Bayesian networks : a better than frequentist approach for parametrization, and a more accurate structural complexity measure than the number of parameters

Sylvain Gelly, Olivier Teytaud

Bayesian networks : a better than frequentist approach for parametrization, and a more accurate structural complexity measure than the number of parameters

Bayesian networks learning

Sylvain Gelly, Olivier Teytaud

TAO-inria, LRI, UMR 8623(CNRS - Universite Paris-Sud), bat 490 Universite Paris-Sud 91405 Orsay Cedex France, email: sylvain.gelly@lri.fr

RÉSUMÉ. *L'apprentissage à partir d'exemples est un problème classiquement étudié, au niveau théorique, via la théorie du processus empirique (fournissant des résultats asymptotiques) ou la théorie de l'apprentissage ([KOL 61, VAP 71]). L'application de ces théories aux réseaux bayésiens est incomplète et nous proposons une contributions, essentiellement via les nombres de couverture. Nous en déduisons de nombreux corollaires et notamment une approche meilleure que fréquentiste pour l'apprentissage de paramètres et un score prenant en compte une mesure d'entropie structurelle qui affine les classiques mesures basées sur le nombre de paramètres seulement. Nous proposons alors des méthodes algorithmiques pour traiter de l'apprentissage qui découle de nos propositions, basées sur BFGS et l'affinage adaptatif du calcul du gradient.*

ABSTRACT. *The problem of calibrating relations from examples is a classical problem in learning theory. This problem has in particular been studied in the theory of empirical process (providing asymptotic results), and through statistical learning theory ([KOL 61],[VAP 71]). The application of learning theory to bayesian networks is still uncomplete and we propose a contribution, especially through the use of covering numbers. We deduce multiple corollaries, among which a better-than-frequentist approach for parameters learning and a score taking into account a measure of structural entropy that has never been taken into account before. We then investigate the algorithmic aspects of our theoretical solution, and propose a new approach based on : i) (possibly derandomized) Monte-Carlo method for the evaluation of the loss function and of its gradient ; ii) BFGS non-linear optimization. Empirical results show the relevance of both the statistical results and the algorithmic solution.*

MOTS-CLÉS : *Apprentissage dans les réseaux bayésiens, score sur les structures, théorie de l'apprentissage.*

KEYWORDS: *Learning in bayesian networks, structural score, learning theory.*

1. Introduction

Bayesian networks are a well known and powerful tool for representing and reasoning on uncertainty. One can refer to [PEA 00],[NAI 04] for a general introduction to bayesian networks. Learning the structure and the parameters of bayesian networks can be done through either expert information or data. Here, we only address the problem of learning from data, i.e. learning a law of probability given a set of examples distributed according to this law. Although a lot of algorithms exist for learning in bayesian networks from data, several problems remain. Furthermore, the use of learning theory for bayesian network is still far from complete.

First, when looking for a bayesian model, one can have different goals e.g. i) evaluating qualitatively some probabilites ; ii) evaluating expectations (of gain or loss). In the first case, evaluating a risk is roughly the question : does a given event happen with probability 10^{-30} or 10^{-5} ? Then, the use of logarithms, leading to maximum likelihood, is justified. In the second case, if we look for the expectation of f (vector of possible values indexed by possible states), the approximation of the real probability vector P by a probability vector Q leads to an error bounded (thanks to Cauchy-Schwartz inequality) by $||P - Q|| \times ||f||$. Therefore, optimizing a criterion monotonous as a function of $||P - Q||^2$ is the natural approach.

Second, when the postulated structure is not the right one, maximum likelyhood (frequency approach for probability estimation) leads to very unstable results. We then propose a non-standard and tractable loss function for bayesian networks and evidences of the relevance of this loss function.

Futhermore, the purpose of this paper is to provide some theoretical insights into the problems of learning bayesian networks. The use of statistical learning theory provides bounds on the number of examples needed to approximate the distribution for a given precision/confidence, depending upon some complexity measures ; using covering numbers, we show the influence of structural entropy, as a refinement of scores based on the number of parameters only. We also provide, among other things, an algorithm which is guaranteed to converge to an optimal (in size) structure as the number of i.i.d examples goes to infinity.

We also make comparisons between the form of our bound to the form of the different scores classically used on bayesian network structure learning.

The paper is organized as follows : in section 2 we present an overview of our most concrete results. In section 3 we briefly survey some classical ways to learn bayesian networks from data and discuss the contribution of this paper in regard of existing results. In section 4 we introduce formally the problem and the notations. Section 5 first recalls some classical results of learning theory and presents our result about evaluation of VC-dimensions and covering numbers. We then generalize our results to more general bayesian networks, with hidden variables, in section 5.3. Section 6 shows usefull corollaries applied to structure learning, parameters learning, univer-

sal consistency, and others. Section 7 presents algorithmic details. Section 8 present empirical results.

2. Overview of results

The usual learning methods for parameters (section 3.1) lead asymptotically to the best parameters if the structure of the bayesian network is exact. However, we show that the classical frequentist method is not-optimal if the structure does not match the decomposition of the joint law. On the other hand, we prove universal consistency of global fitting during the minimization of the empirical error (section 6.2).

We obtain risk bounds. Therefore, given a number of example, and after learning, we can say that the probability to get an error larger than ϵ is bounded by δ. Equivalently, we can classically deduce the number of examples needed to have an error lower than ϵ with probability at least $1 - \delta$.

We address also the case with hidden variables (section 5.3). We apply these bounds either in the case of a finite number of variables (section 6.3) and infinite number of variables (section 6.4).

Section 6.5 and theorem 8 provides an algorithm that guarantees universal consistency and asymptotic convergence towards the "good" structure asymptotically. The "good" structure is given in the sense of the user-defined complexity of the structure. Hence, we prove that the algorithm gives us a not too complex structure.

Let's now compare the form of our bound to the form of existing scores. This comparison gives interesting insights on what is important to measure the complexity of a structure. The first lemmas helps calculating the covering number of the set of bayesian networks for a given structure. These covering numbers are directly related to the complexity of the structure. Theorem 7 states a bound that contains R and $H(r)$ where R is the number of parameters of the structure and where $H(r) = -\sum_{k=1}^{a}(r(k)/R)\ln(r(k)/R)$ with $r(k)$ the number of parameters for the node k. Hence, $H(r)$ is the entropy of the number of parameters calculated over the nodes.

We show then that the number of parameters of the bayesian network is not the only measure of the complexity. Hence, the AIC, BIC or MDL measure are quite different because they don't take into account this $H(r)$. See figure 1 for an illustration of the role of the entropy term.

We also show (difference between theorem 6 and theorem 7) that we have a tighter bound if we consider the number of parameters node by node, without trying to gather the nodes in a more smart way. This means, that more complex patterns on the structure of the bayesian network do not play a role, for our bound. Only the distribution of the number of parameters between the different nodes is important.

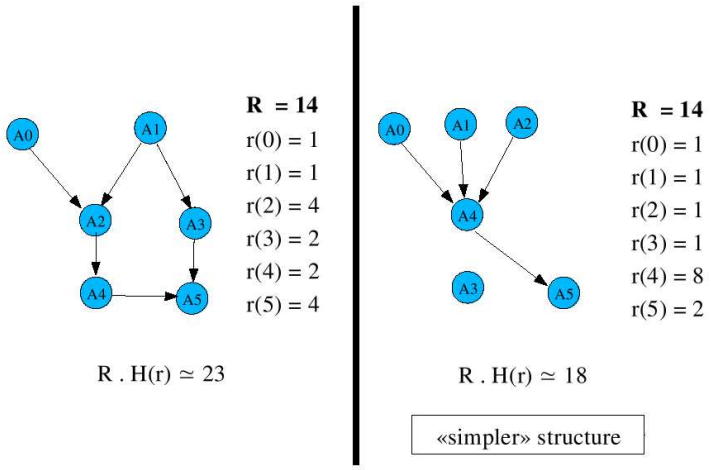

Figure 1. *Role of the entropy term in the score of a bayesian network structure. The two structures have the same number of parameters ($R = 14$), but have different distribution of the parameters over the structure. Hence, they have different entropy terms, and the right hand structure is considered "simpler" by our score.*

In section 7, we then present algorithms for optimizing our loss function. We use BFGS, and the complexity of the loss function implies some non-straightforward algorithms for evaluating both the value of the loss function and the the gradient of the loss function.

Empirical results (section 8) then shows both the statistical relevance of our approach and the algorithmic efficiency of our method.

3. Bayesian network learning

The problem of learning a bayesian network can be divided in two parts :

– Learning the structure of the network, which is related to a graph, and not to the values of the probabilities.

– Given a structure, learning the parameters of the bayesian network, i.e. the conditional probabilities among variables.

Learning the structure, is a much more challenging problem than estimating the parameters. Hence, the larger part of the works have addressed this issue.

3.1. *Learning parameters*

The classical approach for learning parameters is the likehood maximization. This leads, with the classical decomposition of the joint probability in a product, to estimate separately each term of the product with the data. This method asymptotically converges toward the true probability, if the proposed structure is exact.

The bayesian method rather tries to calculate the most probable parameters given the data, and this is equivalent, with the Bayes theorem, to weight the parameters with an *a priori* law. The most used *a priori* is the Dirichlet distribution (see for example [ROB 94]).

3.2. *Structure learning*

Structure learning can be divided in two different methods :

– Find causal relations (and independencies and conditional dependencies) between the random variables, and deduce the structure of the graph.

– Map every structure of bayesian network to a score and search into the space of all structures for a "good" bayesian network, i.e., a structure with a good score.

The space of all structures is super-exponential, so heuristics must be defined when using the second method (limiting to the tree structures, sorting the nodes, greedy search). The search could also be done on the space of Markov equivalent structures (the structures which encode the same probability law), which has better properties ([CHI 02]). Our work, among other results, provide a score to the structures of bayesian networks, and so is closer to the second category. This score includes the influence of the structural entropy of the network.

3.2.1. *Learning causality*

The principle of this method is the research of the independencies (conditionally or not) between the variables. We can cite the algorithms IC, IC*, [PEA 00], PC, [SPI 93], and more recently BN-PC of Cheng et al. [CHE 97a], [CHE 97b],[CHE 02].

The classical statistical tests used to test the independencies between variables is the χ^2 test. For hidden variables, the method is more complex, and we must distinguish several types of causality. We will not go further on this point here.

3.2.2. *Algorithms based on a score*

The notion of score of a structure is generally based on the Occam's razor principle. The score measures the "complexity" of the structure. Therefore, the algorithm choose a compromise between the empirical error made by the structure and the score of this structure. $Dim(bn)$ denotes the "dimension" of the bayesian network, which counts the number of parameters.

Here follows some well known scores for bayesian networks.

– AIC criteria [AKA 70] or BIC [SCH 78] use essentially $dim(bn)$ to penalize the complexity of the bayesian network.

– The Minimum Description Length (MDL) principle [RIS 78] uses the number of arcs and the number of bits used to code the parameters.

– The bayesian approach puts an *a priori* probability on the structure. For example, the bayesian Dirichlet score [COO 92] assumes a Dirichlet *a priori* on the parameters. Some variants exist, like BDe [HEC 94] , or BDgamma [BOR 02] which uses an hyperparameter, or methods using *a priori* probabilities on each child/parent relation (given for example by an expert).

4. Problem definition and notations

Let A_1, ...A_a be a binary random variables. We note $\mathcal{A} = \{A_1, \ldots, A_a\}$. For the sake of clarity, we restrict our attention to binary random variable, without loss of generality.

4.1. *Notations*

We note A_b, where b is a subset of $[1, a]$, the random variable product of A_i where $i \in b$. If $b = \emptyset$, then A_b is the event always true. A **bayesian network** is a family K_1,...,K_a of subsets of $[1, a]$ where $i \notin K_i$. We can and we will assume that $i < K_i$, i.e. that i is smaller than every element in K_i, without loss of generality. An **instanced bayesian network** ibn, associated with a bayesian network bn, is a law on $(A_1, \ldots, A_i)$ such that $ibn(A_1, \ldots, A_a) = \prod_j P(A_j|A_{K_j})$. With bn a bayesian network, and ibn an instance of bn, we will say by abuse that $ibn \in bn$. We will map ibn with a vector of size 2^a corresponding to all the probabilities of all events $(A_1 = v_1, \ldots, A_a = v_a)$. A **bayesian network** bn is said **well defined** if there exists an instance $ibn \in bn$. We call **parameter** of a bayesian network (BN), one of the real numbers $P(A_j|A_{K_j})$. We call **number of parameters** of a BN, and we note $p(bn) = \sum_j 2^{\#K_j}$, where $\#b$ is the cardinal of b.

We consider $\hat{P}$ an empirical law (i.e. a sample of Dirac masses located at examples). Let P be a target law of probability. The sample leading to $\hat{P}$ is assumed independent and identically distributed (i.i.d.). We note E and $\hat{E}$ the expected value operators associated to P and $\hat{P}$ respectively. We note χ the random variable

$$\chi = (0, 0, 0, 0, \ldots, 0, 1, 0, \ldots, 0, 0, 0) \in \{0, 1\}^{2^a}$$

(all zeros except one 1 on the ith position with probability the probability of the i^{th} set of possible values of $A_1 \ldots A_a$).

For Q a vector of size 2^a, of sum 1, identified to a probability distribution on the random vector $(A_1, \ldots, A_a)$ (more precisely $Q(i)$ is the probability of $(A_1 =$

$a_1, ..., A_a = a_a)$, with $(a_1, ...a_a)$ the ith tuple of size a, among the 2^a tuples possible), we define

$$L(Q) = E(\sum_{i \in [1,2^a]} |Q(i) - \chi(i)|^2)$$

where $\sum$ is the sum operator on vector, and $\hat{L}(Q) = \hat{E}(\sum_{i \in [1,2^a]} |Q(i) - \chi(i)|^2)$. If bn is a well defined BN, we note $L(bn) = \inf_{ibn \in bn} L(ibn)$.

4.2. *Preliminary lemmas and propositions*

To spot the interest of $L(.)$ and $\hat{L}(.)$, we can note the

Lemma 0 :

With $N(Q) = \sum_{i \in [1,2^a]} (P(i) - Q(i))^2$. and $\hat{N}(Q) = \sum_{i \in [1,2^a]} (\hat{P}(i) - Q(i))^2$, we claim :

$$L(Q) = N(Q) + 1 - \sum_{i \in [1,2^a]} P_i^2$$

$$\hat{L}(Q) = \hat{N}(Q) + 1 - \sum_{i \in [1,2^a]} \hat{P}_i^2$$

Moreover, we claim the

Proposition A :

With probability $1 - \delta$, with $x^* \in argminL(.) = argminN(.)$, for all $\hat{x} \in argmin\hat{L} = argmin\hat{N}$, with $\sup_\delta X$ the $1 - \delta$ quantile of X :

$$L(\hat{x}) \leq L(x^*) + 2 \sup_\delta |L - \hat{L}|$$

Proof :

$$\begin{aligned} L(\hat{x}) &\leq \hat{L}(\hat{x}) + \sup_\delta |L - \hat{L}| \\ \hat{L}(\hat{x}) &\leq \hat{L}(x^*) \\ \hat{L}(x^*) &\leq L(x^*) + \sup_\delta |L - \hat{L}| \end{aligned}$$

Summing this three inequalities, we get the expected result. ■

And finally :

Proposition B :

With probability $1 - \delta$, with $x^* \in argminL(.) = argminN(.)$, For all $\hat{x} \in argmin\hat{L} = argmin\hat{N}$, with $\sup_\delta X$ the $1 - \delta$ quantile of X :

$$N(\hat{x}) \leq N(x^*) + 2 \sup_{\delta} |L - \hat{L}|$$

Proof : Consequence of lemma 0 and proposition A. ■

All these elements confirm the interest of $\hat{L}$, which has both the interest of being an empirical average and the advantage of being closely related to natural cost functions.

5. Learning theory results

The VC dimension ([VAP 71]) is the more classical tool of learning theory. It quantifies the inaccuracy of a learning depending on the size of the search space. This type of calculus has already been done in [WOC 02]. We show similarly results in section 5.1. The use of covering numbers, already known on the time of [KOL 61], allows more precise bounds, as shown in section 5.2.

We will note $F(H, \delta)$ the smallest real Δ such that $P(\sup_{h \in H} |\hat{L}(h) - L(h)| \geq \Delta/\sqrt{n}) \leq \delta$, with n the number of examples. $F(H, \delta)$ depends upon n, but in many cases the dependency upon n can be removed (i.e. the supremum on n is not a bad approximation) and so we often refer to $F(H, \delta)$.

5.1. *Bounds based on VC dimension*

For a bayesian network bn, with probability at least $1 - \delta$:

$$\sup_{ibn \in bn} |\hat{L}(ibn) - L(ibn)| \leq F(\{ibn\}, \delta)/\sqrt{n}$$

The application $(a_1, \ldots, a_a) \mapsto \log\ P(A = 1 = a_1, \ldots, A_a = a_a)$ is linear in the log of the parameters of the bayesian network. Hence, the VC-dimension of b_n is upper-bounded by the number of parameters. Combining with increasing functions preserves the VC dimension, and so the VC dimension of bn, seen as application mapping $A_{[1,a]}$ to a probability is upper bounded by the number of parameters. We then deduce the

Theorem C :

The VC dimension of the set bn of instanced bayesian networks is upper bounded by the number of parameters V of bn. So thanks to classical results of learning theory

$$P(\exists ibn \in bn | \hat{L}(ibn) - L(ibn)| \geq \epsilon) < 8(32e/\epsilon) \log(128e/\epsilon))^V \exp(-n\epsilon^2/32)$$

if $n \geq V$, and the covering number of ibn for the metric $d(ibn_1, ibn_2) = E(|ibn_1(A_{[1,a]}) - ibn_2(A_{[1,a]})|)$ is upper bounded by $e(R+1)(4e/\epsilon)^R$.

Proof : These results are classical in learning theory. See e.g. [ANT 99, Th18.4 and 17.4] for the upper bound on the probability and [ANT 99, Th18.4, p251] for the covering number. We note that our results, even if they use a norm $N1(.)$, defined in the sequel, are better.

5.2. *Bound based on the covering number*

The covering numbers are a classical tool of learning theory. Inequalities of large deviations coming from this tool are usually tighter than those coming from VC-dimension.

5.2.1. *Introduction*

If one can cover $\mathcal{F}$ with $N1(\mathcal{F},\epsilon)$ ϵ balls for the distance $d(x,y)=\sum|x_i-y_i|$, if $\hat{L}$ and L are between 0 and 2, then :

1) the risk, for a given function, to have a deviation $|\hat{L}-L|$ more than 2ϵ, is bounded by $2\exp(-2n\epsilon^2)$;

2) The risk to have at least one of the centers of the balls having a deviation more than 2ϵ is upper bounded by $2N1(\mathcal{F},\epsilon)\exp(-2n\epsilon^2)$;

3) If $d(f,g)\leq\epsilon\Rightarrow|L(f)-L(g)|\leq\epsilon$ and $d(f,g)\leq\epsilon\Rightarrow|\hat{L}(f)-\hat{L}(g)|\leq\epsilon$, (which is the case here, see lemma 2), then the risk to have at least a function in $\mathcal{F}$ having a deviation more than 4ϵ is upper bounded by $2N1(\mathcal{F},\epsilon)\exp(-2n\epsilon^2)$. Indeed, if for all g of ϵ-skeleton C, we have $|\hat{L}(g)-L(g)|\leq2\epsilon$, so we can map every f to one $g\in C$ such that $d(f,g)<\epsilon$ and so

$$|\hat{L}(f)-L(f)|\leq|\hat{L}(f)-\hat{L}(g)|+|\hat{L}(g)-L(g)|+|L(g)-L(f)|\leq\epsilon+2\epsilon+\epsilon\leq4\epsilon$$

The risk to have, among $\mathcal{F}$, a deviation more than ϵ is then upper bounded by $\delta=2N1(\mathcal{F},\epsilon/4)\exp(-2n(\epsilon/4)^2)$.

Then we can write :

Proposition (maximal deviation for a given covering number) :

$$\sqrt{n}F(\mathcal{F},\delta)\leq\inf\{\epsilon|\log(2N1(\mathcal{F},\epsilon/4))-n\epsilon^2/8\leq\log\delta\}$$

A lot of variations of this type of result exists in the literature. One can for example see [VID 97] and [ANT 99].

The covering number $N_\infty(\mathcal{F},\epsilon)$ of $\mathcal{F}=[0,1]^{2^a}$ is upper bounded by $\lceil1/2\epsilon\rceil^{2^a}$ for the distance $d(x,y)=\sup_i|x_i-y_i|$.

The covering number $N1(\mathcal{F},\epsilon)$ of $\mathcal{F}=\{ibn\in bn\}$ is upper bounded as explained in the following subsection for the distance $d(x,y)=\sum|x_i-y_i|$. These both covering numbers deal with multi-valued functions ; this is different from usual covering numbers.

5.2.2. *Covering number of $\mathcal{F}$*

We assume, without loss of generality that the nodes of the bayesian network are topologically sorted ($i<K_i$ for i node of the BN)

Let E_k a partition of the node set such as :

– If $k \leq k'$ then $\forall (i,j) \in E_k \times E_{k'}, i \leq j$

– There is no edge between two nodes of a same E_k.

We call depth the number k corresponding to the partition E_k and l_k the number of the last element (node) of E_k. By convention, $E_0 = \emptyset$ and $l_0 = 0$.

Lemma 1 :

$$N1(F_k, 2^{nbe(k)}\epsilon' + \epsilon) \leq N(F_{k-1}, \epsilon) N_{\inf}(T_k, \epsilon')$$

where

– F_k indicates the set of the functions calculated by the bayesian network until the level k (i.e. using only the nodes of $\bigcup_{i=1}^{k} E_i$).

– $N_{\inf}$ indicates the covering number for the sup norm.

– T_k indicates the set of the vectors of the probabilities involved in the transition from the level $k-1$ to the level k (it is $[0,1]^{2^{l_k}}$).

– $nbe(k)$ indicates the number of the nodes of the bayesian network in the level k, so $\#E_k$;

– $l_k = \sum_{i=1}^{k} nbe(i)$;

Lemma 2 : $|L(Q) - L(Q')| \leq \sum_i |Q_i - Q'_i|$.

One can derive the **lemma 3 :** $N_\infty([0,1]^h, \epsilon) \leq \lceil \frac{1}{2\epsilon} \rceil^h$.

Lemma 4 : $N_\infty(T_k, \epsilon) \leq \lceil \frac{nbe(k)}{2\epsilon} \rceil^{r(k)}$.

where T_k indicates the set of the vectors of the conditional probabilities involved in the transition from the level $k-1$ to the level k and where $r(k)$ indicates the number of parameters of the network involved in the transition between level $k-1$ and k.

Precisely, for a fixed k, T_k is the set of $P(E_k | \bigcup_{i=1}^{k-1} E_i)$, the $E_i, i = 1, ..., k$ taking the 2^{l_k} possible values. $r(k)$ indicates the number of the $P(A_i|K_i)$ with $A_i \in E_k$, i.e. the number of parameters for this level.

Lemma 5 : Let K be the number of levels ; then

$$lN1(K) \leq \sum_{i=1}^{K} r(i) \ln(\lceil \frac{nbe(i) 2^{nbe(i)-1}}{\Delta_i} \rceil)$$

where $\epsilon_i > 0, i = 1...K$, $\epsilon_i < \epsilon_K, i = 1...K-1$, $\epsilon_0 = 0$, $\Delta(i) = \epsilon_i - \epsilon_{i-1}$ and $lN1(i) = log(N1(F_i, \epsilon_i))$ and with the notation $lN1(0) = 0$.

Theorem 6 :

$$lN1(\epsilon) \leq \sum_{k=1}^{K} r(k) \ln(nbe(k) 2^{nbe(k)-1} + \epsilon) - \sum_{k=1}^{K} r(k) \ln(\epsilon r(k)/R)$$

with $R = \sum_{i=1}^{K} r(i)$, $\epsilon = \epsilon_K$ et $lN1(\epsilon) = lN1(F_K, \epsilon)$, in particular for K the number of the last level.

Theorem 7 : The partition $\{E_k\}$ minimizing the bound of theorem 6 is the one in which all the E_k contain only one node. We have then :

$$\begin{aligned} lN1(\epsilon) &\leq \sum_{k=1}^{a} r(k)\ln(1+\epsilon) - \sum_{k=1}^{a} r(k)\ln(\epsilon r(k)/R) \\ &\leq R\ln((1+\epsilon)/\epsilon) + RH(r) \end{aligned}$$

where $H(r) = -\sum_{k=1}^{a}(r(k)/R)\ln(r(k)/R)$.

Remark 1 : As $R \leq 2^a$, we get a better bound on the covering number than the one we get from the VC-dimension which is $R(2^a/\epsilon)^R$.

Remark 2 : For a fixed R (total number of parameters), our inequality has a term in $log((1/\epsilon)^R)$ and a term which is the entropy of the vector $(r(1), \ldots, r(a))$, which shows that the less the parameters are equally distributed, the more the covering number is well controlled.

Proof of lemma 1 :

Let $k \geq 1$ fixed. Let $Pa(E_k)$ be the set of parent nodes of E_k. Let X be the set of the vectors of size $2^{\sum_{i=1}^{k-1} \#E_i} = 2^{l_{k-1}}$ representing the probabilities (hence of sum 1) of all bayesian networks of a given structure (all $ibn \in bn$) until the level $k-1$. More precisely $X = \{x = P(A_1, ..., A_{l_{k-1}})\}$, the l_{k-1}-tuple of A_i taking all the $2^{l_{k-1}}$ possible values. Let Y be the set of vectors of size $2^{\sum_{i=1}^{k} \#E_i} = 2^{l_k}$ representing the probabilities of $ibn \in bn$ until the level k. More precisely, $Y = \{y = P(A_1, ..., A_{l_k})\}$, the l_k-tuples of A_i taking all the 2^{l_k} possible values.

Let's cluster the vectors of the set X by classes $\tilde{X}_i$ such as for all $x \in \tilde{X}_i$ the values of the parents $Pa(E_k)$ are identical. Let N be the number of such classes. Let $t_i^j, i \in [1, N], j \in [1, 2^{neb(k)}]$ the probability of the jth value of the new variables (of level k) knowing a value of the class $\tilde{X}_i$ (each value of the variables in $\tilde{X}_i$ is appropriate because, by definition of $Pa(.)$, the new variables depend only on $Pa(E_k)$ among $E_1, ..., E_k$).

Let $y, y' \in Y$. We can then claim $y = (y_1, y_2, ..., y_N)$ with $y_i = (t_i^1\tilde{X}_i, t_i^2\tilde{X}_i, ..., t_i^{2^{neb(k)}}\tilde{X}_i)$ and $y' = (y'_1, y'_2, ..., y'_N)$ with $y'_i = (t'^1_i\tilde{X}'_i, t'^2_i\tilde{X}'_i, ..., t'^{2^{neb(k)}}_i\tilde{X}'_i)$.

Let $\epsilon' = sup_{i,j}|t_i^j - t'^j_i|$ and $\epsilon = sup_i\|\tilde{X}_i - \tilde{X}'_i\|_1$. Then :

$$\|y - y'\| = \sum_{i=1}^{N}\sum_{j=1}^{2^{neb(k)}} \|(t_i^j - t'^j_i)\tilde{X}_i + t'^j_i(\tilde{X}_i - \tilde{X}'_i)\|_1$$

$$\|y - y'\| \leq \sum_{i=1}^{N}\sum_{j=1}^{2^{nbe(k)}} \epsilon'\|\tilde{X}_i\|_1 + t'^j_i\|\tilde{X}_i - \tilde{X}'_i\|_1$$

$$= \sum_{i=1}^{N} 2^{neb(k)}\epsilon' \|\tilde{X}_i\|_1 + \|\tilde{X}_i - \tilde{X}'_i\|_1 \leq 2^{nbe(k)}\epsilon' + \epsilon$$

Therefore,

$$N1(F_k, 2^{nbe(k)}\epsilon' + \epsilon) \leq N(F_{k-1}, \epsilon) N_{\inf}(T_k, \epsilon')$$

■

Proof of lemma 2 :

$$|L(Q) - L(Q')| = |E\sum_i (Q_i - \chi_i)^2 - \sum_i (Q'_i - \chi_i)^2| \leq E|\sum_i (Q_i - \chi_i)^2 - \sum_i (Q'_i - \chi_i)^2|$$

$$\leq E|\sum_i |(Q_i - \chi_i) - (Q'_i - \chi_i)|| \leq E|\sum_i |Q_i - Q'_i|| \leq \sum_i |Q_i - Q'_i|$$

■

Proof of lemma 4 : Let k fixed. Let T the set of $P(E_k | \bigcup_{i=1}^{k-1} E_i)$.

Then $P(E_k | \bigcup_{i=1}^{k-1} E_i) = \prod_{A_i \in E_k} P(A_i | K_i)$.
The $P(A_i|K_i)$ are probabilities and therefore lie between 0 and 1.
Let R_k the set of the indexes of the parameters for the level k.
We consider a fixed ϵ-skeleton S of $[0,1]^{R_k}$.
Consider $(p_i)_{i \in R_k}$ a set of parameters for the $P(A_i|K_i)$ at level k.
t_j, the j-th coefficient of the level k, is equal to $t_j = \prod_{i \in h_j} p_i$ where h_j is the list of the indexes of the parameters involved in the calculus of the coefficient a_j ; we note that h_j is the cardinal of $nbe(k)$.
Define $(t_j) = \prod((p_i)_{i \in h_j})$; t is the vector of the coefficients, p is the vector of the parameters. We want to prove that $\prod(S)$ is a $(nbe(k) \times \epsilon)$-skeleton of $\prod([0,1]^{R_k})$.
In order to prove this, we note that by induction on $nbe(k)$ that $\| \prod(p) - \prod(p') \|_\infty \leq nbe(k)\|p - p'\|_\infty$ where $(p, p') \in ([0,1]^{R_k})^2$.

Finally, using lemma 3 : $N_\infty(R_k, \epsilon) \leq \lceil \frac{1}{2\epsilon} \rceil^{r(k)}$. Hence : $N_\infty(T_k, \epsilon) \leq \lceil \frac{nbe(k)}{2\epsilon} \rceil^{r(k)}$ ■

Proof of lemma 5 :

From lemma 4,

$$N_\infty(T_k, \epsilon) \leq \lceil \frac{nbe(k)}{2\epsilon} \rceil^{r(k)}$$

Let K be the number of levels.

From lemma 1, $\forall \epsilon, \epsilon' > 0, \forall 1 \geq k \leq K$:

$$N1(F_k, 2^{nbe(k)}\epsilon' + \epsilon) \leq N(F_{k-1}, \epsilon) N_{\inf}(T_k, \epsilon')$$

Therefore, with the change of variable $\epsilon = 2^{nbe(k)}\epsilon' + \epsilon$:

$$N1(F_k, \epsilon) \leq N(F_{k-1}, \epsilon - \epsilon')N_{\inf}(T_k, \frac{\epsilon'}{2^{nbe(k)}})$$

Hence, with the variable change : $\epsilon' = \epsilon - \epsilon'$:

$$N1(F_k, \epsilon) \leq N(F_{k-1}, \epsilon')N_{\inf}(T_k, \frac{\epsilon - \epsilon'}{2^{nbe(k)}})$$

for all $\epsilon \geq 0$, with $\epsilon = \epsilon_K$,

$$lN1(\epsilon) \leq \sum_{k=1}^{K} r(k) \ln(N_\infty(T_k, \epsilon_k - \epsilon_{k-1}))$$

$$lN1(\epsilon) \leq \sum_{k=1}^{K} r(k) \ln(\frac{nbe(k)2^{nbe(k)-1}}{\epsilon_k - \epsilon_{k-1}})$$

$$lN1(\epsilon) \leq \sum_{k=1}^{K} r(k) \ln(\frac{nbe(k)2^{nbe(k)-1}}{\Delta_k})$$

■

Proof of theorem 6 :

Bounding the integer part, we can transform the lemma 5 as follows :

$$lN1(\epsilon) \leq \sum_{k=1}^{K} r(k) \ln(nbe(k)2^{nbe(k)-1} + \Delta_k) - \sum_{k=1}^{K} r_k \ln(\Delta_k)$$

and bounding Δ_k by ϵ,

$$lN1(\epsilon) \leq \sum_{k=1}^{K} r(k) \ln(nbe(k)2^{nbe(k)-1} + \epsilon) - \sum_{k=1}^{K} r_k \ln(\Delta_k)$$

In particular, with $\Delta_k = \frac{r(k)}{\sum_k r(k)}$ (which comes from the Kuhn-Tucker condition for the maximization $\sum r_k \ln(\Delta_k)$ under the constrain $\sum Delta_k = \epsilon$), we get

$$lN1(\epsilon) \leq \sum_{k=1}^{K} r(k) \ln(nbe(k)2^{nbe(k)-1} + \epsilon) - \sum_{k=1}^{K} r_k \ln(\epsilon r(k)/R)$$

■

Proof of theorem 7 :

The theorem 6 is true for any splitting of the network in levels, provided that a node of a level k does not depend upon another node of the same level. ($E_k \bigcap Pa(E_j) = \emptyset$).

We can now optimize the result by changing the limits of the levels.

Let $k(i)$ be the level in which the node i belongs. Let $s(i)$ the number of parameters associated to the node i, i.e. $2^{\#K_i}$. We have then $\forall i \in [1,a], r_{k(i)} = \sum_{j=l_{k(i)}}^{l_{k(i)+1}} s(j)$.

Then using Theorem 6 :

$$lN1(\epsilon) \leq \sum_{k=1}^{K} r(k) \ln(\frac{R(nbe(k)2^{nbe(k)-1}+\epsilon)}{\epsilon r(k)})$$

hence :

$$lN1(\epsilon) \leq \sum_{k=1}^{K} (\sum_{j=l_k}^{l_{k+1}} s(j)) \ln(\frac{R(nbe(k)2^{nbe(k)-1}+\epsilon)}{\epsilon \sum_{j=l_k}^{l_{k+1}} s(j)})$$

so :

$$lN1(\epsilon) \leq \sum_{i=1}^{a} s(i) \ln(\frac{R((l_{k(i)+1}-l_{k(i)})2^{l_{k(i)+1}-l_{k(i)}-1}+\epsilon)}{\epsilon \sum_{j=l_{k(i)}}^{l_{k(i)+1}} s(j)})$$

$$lN1(\epsilon) \leq \sum_{i=1}^{a} s(i) \ln(\frac{R(c(k(i))+\epsilon)}{\epsilon \sum_{j=l_{k(i)}}^{l_{k(i)+1}} s(j)})$$

with $c(k) = (l_{k+1}-l_k)2^{l_{k+1}-l_k-1}$

Let's assume that there exists a E_k of cardinal > 1. To simplify the notations, we can assume, without loss of generality that $k = 1$ and $s(l_1) = Min_{i \in E_1} s(i)$. Let $l = l_1$. Let $C(n) = n2^n$.

We are going to prove that the bound is better for the partition such that we remove l_1 from the level k, therefore adding a level only composed of node l_1. We can remark that this new partition respect the constraints if these were respected for the first partition.

The terms of the bound which are modified for the first and second partition are respectively :

$$\sum_{i=1}^{l-1} s(i) \ln(\frac{C(l)}{s(l)+\sum_{j=1}^{l-1} s(j)}) + s(l) \ln(\frac{C(l)}{s(l)+\sum_{j=1}^{l-1} s(j)})$$

and

$$\sum_{i=1}^{l-1} s(i) \ln(\frac{C(l-1)}{\sum_{j=1}^{l-1} s(j)}) + s(l) \ln(\frac{1}{s(l)})$$

The difference between the two bounds can be written :

$$d = \sum_{i=1}^{l-1} s(i) \ln(\frac{C(l)\sum_{j=1}^{l-1} s(j)}{C(l-1)(s(l)+\sum_{j=1}^{l-1} s(j)}) + s(l) \ln(\frac{C(l)s(l)}{s(l)+\sum_{j=1}^{l-1} s(j)})$$

So :

$$d = A\ln(\frac{C(l)A}{C(l-1)(A+B)}) + B\ln(\frac{C(l)B}{A+B})$$

with $A = \sum_{j=1}^{l-1} s(j)$ et $B = s(l)$. Therefore :

$$\frac{d}{A+B} = \lambda\ln(\lambda C(l)/C(l-1)) + (1-\lambda)ln(C(l)(1-\lambda))$$

with $\lambda = \frac{A}{A+B}$.

The minimum of this expression is for $\lambda = \frac{C(l-1)}{1+C(l-1)}$ and so $0 \leq \ln(\frac{C(l)}{1+C(l-1)}) \leq \frac{d}{A+B}$ ■

5.2.3. *Summary of the results*

We have calculated an upper bound on the covering number of the family of instanced bayesian networks $ibn \in bn$ for a given structure bn. This structure determines the number of parameters $r(k)$ for $k \in [1, K]$ (and $R = \sum_{k=1}^{a} r(k)$).

Then, theorem 7 states that for all $\epsilon > 0$:

$$lN1(\epsilon) \leq \sum_{k=1}^{a} r(k)\ln(1+\epsilon) - \sum_{k=1}^{a} r(k)\ln(\epsilon r(k)/R) \tag{1}$$

The lemma 2 states that the conditions $d(f,g) \leq \epsilon \Rightarrow |L(f) - L(g)| \leq \epsilon$ and $d(f,g) \leq \epsilon \Rightarrow |\hat{L}(f) - \hat{L}(g)| \leq \epsilon$ are true. So we can here apply the results stated in the subsection 5.2.1, and then the risk to have, among $\mathcal{F}$, a deviation more than ϵ is then upper bounded by $\delta = 2N1(\mathcal{F}, \epsilon/4)\exp(-2n(\epsilon/4)^2)$. Therefore, $F(\mathcal{F}, \delta) \leq \sqrt{n}\inf\{\epsilon | \log(2N1(\mathcal{F}, \epsilon/4)) - n\epsilon^2/8 \leq \log\delta\}$. We can then rewrite this as

$$P(\sup_{ibn \in bn} |\hat{L}(ibn) - L(bn)| > \epsilon) \leq 2N1(bn, \epsilon/4)\exp(-n\epsilon^2/8) \tag{2}$$

with $\tilde{bn} = \{i\tilde{bn}; ibn \in bn\}$, and $F(\tilde{bn}, \delta) \leq F(bn, \delta)$.

And the equation above (2), using equation (1), can be solved in ϵ (depending upon R, $H = H(r)$, n and δ, as follows :

$$C = 2(H - \frac{1}{R}\log(\delta/2)) \qquad B = \frac{4n}{R}\exp(C)$$

$$A = -R \times W_{Lambert}(B) - 2\log(\delta/2) + 2RH \qquad \epsilon = \frac{4}{\exp(-A/(2R))}$$

$$\epsilon = 4\left(\frac{\delta}{2}\right)^{-\frac{1}{R}} \exp\left(-\frac{1}{2}W_{Lambert}\left(\frac{4n}{R}\frac{e^{2H}}{\frac{\delta}{2}^{\frac{2}{R}}}\right) - \frac{1}{R} + H\right)$$

Where $W_{Lambert}$ is the function such as $W_{Lambert}(x) \times e^{W_{Lambert}(x)} = x$.

Therefore, structural risk minimization leads to the optimization of $\hat{L} + \epsilon(R, H(r), n, \delta)$ where n is the number of examples, δ is a risk threshold, and R and $H(r)$ only depend on the structure.

This provides a score for structural learning.

5.3. *Results with hidden variables*

We here consider the case in which variables are hidden, so only a part of all the variables are involved in the calculus of $\hat{L}$ or L. It is important to remark that it is not equivalent to reduce the bayesian network to a smaller bayesian network. For example, a network with a hidden variable B and observed variables A_i for $i \in [1, d]$, with dependencies $P(A_i|B)$, has only $2d+1$ parameters and is difficult to model (i.e. would need much more parameters) with a bayesian network which has only the A_i as variables.

By mapping a bayesian network to a vector (of sum 1) of the probabilities it calculates, a bayesian network in which some variables are hidden can be mapped to a reduced vector (the vector of marginalized probabilities). If all the variables are binary (which is the case in this paper), the number of probabilities to code is divided by 2 for each variable which becomes hidden. An instance of a bayesian network ($\tilde{ibn}$) which has v variables, and among them l hidden variables, can be identified to an element of $[0,1]^{2^{v-l}}$ summing to 1, whereas the bayesian network ibn corresponding which does not have hidden variables, gives 2^v probabilities (hence a vector in $[0,1]^{2^v}$, summing to 1). $\tilde{ibn}$ then equals $summation(ibn)$, where $summation(.)$ is $(x_1, \ldots, x_n) \mapsto \sum_{i=1}^{n} x_i$.

As $summation(.)$ is 1-lipschitz for the distance $d(x,y) = \sum |x_i - y_i|$ (i.e. $d(\tilde{x}, \tilde{y}) \leq d(x,y)$), we deduce :

Proposition maximal deviation in a bayesian network with hidden variables :

The risk to have a deviation at least ϵ for a $\tilde{ibn} \in \tilde{bn}$ is upper bounded as follows :

$$P(\sup_{ibn \in bn} |\hat{L}(ibn) - L(bn)| > \epsilon) \leq 2N1(bn, \epsilon/4)\exp(-n\epsilon^2/8)$$

with $\tilde{bn} = \{\tilde{ibn}/ibn \in bn\}$, and $F(\tilde{bn}, \delta) \leq F(bn, \delta)$.

Remarks : We can notice that we don't improve the bound in spite of the fact that the number of parameters is slower. We can of course bound $F(\tilde{bn}, \delta)$ by

$F([0,1]^{v-l},\delta)$ if the number of hidden variables is so large that this rough bound becomes the best.

6. Paradigms of learning

Many applications of the calculus above can be defined, in the same spirit of use of covering numbers, to give :

– non-parametric non-asymptotic confidence intervals ;

– universally consistent algorithms.

We state in the sections below some of the numerous corollaries one can deduce from the calculus of covering numbers above. These corollaries also hold with hidden variables.

6.1. *Choose between several structures of bayesian network*

Let's assume that someone have to choose between several structures $bn_1,\ldots,bn_h$. Consider the algorithm that chooses bn_{i_0} such as $\inf_{ibn\in bn_{i_0}} \hat{L}(ibn) + F(bn_{i_0},\delta)/\sqrt{n}$ is minimal and chooses $i\hat{bn} \in bn_{i_0}$ such as $i\hat{bn} = argmin_{ibn\in bn_{i_0}} \hat{L}(ibn)$. So, the algorithm chooses the *structure* minimizing the empirical error penalized by a term depending upon the complexity of the structure. Then, it chooses the *bayesian network* of this structure minimizing the empirical error.

Corollary C1 : Then, $L(i\hat{bn}) \leq L(ibn') + \epsilon$ for all $ibn' \in \cup bn_i$, with $\epsilon = 3\sup F(bn_i,\delta)/\sqrt{n}$, with a risk upper bounded by $h\delta$.

Proof :

Define $ibn* = argmin_{ibn\in\cup_i bn_i} L(ibn)$. Define $\hat{i}bn_i = argmin_{ibn\in bn_i} \hat{L}(ibn)$.

Then, $P(L(\hat{i}bn_i) - \hat{L}(\hat{i}bn_i) > \epsilon/3) \leq \delta$ with $\epsilon = 3\sup F(bn_i,\delta)/\sqrt{n}$.

So, simultaneously for all i, $L(\hat{i}bn_i) - \hat{L}(\hat{i}bn_i) \leq \epsilon/3$, with probability $1-h\delta$.

And therefore, $L(i\hat{bn}) \leq \hat{L}(i\hat{bn}) + \epsilon/3$ and by definition of $i\hat{bn}$ (note that $\hat{L}(i\hat{bn}) + F(bn_{i_0})/\sqrt{n} \leq \hat{L}(ibn*) + F(bn*,\delta)/\sqrt{n}$ where $ibn* \in bn*$), $L(i\hat{bn}) \leq \hat{L}(ibn^*) + 2\epsilon/3$ and therefore $L(i\hat{bn}) \leq L(ibn^*) + \epsilon$ ■

(the constant 3 in ϵ is not optimal)

This provides a natural criteria to choose between several structures, in the spirit of the method of *"structural risk minimization"*, which is classical in learning theory.

6.2. *Comparison between local and global fitting : consistency of the minimization of* $\hat{L}$

Corollary C2 : Consider bn a bayesian network. Then for any distribution P,

$$L(argmin_{ibn \in bn} \hat{L}) \rightarrow \inf_{bn} L$$

whereas for some distributions P,

$$L(ibn \,/ \forall P(A_i|A_{K_i}) \in bn,\, ibn(A_i, A_{K_i})/ibn(A_{K_i}) = \hat{P}(A_i\, A_{K_i})/\hat{P}(A_{K_i})) \not\rightarrow inf_{bn} L$$

(i.e., calibrating each coefficient of bn on $\hat{P}$ leads asymptotically to a non-optimal ibn), with $ibn(B)$ for B a set of variable, is the probability given by the bayesian network ibn for the variables B.

Proof : The convergence $L(argmin_{ibn \in bn} \hat{L}) \rightarrow \inf_{bn} L$ is an immediate consequence of the finiteness of the covering number for any ϵ ; the VC-dimension being finite, the convergence is indeed almost sure. One can note that the same result holds with well-chosen nested families bn_n, increasing with n, as explained in section 6.4.

The counter-example for the second result is :

Let P be the law defined such as : $P(A = true \wedge B = true) = a$, $P(A = false \wedge B = false) = 1 - a$ (and P=0 for the 2 others events). Assume that $bn = \{P(A), P(B)\}$ (so the structure bn assume the independence).

Then calibrating bn on $\hat{P}$ leads to $ibn(A) = \hat{P}(A) \rightarrow a$, $ibn(B) = \hat{P}(B) \rightarrow a$. $N(ibn)$ (equals to L plus a constant) is, for $x = ibn(A)$ and $y = ibn(B)$ (i.e. x is the probability given by the bayesian network ibn for the event $A = true$, and y the probability for $B = true$) :

$$(xy - a)^2 + x^2(1 - y)^2 + y^2(1 - x)^2 + ((1 - x) * (1 - y) - (1 - a))^2$$

the derivative of this expression w.r.t x (as well as w.r.t. y), in $x = a, y = a$ is positive for $0 < a < \frac{1}{2}$ and negative for $\frac{1}{2} < a < 1$. So, the solution $x = a, y = a$ is not the minimum of this equation except if $a = \frac{1}{2}$. ■

6.3. *Universal consistency and bound with a finite number of variables*

We assume that a heuristic system is given in order to rank dependencies between variables, for the building of the structure. This method, whenever required, provides a dependency $A_j \rightarrow A_i$ that increases a dependency $P(A_i|A_{K_i})$ to a dependency $P(A_i|A_{K_i \cup \{j\}})$. This method is designed to increase step by step the complexity of the structure.

Consider the following algorithm, for $\epsilon(n)$ a sequence converging to 0 as $n \rightarrow \infty$:

– Consider n the number of examples and δ the risk threshold chosen by the user ;

– Heuristically sort the list of dependencies (possibly using a separate database) ;

– As long as the next dependency added to bn does not lead to $F(bn,\delta)/\sqrt{n} > \epsilon(n)$, add the dependency the most suitable according to the heuristic ;

– Choose $i\hat{bn} \in bn$ minimizing $\hat{L}$;

– Claim $L(ibn) \leq \hat{L}(ibn) + F(bn,\delta)/\sqrt{n}$.

Corollary C3 :

– with confidence at least $1-\delta$, the bound provided on $L(ibn)$ is true ;

– in the limit of a large number of examples, $L(i\hat{bn})$ converges to $inf_{ibn}L(ibn)$ (inf among any ibn, independently of the structure, and not only $inf_{ibn\in bn}L(ibn)$), at least if the heuristic, within a finite number of increases of the structure, leads to bn such that $inf_{ibn\in bn}L(ibn) = inf_{ibn}L(ibn)$ (this is a small and natural hypothesis as the heuristic can simply lead to the complete graph between observable variables if the number of dependencies is sufficiently large).

The proof is a consequence of the convergence of $F(bn,\delta)/\sqrt{n}$ to 0 (as it is upper bounded by $\epsilon(n)$) as $n \to \infty$.

6.4. *Universal consistency and confidence intervals with infinitely many variables*

We consider here an infinite number of states, but a finite number of examples. Variable j of example i is noted $a_{i,j}$. The sequence of vectors [1] $(a_{i,1},\ldots,a_{i,743},\ldots)$ for $i \in \mathbb{N}$ is assumed independently identically distributed. The algorithm is as follows :

1) the user provides n, ϵ and δ ; an oracle provides the $a_{i,j}$ when they are required by the program.

2) evaluate bn maximal for the inclusion[2] (chosen by any heuristic among multiple possible solutions, provided that bn increase as n increases), such that $F(bn,\delta)$ is upper-bounded by ϵ ; the variables modelled by b_n are the observable ones among the union of the A_j and A_{K_j} such that bn is defined by the $P(A_j|A_{K_j})$;

3) choose $ibn \in bn$ minimizing $\hat{L}$;

4) provide to the user a bound $L(ibn) \leq \hat{L}(ibn) + F(bn,\delta)/\sqrt{n}$;

Corollary C4 :

Let's note $mod(bn)$ the set of events which are deterministic functions of observable variables modelled by bn.

– for any E event depending upon a finite number of A_j, $ibn(E)$ is evaluated if n is large enough and its value converges to $P(E)$ as $n \to \infty$, if at least the heuristic method guarantees that for a given increasing sequence of integers k_i, the number of

1. There are infinitely many vectors but these vectors are countable.
2. We say that a bayesian network bn_1 is included in a bayesian network bn_2 if any dependency in bn_1 is a dependency in bn_2 within a renumbering of latent variables.

dependencies is bounded by k_i as long as the i^{th} observable variable is not added to the network (this is a natural requirement).

– the bound provided on $L(ibn)$ holds with probability at least $1-\delta$.

– thanks to the Borell-Cantelli lemma (see e.g. [VID 97, p26]), one can write that if $\sum_n \delta_n$ is finite (for example $\delta_n = 1/n^2$) and if $F(bn_n, \delta_n)/\sqrt{n} \to 0$ as $n \to \infty$, with bn_n the structure chosen for a number n of examples, then there is almost sure convergence of $\sup |P(E) - ibn(E)|$ for $E \in mod(b_n)$ to 0 ; we must ensure $\delta_n \leq \delta$ to assert, moreover, that the bound $\hat{L}(ibn) + F(bn, \delta)/\sqrt{n}$ holds.

6.5. *Universal consistency and convergence to the right network of dependencies*

We propose in this section an algorithm in order to build bayesian networks having two important properties :

– it is universally consistant ;

– the size of the structure converges to the optimal one.

The second point is not trivial, as it is very difficult to guarantee convergence to a non-redundant structure.

Precisely, we claim the

Theorem 8 : universal consistency and convergence to the right structure

Define

$$ibn \in argmin_{U(ibn)\leq n} \hat{L}(ibn) + R(ibn, n)$$

where U is an application which associates a real number to any instantiated bayesian network, such that $\forall (ibn_1, ibn_2) \in bn\ U(ibn_1) = U(ibn_2)$ (i.e., two bayesian networks having the same structure have the same image through U) , and where $R(ibn, n) = R'(ibn)R(n)$ associates a real number to an instantiated bayesian network ibn and to a sample size n.

We note in the sequel (by abuse of notation) $U^{-1}(n) = \{ibn; U(ibn) \leq n\}$.

Then :

1) **universal consistency :** if H0, H1 and H2 hold, then $L(ibn)$ almost surely goes to L^* ;

2) **convergence of the size of the structure :** if H0, H1, H2 and H3 hold, then $R'(ibn) \to R'(ibn^*)$ where ibn^* is such as $L^* = L(ibn^*)$.

H0 : for n sufficiently large, $ibn^* \in U^{-1}(n)$;

H1 : $sup_{ibn \in U^{-1}(n)}\ R'(ibn)R(n) \to 0$ as $n \to \infty$;

H2 : $F(U^{-1}(n), 1/n^2)/\sqrt{n} \to 0$ as $n \to \infty$;

H3 : $F(U^{-1}(n), 1/n^2)/(R(n)\sqrt{n}) \to 0$ as $n \to \infty$;

Proof :

Define $bn = U^{-1}(n)$ and $\epsilon(bn, n) = sup_{ibn \in U^{-1}(n)} |\hat{L}(ibn) - L(ibn)|$.

Let's proof the universal consistency under hypothesis H0, H1, H2.

$$L(ibn) \leq \hat{L}(ibn) + \epsilon(bn, n)$$
$$\leq \inf_{ibn' \in bn} \hat{L}(ibn') + R(ibn', n) - R(ibn, n) + \epsilon(bn, n)$$
$$\leq \inf_{ibn' \in bn} L(ibn') + \epsilon(bn, n) + R(ibn', n) - R(ibn, n) + \epsilon(bn, n)$$
$$\leq \inf_{ibn' \in bn} L(ibn') + R(ibn', n) + 2\epsilon(bn, n)$$

Thanks to H1, we only have to prove that $\epsilon(bn, n) \to 0$ almost surely.

By definition of $F(.,.)$, $P(\epsilon(bn, n) \geq F(bn, 1/n^2)/\sqrt{n}) \leq 1/n^2$.

In particular, for any ϵ, H2 implies that for n sufficiently large, $F(bn, 1/n^2)/\sqrt{n} < \epsilon$, and so $P(\epsilon(bn, n) > \epsilon) \leq 1/n^2$. Thanks to the Borell-Cantelli lemma, the sum of the $P(\epsilon(bn, n) > \epsilon)$ being finite for any $\epsilon > 0$, $\epsilon(bn, n)$ almost surely converges to 0.

We have achieved the proof of consistency. We now start the proof of the convergence of the size of the structure.

Thanks to H0, if n is sufficiently large, $ibn^* \in bn$. We restrict our attention to such n.

$$\hat{L}(ibn) + R(ibn, n) \leq \hat{L}(ibn^*) + R(ibn^*, n)$$
$$R'(ibn)R(n) \leq R'(ibn^*)R(n) + \hat{L}(ibn^*) - \hat{L}(ibn)$$
$$R'(ibn)R(n) \leq R'(ibn^*)R(n) + L^* + 2\epsilon(bn, n) - L(ibn)$$
$$R'(ibn) \leq R'(ibn^*) + 2\epsilon(bn, n)/R(n)$$

It is then sufficient, using H3, to show that $\epsilon(bn, n)/R(n) \to 0$ almost surely. Let's show this by Borell-Cantelli as well. By definition of $F(.,.)$, $P(\epsilon(bn, n) \geq F(bn, 1/n^2)/\sqrt{n}) \leq 1/n^2$.

In particular, for any ϵ, H3 implies that for n sufficiently large, $F(bn, 1/n^2)/(R(n)\sqrt{n}) < \epsilon$, and so $P(\epsilon(bn, n)/R(n) > \epsilon) \leq 1/n^2$. Thanks to the Borell-Cantelli lemma, the sum of the $P(\epsilon(bn, n)/R(n) > \epsilon)$ being finite for any $\epsilon > 0$, $\epsilon(bn, n)/R(n)$ almost surely converges to 0. ■

7. Algorithmic

We have shown in sections above that the optimization of $\hat{L}$ leads to better generalization properties than the usual local method. Unfortunately, in its basic form, $\hat{L}$ is difficult to evaluate and to optimize. We propose :

– other more practical formulations of $\hat{L}$, and algorithms for computing it (section 7.1),

– methods for adapting these algorithms to the computation of the gradient (section 7.2).

– optimization methods (7.3), including adaptive precision (based on estimates of the precision of the computation of the gradient) and BFGS.

7.1. *Objective functions*

We here present in the following sections :

– a reformulation of the loss function $\hat{L}$;

– an exact method for the computation of $\hat{L}$;

– a Monte-Carlo method for the computation of $\hat{L}$;

– a method inspired by the quota method for the computation of $\hat{L}$;

7.1.1. *Introduction*

Lemma : $\hat{L}(Q) = 1 + S + \frac{1}{n}\sum_{e=1}^{n} -2Q(i_e)$ with n the number of examples, i_e the number of the tuple representing the example e (if e is the example where all the variables are false, then $i_e = 1, \ldots$), and $S = \sum_{i=1}^{2^a} Q(i)^2$.

Proof : $\hat{L}(Q) = \hat{E}\sum |Q - \chi|^2 = \hat{L}(Q) = \frac{1}{n}\sum_{e=1}^{n}\sum_{i=1}^{2^a}(Q(i) - \chi(e))^2$ with $\chi(e)$ the vector χ representing the example e.

$$\hat{L}(Q) = \frac{1}{n}\sum_{e=1}^{n}\left((Q(i_e) - 1)^2 + \sum_{i=1, i\neq i_e}^{2^a} Q(i)^2\right)$$

$$\hat{L}(Q) = \frac{1}{n}\sum_{e=1}^{n}\left(-2Q(i_e)^2 + 1 + \sum_{i=1}^{2^a} Q(i)^2\right)$$

$$\hat{L}(Q) = 1 + S + \frac{1}{n}\sum_{e=1}^{n} -2Q(i_e)^2$$

■

The term $\sum_{e=1}^{n} -2Q(i_e)^2$ is easily tractable, as it can be computed in $O(an)$. Hence, computing $\hat{L}(Q)$ is difficult due to S.

We then propose other formulations of S that allow computational feasibility.

Remark : Many works have been devoted to the computation of sums of probabilities (marginalization for inference). S is a sum of squared probabilities so it is likely that techniques like those involved in [LAU 88, COZ 00, KSC 01, GUO 02] could be

applied in this context also. We can then expect huge improvements in our computation times/precisions.

7.1.2. *Properties of the objective function*

As we want to optimize $\hat{L}$, we want to examine the properties of the objective function. The gradient issue will be examined in section 7.2.

The S term (see above) is convex, and one could think that $\hat{L}$ is also convex. The convexity of $\hat{L}$ depends on the distribution of the examples, by the term $-\frac{2}{n}\sum_{e=1}^{n} Q(i_e)^2$. The figure 3 shows a counter example of an objective function $\hat{L}$ which is not convex. However, we have observed experimentally that $\hat{L}$ is often roughly convex.

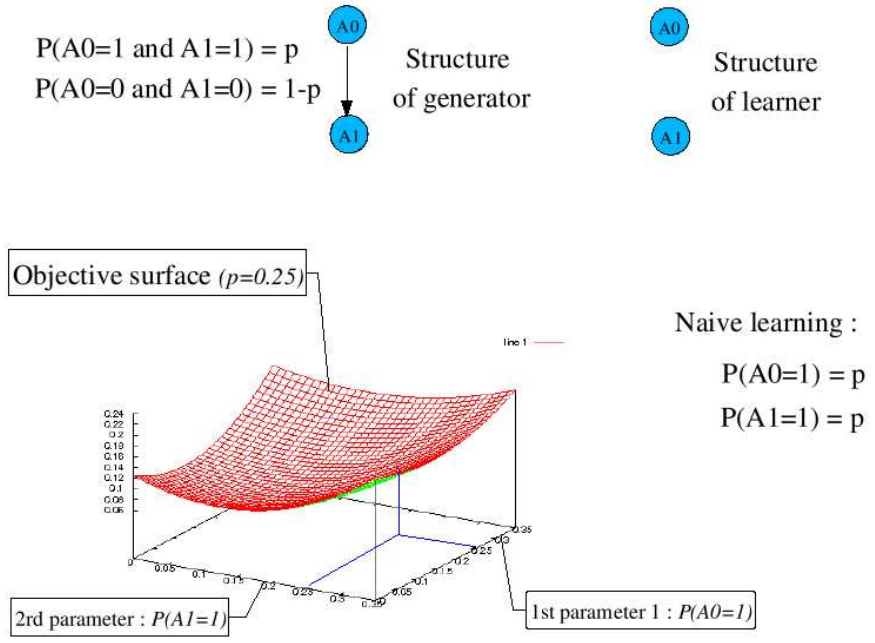

Figure 2. *An illustration of why optimizing globally the parameters is a good idea. We plot here the objective surface ($\hat{L}$) in function of the two parameters for a 2 nodes "naive" bayesian network. The structure of the learner is presented on the up-right of the figure. The structure of the generator of the law is presented on the up-left of the figure. The parameters coming from frequentist learning method is represented by the intersection of the three lines parallel to the axis on the three dimensional graphic. We see that this point is not the optimum of the objective function.*

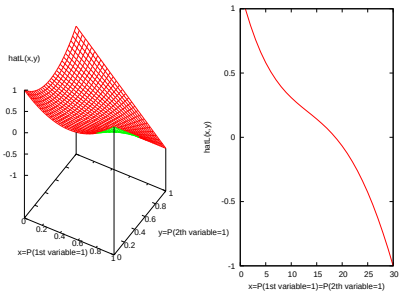

Figure 3. ***Left hand graph*** *: objective function $\hat{L}$ in function of the two parameters for a 2 nodes "naive" bayesian network.* ***Right hand graph*** *: diagonal cut of the left hand graph. This shows that the objective function $\hat{L}$ can be no convex.*

7.1.3. *Exact method for the evaluation of S*

Thanks to the decomposition of the product law we propose an algorithm to compute S in less than 2^a operations. The number of operations required to compute S depends on the structure of the bayesian network. A simpler structure leads to less calculus.

Roughly speaking, the main ideas are as follows :

– we start with a set of nodes F_1, called the front, reduced at the empty set at the beginning, and that moves across the bayesian network (roughly, in "reverse order" for the topological order, as far as this notion makes sense for a set of nodes), and C_1, initialized to the empty set.

– during all the process, S is known at all the nodes in the front.

– for $t = 1, \ldots, a$, the front evolves as follows :

- a_t is the t^{th} node in reverse topological order (multiple possible choices are heuristically decided as explained below) ;

- we add this point to the front ;

- we add in the front all points necessary for its consistency,

- S is computed for any new element in the front so that S is always known for all elements in the front.

F^1 and C^1 are the empty list.

For any $t \in [[1, a]]$,

– F^t is a list of f_t subsets of $[|1, a|]$; $F^t = (F_1^t, ..., F_{f_t}^t)$, is initialized for $t = 1$ at the empty list $F^1 = ()$;

– C^t is a list of f_t subsets of $[|1, a|]$; $C^t = (C_1^t, ..., C_{f_t}^t)$, is initialized for $t = 1$ at the empty list $C^1 = ()$;

– dom_i^t is a subset of $[[1, a]]$;

– L^t is a list of f_t applications from dom_i^t to $\{0, 1\}$.

– $a_t \in [|1, a|]$ is the node chosen (the choice is performed as defined below) at step t.

They are defined by induction ($t \in [[1, a]]$) by :

– a_t is chosen among the last nodes in topological order among the nodes that are different of the a_s for $s < t$ (i.e. $a_t \in [[1, a]] \setminus \{a_s; s < t\}$ and a_t has no successor in $[[1, a]] \setminus \{a_s; s < t\}$) ; if many a_t are possible, it is chosen such that $|C'^t|$ (defined below) is minimal ;

– $I_t = \{i \in [1, f_t]/a_t \in C_i^t\}$ is a list of integers ;

– $C'_t = \bigcup_{i \in I_t} C_i^t \bigcup K_{a_t} \setminus \{a_t\}$ is a subset of $[|1, a|]$;

– $C^{t+1} = (C_i^t)_{i \notin I_t, 1 \leq i \leq f_t}.(C'_t)$ where $a.b$ is the concatenation of lists a and b ;

– $S^{t+1} = (S_i^t)_{i \notin I_t, 1 \leq i \leq f_t} \cdot \left(c' \in 2^{C'_t} \mapsto \sum_{a_t} P(a_t|c')^2 \prod_{i \in I_t} S_i^t(c'_{|dom_i^t})\right)$

(where $a_{|b}$ is the restriction of a to the domain b)

– $F^{t+1} = (F_i^t)_{i \notin I_t, 1 \leq i \leq f_t} \cdot \left(\bigcup_{i \in I_t} F_i^t \cup \{a_t\}\right)$

– $f_{t+1} = |F^{t+1}|$ (length of the list)

– $dom^{t+1} = (dom_i^t)_{i \notin I_t, 1 \leq i \leq f_t} .2^{(}C'_t)$

S is equal to the product of the S^t.

One can verify by induction that for any $t \in [[1, a]]$, for any $1 \leq i \leq f_t$, S_i^t, the following holds : $S_i^t : c \mapsto \sum_{v \in 2^{F_i^t}} P(v|c)^2$. This implies the consistency of the exact method.

7.1.4. *Approximate methods for the computation of S*

$\hat{L}$ and its gradient are hard to compute. In this section we define algorithms approximating S in an efficient manner. The most simple is Monte-Carlo, and we define improvements of Monte-Carlo based on regular samplings.

$S = \sum_{i=1}^{2^a} Q(i)^2$ can be written as $S = EQ(i)$, E being the expectation under the law Q. We are going to approximate this expectation thanks to a finite sample drawn according to law Q (possibly with some bias in order to improve the precision, see below). Now, $\widehat{\hat{L}}$ is the approximate of $\hat{L}$ where S is replaced by an empirical mean on a finite sample.

S is the most computationally expensive term of $\hat{L}$, the other one (see section 4.1) being computable in an exact manner. We present here the Monte-Carlo method, and improved other methods, for the computation of S. The same methods can also be used for the computation of ∇S (see section 7.2) , both of them being necessary either for the gradient descent or for the BFGS method.

We also present the estimation of variance in the case of Monte-Carlo ; the same estimate will be used for other approximate methods.

7.1.4.1. Monte-Carlo method for the computation of S

The most straightforward solution is the Monte-Carlo method : just simulate the law Q associated to the network and average the results. S is therefore approximated by $\sum_{j=1}^{n} Q(e_j)$ where the e_j are i.i.d among $0, 1^a$ with distribution of probability Q.

Now, let's consider the estimation of variance. We consider the case of the approximation of $\nabla \hat{L}$, the case of $\hat{L}$ being similar (just consider dimension 1).

For the sake of clarity, we note $g = \nabla \hat{L}$ the exact gradient and $\hat{g} = \widehat{\nabla \hat{L}}$ the approximate gradient. Then,

$$||\hat{g} - g||^2 = \sum_{i=1}^{d} (\hat{g}_i - g_i)^2 \simeq \sum_i (\frac{\sigma_i N_i}{\sqrt{n}})^2 = \frac{1}{n} \sum_i \sigma_i^2 N_i^2$$

We now assume independence of the N_i. This is an approximation. Then,

$$E||\hat{g} - g||^2 = \frac{1}{n} \sum_i \sigma_i^2 E N_i^2 = \frac{1}{n} \sum_i \sigma_i^2 \text{ as } EN_i^2 = 1$$

$$Var||\hat{g} - g||^2 = \frac{1}{n^2} \sum_i \sigma_i^4 Var(N_i^2) = \frac{2}{n^2} \sum \sigma_i^4 \text{ as } Var(N_i) = 2$$

where the N_i are independent standard normal variables (expectation 0 variance 1), σ_i is the standard deviation of the gradient restricted to coordinate i, n is the number of draws for the Monte-Carlo method.

We can then use as bound on $||\hat{g} - g||^2$ a formula like $\frac{1}{n}\left(\sum \sigma_i^2 + \sqrt{2 \sum \sigma_i^4}\right)$.

7.1.4.2. Quotas Method for the computation of S

A more stable solution is defined as follows. We consider the 2^a possible values of the whole set of variables, in lexicographic order, with their probabilities q_1, q_2, ..., q_{2^a}. Then, we consider $x_i = \frac{2i-1}{2 \times 2^a}$ for $i = 1, \ldots, 2^a$. Then, we consider the average of the q_{j_i} where j_i is minimal such that $\sum_{h=1}^{j_i} q_h \geq x_i$.

If the lexicographic order is with respect to an ordering of variables in topological order, this is easy to implement until large number of examples.

7.2. *Computation of the gradient*

The gradient of S is the main difficulty in the computation of the gradient of $\hat{L}$. We show here how ∇S can be evaluated in a similar manner as S.

Consider the following high-level definition of S :

$$S = \sum_j S_j$$

where j is an index on all possible assignments of the a variables, and

$$S_j = \Pi_{i \in I_j} p_i^2 \Pi_{i \in I'_j} (1 - p_i)^2$$

where $\forall j; I_j \cap I'_j = \emptyset$ and $\forall i; |\{j; i \in I_j \cup I'_j\}| = 1$. Then

$$\frac{\partial S_j}{\partial p_i} = 0 \text{ if } i \notin I_j \cup I'_j$$

$$= 2S/p_i \text{ if } i \in I_j$$

$$= -2S/(1 - p_i) \text{ if } i \in I'_j$$

So, the Monte-Carlo method can be adapted in the following manner :

– draw examples as in the computation of $\hat{L}$;

– for each example, adapt the at most a parameters that are concerned (one per variable).

So, for a given number of examples, the algorithmic complexity is at most multiplied by a. The quota method can be adapted in the same way.

The exact method can be adapted in the following manner :

– for each parameter p_i of the bayesian network :

- fix the value of the parent-variables so p_i is relevant ;
- evaluate S for the bayesian network with these fixed values ;
- apply formulas above providing $\partial S/\partial p_i$.

BFGS is able of approximating the hessian thanks to successive gradients in a very efficient manner ; so the gradient will be enough for optimization below.

7.3. *Optimization*

We now turn our attention to the optimization methods suitable for $\hat{L}$.

Gradient descent is a very simple solution for non-linear optimization. It is used for comparison with BFGS. BFGS is a classical algorithm for non-linear optimization, with the following characteristics :

– superlinear in many cases ;

– needs only the gradient and approximates the hessian thanks to the successive values of the gradient.

We used Opt++ and LBFGSB, freely available on the web, as BFGS optimization softwares. The results presented below come from LBFGSB, a limited BFGS algorithm for bound-constrained optimization.

8. Experiments

We defined in section (5) some objective functions with good statistical properties. We defined in 6 some algorithms for the optimization of these objective functions.

We present in the following subsections :

– the questions (both about statistical significance and algorithmic complexity) that we want to answer by empirical studies ;

– the empirical results.

We aim at answering the following questions :

1) is the entropy of the network relevant or just a second-order theoretical point ?

2) are our algorithms for the computation of S and its derivative efficient in the following cases :

- exact method ;

- Monte-Carlo method with/without random seed or with quota ;

in particular, depending upon the dimension/sample size.

3) is the superiority of the optimization of $\hat{L}$ on the local method as shown in section 6.2 validated by practical experiments ?

8.1. *Is the entropy of the network relevant or just a second-order theoretical point ?*

We have shown in theorem 7 and in section 5.2 that the deviations $L - \hat{L}$ were bounded above by a term depending on the *entropy* of the network, and not only on the number of parameters. We now experiment this element as follows :

– generate randomly a bayesian network,

– randomly draw a data set D with this bayesian network,

– for many values of k :

- generate randomly many learners $l_1, \ldots, l_m$ with k parameters, with entropy $H_1, \ldots, H_m$;

- learn (i.e. optimize $\hat{L}$) with each of these learners. Define $\hat{L}_i$ the empirical error of l_i.

- evaluate L for each of these learning ; define L_i the generalization error of learner l_i.

- plot $L_i - \hat{L}_i$ as a function of H_i

We show the result with 10 nodes in figure 4. Hence, this shows experimentally that, with a number of parameters fixed, the entropy term reflects well the complexity of the structure, whereas it doesn't appear in usual scores. This confirms the theoretical results.

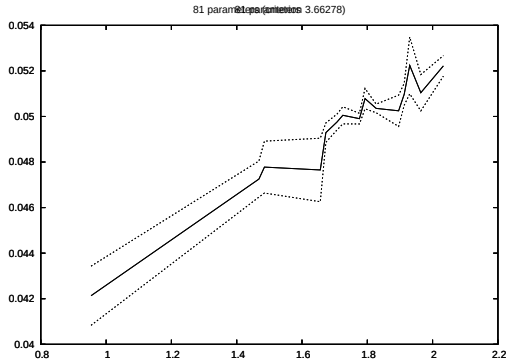

Figure 4. *X-ordinate : entropy. Y-ordinate : average of $L_i - \hat{L}_i$ ($\pm$ standard deviation). The positive correlation is clear.*

8.2. *Are our algorithms for the computation of S and its derivative efficient ?*

We below perform experiments in order to validate the approximation of S and ∇S (sub-section 1) and then test them inside an optimization loop (sub-section 2).

8.2.1. *Preliminary experiments on the approximations of ∇S*

We compare below i) the exact method ii) the Monte-Carlo method iii) the quota method.

The experimental setup is as follows : 10 bayesian networks are generated ; their gradients are computed with each method ; we compute the relative error ; we averaged the results. We experimented random bayesian networks : each node i has 2 parents randomly drawn among $[[i-6, i-3]]$ and 2 parents $i-2$ and $i-1$.

The results are the following for the computation of the derivative. The sample size is the sample size of the approximate methods. These experiments have been ran on a pentium 4, 3.0 GHz.

Algorithm	Time	Relative error
nb nodes=20, sample size = 10000		
Exact	0.68± 0.07	0.
Monte-Carlo	0.04 ± 0.003	0.07 ± 0.02
Quotas	0.04 ± 0.005	0.01 ± 0.003
nb nodes=30, sample size = 10000		
Exact	1.85± 0.15	0.
Monte-Carlo	0.06± 0.004	0.11± 0.02
Quotas	0.06± 0.000	0.05± 0.03
nb nodes=50, sample size = 30000		
Exact	6.26± 0.33	0.
Monte-Carlo	0.29± 0.01	0.19± 0.04
Quotas	0.30± 0.01	0.17± 0.04

The results are :

– the exact method is validated (error 0 at each run) ;

– results are better for the quotas method than for the naive Monte-Carlo method, at least for a moderate number of parameters. For huge dimension (50 nodes, roughly 700 parameters), the quotas method is roughly equivalent to the Monte-Carlo method ; as usually in the general case of quasi-Monte-Carlo methods, the case of huge dimension is difficult (see e.g. [SRI 00]).

8.2.2. *Optimization through the approximate computation of S and ∇S*

We have shown above that approximate methods are precise and fast. We now show that the whole optimization algorithm based on the approximate methods are reliable. The goal is to find optimal values of parameters of the BN for $\hat{L}$. We therefore plot the evolution of the (exact) objective function $\hat{L}$ when we use BFGS with the estimators of S and the estimate of ∇S.

The experimental setup is as follows :

– define $\hat{\hat{L}}$ the approximation of $\hat{L}$ by the quota method ;

– optimize $\hat{\hat{L}}$ thanks to BFGS ; increase the number of examples when

$$Var(\widehat{\nabla \hat{L}}) \geq \alpha \times ||\widehat{\nabla \hat{L}}||^2$$

for typically $\alpha = 0.1$ where

- $\hat{L}$ is the objective function (see section 4.1) ;

- $\widehat{\nabla \hat{L}}$ is the estimate of the gradient of $\hat{L}$ through the quota method ;

- $Var(\widehat{\nabla \hat{L}})$ is the variance of the estimate of $\nabla \hat{L}$ by the quota-method,

The structure of the bayesian networks used for learning is as shown in section 8.2.1. This generation is in favor of the exact method as the network has bounded

width. The results are presented in figure 5. Thanks to the particular structure of the network, the exact method remains very efficient even for 30 nodes ; the approximate method is however faster than the exact one. Note that the approximate method can deal with the general case, whereas the exact one might be no more tractable.

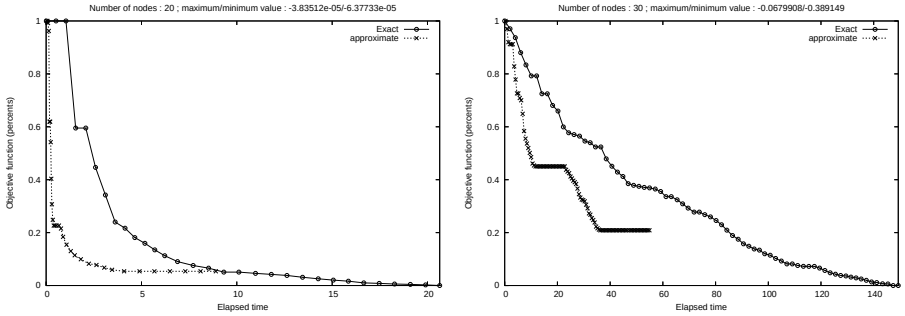

Figure 5. *Evolution of $\hat{L}$ as time increases, when i) BFGS uses the exact $\nabla\hat{L}$ and $\hat{L}$; ii) BFGS uses the approximate $\widehat{\nabla\hat{L}}$ and $\widehat{\hat{L}}$. First : 20 nodes. Second : 30 nodes. The structure and parameters are randomly drawn as explained in section 8.2.1. We see that the computations are much faster with the approximate methods.*

These experiments have shown that the optimization is pratical, and so we demonstrate the "proof of concept" of the method. In order to treat much larger bayesian networks, we can expect huge improvements as pointed out in 7.1.1 from adaptation of state of the art inference algorithms.

8.3. *Is the superiority of the optimization of $\hat{L}$ on the local method as shown in section 6.2 validated by practical experiments ?*

The experimental setup is as follows :

– randomly draw one generator G ;

– randomly draw structure S for learning ;

– generate n examples from G ;

– learn a bayesian network with structure S on these examples ;

– compute L with respect to the distribution associated to G.

In our experiments, all networks have size 10 nodes, and for the sake of statistical significance, 10 different generators were drawn and tested, and the random draws of the structures were paired (i.e., the 50 different S drawn for each generator are the same for all the generators). We tested n between 100 and 900. The results are presented in figure 6.

We plot the difference between the generalization error and the generalization error of the generator (best possible error with this distribution). Zero error can then be

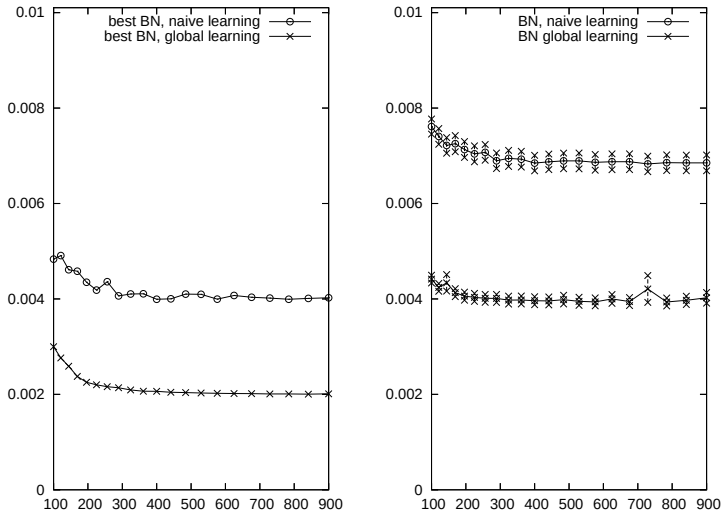

Figure 6. *X-coordinate : Number of examples. Y-coordinate : Error in generalization minus optimal generalization error ($L - L(g)$ where g is the generator). Results are averaged among 10 randomly drawn generators and 50 randomly drawn structures for learning.*

achieved with a perfect structural learning algorithm, and perfect parameters. Here we can experimentally see that the global optimization method divide by a factor of two the error, without any structural learning algorithms, which are generally very costly and suboptimal.

9. Conclusion

In this paper, we have proposed insights in bayesian network learning using statistical learning theory.

Among all, we

1) proposed a criterion of quality of an instanced bayesian network, and evidences of its relevance for some applications ;

2) have proved bounds on covering numbers of bayesian networks (theorem 7, section 5) ;

3) proposed scores for choosing between structures, with a score taking into account the structural entropy of the network. Multiple corollaries have been shown in section 6.

4) proposed a paradigm for parameter-learning which is better than the traditional frequentist method.

5) proposed an algorithm with guaranteed universal consistency and almost sure convergence towards a structure with optimal size (theorem 8) ;

We then proposed new algorithms (section 7) in order to treat our loss function which is more complicated than the frequentist parametrization. Section 7 shows experimentally the relevance of the work and the adequacy between the theoretical results and experiments.

The drawback of the loss function is the computation over head for learning the parameters. However, in order to learn in larger bayesian networks, we can expect improvement using adaptations of inference algorithms(see remark in 7.1.1). Futhermore, the experiments in 8.3 show that we can in some respect avoid a very costly structural learning procedure.

Moreover, the entropy term in the estimation of the overfitting of a structure is theoretically and experimentally shown (theorem 7 and section 8.1). More precise structural learning experiments, using a score taking into account the entropy of the structure, have yet to be conducted.

Acknowledgements

We thank the Pascal network of Excellence for its support to the IA/TAO team. We thank the authors of the freely available Opt++ [MEZ 94, MEZ] and LBFGSB library [ZHU 94, BYR 95].

10. Bibliographie

[AKA 70] AKAIKE H., « Statistical predictor identification », *Ann. Inst. Statist. Math., 22 :203-217*, 1970.

[ANT 99] ANTONY M., BARTLETT P., « Neural network learning : Theoretical Foundations », *Cambridge University Press*, 1999.

[BOR 02] BORGLET C., KRUSE K., « Graphical Models - Methods for data analysis and Mining », *John Wiley and Sons, Chichester, UK*, 2002.

[BYR 95] BYRD R., LU P., NOCEDAL J., C.ZHU, « A limited memory algorithm for bound constrained optimization », *SIAM J. Scientific Computing, vol.16, no.5*, , 1995.

[CHE 97a] CHENG J., BELL D., LIU W., « An algorithm for bayesian network construction from data », 1997.

[CHE 97b] CHENG J., BELL D., LIU W., « Learning belief networks from data : An information theory based approach », 1997.

[CHE 02] CHENG J., GREINER R., KELLY J., BELL D., LIU W., « Learning Bayesian networks from data : An information theory based approach », 2002.

[CHI 02] CHICKERING D. M., « Learning equivalence classes of bayesian-network structures », *The Journal of Machine Learning Research, Volume 2 , (March 2002), Pages : 445 - 498*, , 2002.

[COO 92] COOPER G., HERSOVITS E., « A Bayesian method for the induction of probabilistic networks from data », *Machine Learning, 9 :309-347*, 1992.

[COZ 00] COZMAN F., « Generalizing Variable Elimination in Bayesian Networks », , 2000.

[GUO 02] GUO H., HORVITZ E., HSU W., SANTOS E., « A Survey of Algorithms for Real-Time Bayesian Network Inference », *Working Notes of the Joint Workshop (WS-18) on Real- Time Decision Support and Diagnosis, AAAI/UAI/KDD-2002. Edmonton, Alberta, CANADA, 29 July 2002. Menlo Park, CA : AAAI Press*, , 2002.

[HEC 94] HECKERMAN D., GEIGER D., CHICKERING M., « Learning Bayesian networks : The combination of knowledge and statistical data », *Ramon Lopez de Mantaras et David Poole, editors, Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, 1994.

[KOL 61] KOLMOGOROV A.-N., TIKHOMIROV V.-M., « ϵ-entropy and ϵ-capacity of sets in functional spaces », *Amer. Math. Soc. Transl. 17, pp 277-364*, 1961.

[KSC 01] KSCHISCHANG F. R., FREY B., LOELIGER H.-A., « Factor graphs and the sum-product algorithm », *IEEE Trans. Inform. Theory*, vol. 47, n° 2, 2001, p. 498-519.

[LAU 88] LAURITZEN S., SPIEGELHALTER D., « Local computations with probabilities on graphical structures and their applications to expert systems », *J. Royal Statistical Society B, 50, pp. 157-224*, , 1988.

[MEZ] MEZA J., HOUGH P., WILLIAMS P., « Opt++ ».

[MEZ 94] MEZA J. C., « OPT++ : An Object-Oriented Class Library for Nonlinear Optimization », , 1994.

[NAI 04] NAIM P., WUILLEMIN P.-H., LERAY P., POURRET O., BECKER A., « Réseaux bayésiens », *Eyrolles*, 2004.

[PEA 00] PEARL J., « Causality : models, reasonings and inference », *Cambridge University Press, Cambridge, England*, 2000.

[RIS 78] RISSANEN J., « Modeling by shortest data description », *Modeling by shortest data description*, 1978.

[ROB 94] ROBERT C., « The Bayesian Choice : a decision theoric motivation », *Springer, New York*, 1994.

[SCH 78] SCHWARTZ G., « Estimating the dimension of a model », *The annals of Statistics, 6(2) :461-464*, 1978.

[SPI 93] SPIRTES P., GLYMOUR C., SCHEINES R., « Causation, Prediction, and Search », 1993.

[SRI 00] SRINIVASAN A., « Low-discrepancy Sets for High-Dimensional Rectangles : A Survey », *Bulletin of the European Association for Theoretical Computer Science, Number 70, pages 67-76*, 2000.

[VAP 71] VAPNIK V., CHERVONENKIS A., « On the uniform convergence of relative frequencies of events to their probabilities », *Theory of probability and its applications, 16 :264-280*, 1971.

[VID 97] VIDYASAGAR M., « A theory of learning and generalization », *Springer*, 1997.

[WOC 02] WOCJAN P., JANZING D., BETH T., « Required sample size for learning sparse bayesian networks with many variables », *LANL e-print cs.LG/0204052*, 2002.

[ZHU 94] ZHU C., BYRD R., P.LU, NOCEDAL J., « L-BFGS-B : a limited memory FORTRAN code for solving bound constrained optimization problems », , 1994.

Annexe pour le service de fabrication

Article pour la revue :

Auteurs :
Sylvain Gelly, Olivier Teytaud

Titre de l'article :
Bayesian networks : a better than frequentist approach for parametrization, and a more accurate structural complexity measure than the number of parameters

Titre abrégé :
Bayesian networks learning improvements

Traduction du titre :
00 00 00 00 00

Date de cette version :
25 octobre 2005

Coordonnées des auteurs :

– téléphone : 00 00 00 00 00
– télécopie : Roger.Rousseau@unice.fr
– Email :

Logiciel utilisé pour la préparation de cet article :
LaTeX, avec le fichier de style `article-hermes.cls`,
version 1.10 du 17/09/2001.

Formulaire de copyright :
Joindre le formulaire de copyright signé, récupéré sur le web à l'adresse
`http://www.hermes-science.com`