equal to the best selection of value of s, but that is not the case of Earthquake and Survey networks, as there are many selections of s producing networks with lowest divergences than the divergence of the pair with best score. For this reason, is not always a good idea to select the equivalent sample size by using an empirical likelihood approach (the sample size giving rise to greatest likelihood). Other observation is that the shape of the densities of the divergences is quite different by network. For example, in Survey the lowest divergences are obtained with the highest values of s, while in Cancer a minimum is obtained for a low sample size of 2.

4.4 Experiment 4

In this case we have tested the evolution of the number of possible networks (elements in $M_{\mathcal{D}}^{\alpha}$) as a function of the sample size. For this aim, instead of fixing a sample size of 500, we have repeated the generation of a sample and the estimation of the models $M_{\mathcal{D}}^{\alpha}$ for different samples sizes ($n = 400, 500, 1000, 2000, 5000, 10000, 20000, 40000$) and for each value of n we have compute the number of models in $M_{\mathcal{D}}^{\alpha}$ (repeating it 200 times). Finally Fig. 4 shows the average number

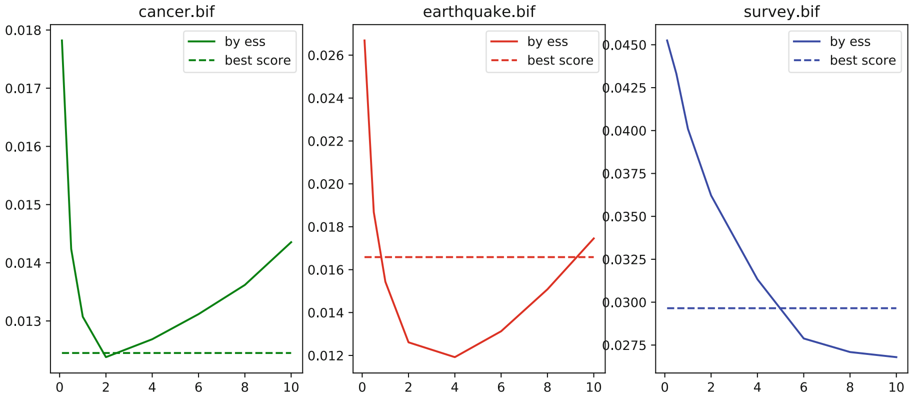

Fig. 3. Kulback Leibler of the best network and the best network by ESS s.

of models for each sample size. As it can be expected the number of models decreases when the sample size increases, very fast at the beginning and more slowly afterwards. In some cases, there are minor increasings in the average number of models when the sample size increases. We think that this is due to the fact that the density of the number of models has a long queue to the right, existing the possibility of obtaining some few cases with a high number of models. This fact can produce this small local irregularities.

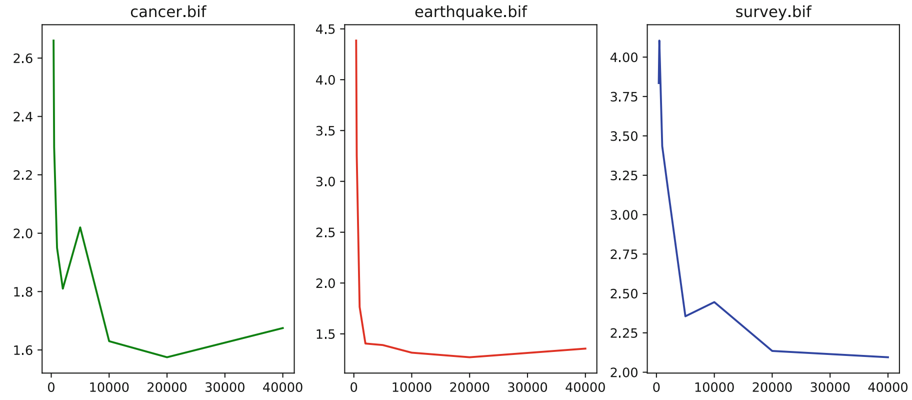

Fig. 4. Evolution of the number of possible networks as a function of the sample size.

5 Conclusions and Future Work

In this paper we have applied the general procedure proposed in Moral [15] to learn a generalized credal network (a set of possible Bayesian networks) given a dataset of observations. We have implemented algorithms for its computation and we have shown that the results applied to learning from samples simulated from small networks are promising. In particular, our main conclusion is that the usual procedure of selecting a network with the highest score does not make too much sense, when there is a set of networks that are equally plausible and that represents probability distributions with a similar divergence to the one associated with the true network. Even in this family, we can find networks using other alternative procedures with smaller divergences to the original one, as the case of considering the minimum of the maximum of Kullback-Leibler divergences in the family of possible models.

Our plans for future work are mainly related to making scalable the proposed procedures and algorithms. When the number of variables increases a direct application of the methods in this paper can be unfeasible. We could try to use more accurate bounds to prune A* search [3], but even so, the number of networks for a threshold could be too large to be computed. Experiments in this line are necessary. Then it would be convenient to develop approximations that could learn a set of significant networks from the full family of possible ones. Other line of research is to integrate several networks into a more compact representation: for example if a group of networks share the same structure with different probabilities try to represent it as a credal network with imprecision in the probabilities.

Other important task is to try to use the set of possible models to answer structural questions, as: is there a link from X_i to X_j? An obvious way to answer it is to see whether this link is present in all the networks of set of learned models, in none of the networks, or in some of them but not in all. In that case, the answer could be yes, no, or possibly. But a theoretical study justifying this or alternative decision rules would be necessary, as well as algorithms designed to answer these questions without an explicit construction of the full set of models.

References

1. Ankan, A., Panda, A.: pgmpy: Probabilistic graphical models using Python. In: Proceedings of the 14th Python in Science Conference (SCIPY 2015). Citeseer (2015). https://doi.org/10.25080/Majora-7b98e3ed-001
2. Cattaneo, M.E.G.V.: A continuous updating rule for imprecise probabilities. In: Laurent, A., Strauss, O., Bouchon-Meunier, B., Yager, R.R. (eds.) IPMU 2014. CCIS, vol. 444, pp. 426–435. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08852-5_44
3. Correia, A.H., Cussens, J., de Campos, C.P.: On pruning for score-based Bayesian network structure learning. arXiv preprint arXiv:1905.09943 (2019)
4. Correia, A.H.C., de Campos, C.P., van der Gaag, L.C.: An experimental study of prior dependence in Bayesian network structure learning. In: International Symposium on Imprecise Probabilities: Theories and Applications, pp. 78–81 (2019)
5. Couso, I., Moral, S.: Sets of desirable gambles: conditioning, representation, and precise probabilities. Int. J. Approximate Reasoning **52**(7), 1034–1055 (2011). https://doi.org/10.1016/j.ijar.2011.04.004

6. Cozman, F.: Credal networks. Artif. Intell. **120**, 199–233 (2000). https://doi.org/10.1016/S0004-3702(00)00029-1
7. Friedman, N., Koller, D.: Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. Mach. Learn. **50**, 95–125 (2003). https://doi.org/10.1023/A:1020249912095
8. Gärdenfors, P., Sahlin, N.E.: Unreliable probabilities, risk taking, and decision making. Synthese **53**(3), 361–386 (1982). https://doi.org/10.1007/BF00486156
9. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data. Mach. Learn. **20**(3), 197–243 (1995). https://doi.org/10.1023/A:1022623210503
10. Karan, S., Zola, J.: Exact structure learning of Bayesian networks by optimal path extension. In: 2016 IEEE International Conference on Big Data (Big Data), pp. 48–55. IEEE (2016). https://doi.org/10.1109/BigData.2016.7840588
11. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT press, Cambridge (2009)
12. Liao, Z.A., Sharma, C., Cussens, J., van Beek, P.: Finding all Bayesian network structures within a factor of optimal. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7892–7899 (2019). https://doi.org/10.1609/aaai.v33i01.33017892
13. Masegosa, A.R., Moral, S.: Imprecise probability models for learning multinomial distributions from data. Applications to learning credal networks. Int. J. Approximate Reasoning **55**(7), 1548–1569 (2014). https://doi.org/10.1016/j.ijar.2013.09.019
14. Moral, S.: An empirical comparison of score measures for independence. In: Proceedings of the 10th IPMU International Conference, pp. 1307–1314 (2004)
15. Moral, S.: Learning with imprecise probabilities as model selection and averaging. Int. J. Approximate Reasoning **109**, 111–124 (2019). https://doi.org/10.1016/j.ijar.2019.04.001
16. Neapolitan, R.: Learning Bayesian Networks. Prentice Hall, Upper Saddle River (2004)
17. Pearl, J.: Probabilistic Reasoning with Intelligent Systems. Morgan & Kaufman, San Mateo (1988). https://doi.org/10.1016/C2009-0-27609-4
18. Quaeghebeur, E.: Desirability. In: Introduction to Imprecise Probabilities, chap. 1, pp. 1–27. Wiley (2014). https://doi.org/10.1002/9781118763117.ch1
19. Scutari, M.: Bayesian network repository of bnlearn (2007). https://www.bnlearn.com/bnrepository/
20. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
21. Silander, T., Kontkanen, P., Myllymäki, P.: On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter. In: Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence, pp. 360–367. AUAI Press (2007)
22. Walley, P.: Towards a unified theory of imprecise probability. Int. J. Approximate Reasoning **24**, 125–148 (2000). https://doi.org/10.1016/S0888-613X(00)00031-1