CONVERGENCE OF ITERATIVE ALGORITHMS FOR LEARNING BAYESIAN NETWORKS

J. C. F. da Rocha[1], A.M. Guimarães[1], V.A. Kozlowski Jr[2]

[1]Department of Informatics – State University of Ponta Grossa (UEPG)
Av. Gen. Carlos Cavalcanti, 4748 – 84030.130 – Ponta Grossa – PR

[2]Department of Odontology - UEPG

jrocha@uepg.br

Abstract. *The formalism of the bayesian networks has been employed in the development of many intelligent systems. This work considers applicatins which demand the utilization of online learning methods, more specifically methods for online parameter learning. Online learning methods update the bayesian network parameters as new data samples/observations are collected. In this context, it is import to consider the convergence of the learning method in relation to the empirical distribution of the data. Given that this work proposes a experimental protocol to quantify the convergence/divergence of models generated by online learning procedures. An application example it is also presented.*

Keywords: Iterative learning, Bayesian networks, EM algorithm

1. Introduction

Data collection is a complex task in many areas. Besides the technical challenges inherent in the methods of data acquisition is needed consider factors related to the costs and risks underlying the task. An additional difficulty occurs when fetching distributed geographically and over time. That is, in some cases data is obtained over the years in different locations and by different technicians or researchers who employ a particular procedure on a sample of the local population.

In this context, this paper considers the following problem: a team wants to develop an intelligent system [Russell and Norvig 1995] must issue forecasts or diagnosis about the state of certain variables in the domain of an application The system employs the formalism of probabilistic Bayesian networks [Pearl 1988] to model the domain and is assumed that the network should be generated by methods of machine learning [Mitchell 1997].Once the data collection process must occur over a long period of time the development team plans to test the use of iterative learning procedures. Iterative learning methods update the model when new data is entered in the database used to induce the model.

A Bayesian network is a compact representation of a joint probability distribution. The structure of a network is defined by a directed acyclic graph whose nodes symbolize and random variables and arcs represent probabilistic influences.Each node stores the conditional distributions of the variable represented by him. The automatic learning of Bayesian networks can be abstracted in two steps: learning the network structure and

training of the numerical parameters (conditional probabilities local). This paper considers only the second step.

In order to fulfill its task the development team will need criteria to select a method suitable for learning their goals [Friedman et al. 1997].Must be noted that there are several criteria to evaluate the suitability of a Bayesian network generated by machine learning methods [Provan 1994]. However, this work considers the use of metrics to assist in the selection of iterative procedures. To this end, proposes to use the measurement of the Kullback-Leibler divergence to quantify the convergence of the models generated relation to the data used in the training stage. The basic idea is to use the distance the Kullback-Leibler to measure the differences between the empirical distribution and the generated models as that the number of training examples of the base increases.

The paper is organized as follows. Section 2 presents an overview of Bayesian networks, iterative learning Bayesian networks and Kullback-Leibler measure. Section 3 describes an approach to employing the above concepts in a protocol to compare the performance of algorithms iterative. Section 4 shows an example of application. Section 5 contains the final considerations.

2. Literature Review

This section reviews some concepts used in defining the proposed experimental protocol.

2.1. Bayesian Networks

Let $\mathbb{X} = \{X_1, ..., X_n\}$ be a set of discrete random variables with sample space $\Omega_{X_i} = \{x_{i,1} \dots x_{i,n_i}\}$. A Bayesian network [Pearl 1988] $\mathcal{BN} = (\mathcal{G}, \mathcal{F})$ is defined by a directed acyclic graph $\mathcal{G}$ whose nodes are elements of $\mathbb{X}$ and the edges express dependency relationships probabilistic between variables connected. The uncertainty in relationships is encoded by a collection of functions $\mathcal{F}$ of conditional probabilities Each node X_i stores a probability table conditionals (TPC) of the form $p(X_i|pa(X_i)_1) \dots p(X_i|pa(X_i)_*)$, where $pa(X_i)$ denotes the parents of X_i in $\mathcal{G}$ and $pa(X_i)_*$ indicates a joint instation of $pa(X_i)$. Thus, if $pa(X_i)$ have r_i instantiations, the TPC of X_i has distributions $p(X_i|pa(X_i)_1) \dots p(X_i|pa(X_i)_{r_i})$.

Let $d(X_i)$ is the set of descendants of X_i in $\mathcal{G}$.The formalism of Bayesian networks assume the following Markov condition: every variable X_i is conditionally independent of the variables in $\mathbb{X} \setminus d(X_i) \cup \{X_i\}$ given the state of variables in $pa(X_i)$.The resulting structure is that a Bayesian network is an implicit representation of a joint probability distribution of $p(\mathbb{X})$. The distribution $p(\mathbb{X})$ can be calculated from $\mathcal{BN}$ using the expression:

$$p(\mathbb{X}) = \prod_{i=1}^{n} p(X_i|pa(X_i)) .$$

The Bayesian networks can be employed to solve problems of reasoning based on evidence. As an example, $\mathcal{BN}$ is a network whose graph is a tree where the root is variable called C, which symbolizes the possible category labels that can be associated with an object ,and whose leaves $X_1 ..., X_n$ represent the attributes of the object. In this network [1], the nodes $X_1 ..., X_n$ are children of C, which is indicated by $ch(C) = \{X_1 ..., X_n\}$,

[1]The topology presented in the example is of a type naive Bayes classifier [Duda and Hart 1987].

see Figure 1. Given a evidence e that reports the value some attributes of a given object O, algorithms can be employed to update belief to calculate the posterior distribution $p(C|e)$ [Zhang and Poole 1996, Pearl 1988, Neapolitan 1990]. Let $c_1, ..., c_{n_c}$ the sample space of C, as computed distribution $p(C|e)$ it is possible select the label more suitable to $mathcalO$ by choosing the hypothesis with the greater conditional probability [Friedman et al. 1997].

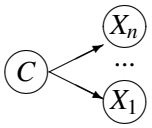

Figura 1. A Bayesian classifier.

2.2. Learning Bayesian networks

The topology and parameters of a Bayesian network can be specified directly by the development team or may be induced by machine learning algorithms [Heckerman 1995] [Krause 1998]. Currently, the possibility of storing a large volume of Data has been growing interest in automated methods.

Learning Bayesian networks can be abstracted in two steps. The first is the induction of the topology of the graph and the second is the specification of TPCs. This paper considers only the second task and therefore assumes that the network structure is fixed in a previous step of the process of knowledge engineering [Parsaye and Chignell 1988] [Pearl 1988]. The problem of *learning or training parameters* can be categorized as *complete training data* or textit training data incomplete. In training with complete data the tally for each input $P\left(x_{i,j}|pa(X_i)_l\right)$ of a TPC consisting primarily in the calculation of the frequencies observed in the training set. Training with incomplete data requires the use of methods to estimate the value of the parameters even when the values some attributes are not known in some records the training base.

2.2.1. Iterative Learning Bayesian networks

An iterative algorithm for learning parameters of a Bayesian network modifies the values of the same as new data is inserted into the training base. These algorithms assume that the initial values of the parameters of the model were specified previously subjectively or obtained from a small sample and therefore should be updated when more information was available [Bauer et al.]. Cohen, Bronstain e Cozman [Cohen et al. 2001] present an algorithm for iterative learning of Bayesian networks which is called *Voting EM*. The update rule for the case of training with complete data is described below.

Let $\mathbb{D}$ be a database whose n-tuples d_t are defined about $\times_{i=1}^{n}\Omega_{X_i}$, com $t = 1..m$. The database $\mathbb{D}$ is *training base* from which the model parameters must be calculated. The elements of $\mathbb{D}$ are called cases or examples. It is assumed that each case at $\mathbb{D}$ registers an event which was generated by a random process that agrees with the joint probability distribution $p(\mathbb{X})$.

Let $P^{T-1}\left(x_{i,j}|pa(X_i)_l\right)$ be also the input current value of the variable (l, j) of TPC of the variable X_i and $P^{T}\left(x_{i,j}|pa(X_i)_l\right)$the value of this entry after the update is performed

by the *Voting EM* algorithm. Additionally, *C1* is a logical variable which represents the condition $(P(pa(X_i)_j|d_t) = 1 \bigwedge P(x_{i,j}|d_t) = 1)$. This condition indicates that the example d_t, being processed, contains positive evidence about of $x_{i,j}$ given $pa(X_i)_l$. The term *C2* refers to the condition $(P(pa(X_i)_j|d_t) = 1 \bigwedge P(x_{i,j}|d_t) = 0)$, and indicates that d_t agrees with $pa(X_i)_l$ but no with $x_{i,j}$. Given theses conditions the adjustment rule applied by *Voting EM* algorithm is:

$$P^T\left(x_{i,j}|pa(X_i)_k\right) = \begin{cases} \eta + (1-\eta)P^{T-1}\left(x_{i,j}|pa(X_i)_l\right) & : C1; \\ (1-\eta)P^{T-1}\left(x_{i,j}|pa(X_i)_l\right) & : C2; \\ P^{T-1}\left(x_{i,j}|pa(X_i)_l\right) & \text{otherwise.} \end{cases} \quad (1)$$

In this rule the parameter $\eta \in [0,1]$ establishes a learning rate [Mitchell 1997]. If the value of η is low, close to 0, the learning process is conservative because each update promotes only minor adjustments when new cases are entered into $\mathbb{D}$. However as η increases the influence of more recent cases on the resulting model also increases.

2.3. The distance of Kullback-Leibler

As stated,the objective of this paper is to define an experimental protocol to compare the behavior of interactive algorithms with respect to their convergence to the empirical distribution of the data when the size of the training base increases. For this,it is necessary to employ a measure that quantifies the distance between the probability distribution represented by the model and the induced probability distribution of cases in the training base. Once the issue is put this way this paper proposes the use of the measure of divergence between two distributions proposed by Kullback-Leibler (*KL*) [Abbell et al. 2006].

Let p and p^* two probability distributions the measure of the Kullback-Leibler divergence is given by:

$$KL(p^*, p) = \sum_{u \in \Omega_{\mathbb{U}}} P^*(u) \log \frac{P^*(u)}{P(u)}. \quad (2)$$

Exemplo 1 *Given the set of variables* $\mathbb{X} = \{X_1, X_2\}$ *whose sample space is defined by the events* $u_1 \equiv (x_{1,1} \wedge x_{2,1}) \ldots, u_4 \equiv (x_{1,2} \wedge x_{2,2})$ *and the distributions* $p_1(\mathbb{X}) = (0,2;0,3;0,1;0,4)$ *and* $p^* = (0,3;0,1;0,3;0,3)$ *the Kullback-Leibler distance between* p_1 *and* p^* *é 0,11.*

Thus, if $p_{\mathbb{D}}$ is the empirical distribution of the data and $p(\mathbb{X})$ is the joint distribution associated with the Bayesian network $\mathcal{BN}$ generated by an algorithm A, who want to test, measure the distance of $p(\mathbb{X})$ against [2] to $p_{\mathbb{D}}$ is:

$$\Delta(\mathtt{A}) = KL(p_{\mathbb{D}}, p(\mathbb{X})) = \sum_{d_t \in \mathbb{D}} P_{\mathbb{D}}(d_t) \log \frac{P_{\mathbb{D}}(d_t)}{P(d_t)}, \quad (3)$$

where d_t is an instantiation of the variables in $\mathbb{X}$.It should be noted that since $\mathcal{BN}$ encodes the distribution $p(\mathbb{X})$ is possible to explore the network structure to calculate $P(d_t)$ efficiently [Pearl 1988].

[2] The distance Kullback-Leibler measure is not symmetric.

3. Convergence of iterative algorithms for learning

This section describes an experimental protocol for compare the convergence of different algorithms for iterative learning Bayesian networks. Let $\mathtt{A}_1, \mathtt{A}_2, ...\mathtt{A}_\mathtt{L}$ the algorithms to be tested the central procedure of the protocol is to calculate the values of $\Delta(\mathtt{A}_1)$ (Equation 3), as follows:

- PROCEDURE 1
    1. train Bayesian networks with iterative algorithms $\mathcal{A}_1 ... \mathcal{A}_L$; this step results in the Bayesian networks $\mathcal{BN}_1$, $\mathcal{BN}_2$ ... $\mathcal{BN}_L$;
    2. calculate $\Delta(\mathcal{A}_1) ... \Delta(\mathcal{A}_L)$ to the bases generated.

The values computed by the above procedure allows to compare the convergence of the algorithms at a given instant of iterative learning when the training base has m cases. Performing an empirical analysis of behavior of the algorithms tested for the growth of the database requires you to obtain indicative of convergence throughout the learning process. Thus, the training sets are $\mathbb{D}_1 ... \mathbb{D}_T$ such that: (a) $\mathbb{D}_t \subset \mathbb{D}_{t+1}$; and (b) m_t first cases of $\mathbb{D}_{t+1}$ are those that compose $\mathbb{D}_t$. The procedure presented below specifies the manner in which the proposed protocol evaluates the development of convergence of the algorithms tested with increasing the number of cases on the basis of training:

- PROCEDURE 2
    1. select the training bases $\mathbb{D}_1 ... \mathbb{D}_T$ and the algorithms $\mathcal{A}_1 ... \mathcal{A}_L$;
    2. for each $\mathbb{D}_t$ run PROCEDURE 1 on each $\mathcal{A}_l$; for each pair ($\mathbb{D}_t$, $\mathcal{A}_l$) this step results in $\Delta(\mathcal{A}_l)_t$;
    3. obtain statistics that encourage the trend of convergence of measures (such as regression, moving average charts);
    4. attach that information to the selection process of the algorithm.

4. An example of application

This section presents an application example the protocol described in the previous section. The application involves the use of a model, still in development, to issue diagnostics in periodontology. Figure 2 shows the topology of the Bayesian network used in the experiment.

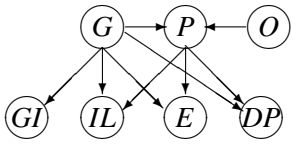

Figura 2. Bayesian network of the example.

In this network nodes G e P indicate diagnosis of gingivitis and periodontitis, respectively. Nodes E, IL e DP denote the presence of sites of exudate, insertion loss than 3mm and points with depth survey of more than 5mm.The example assumes that these three symptoms are influenced by variables G and P. The knot GI conditioned by G symbolizes the gingival index of Löe-Silness [Loe and Silness 1963],in

more detail the preposition that the *gingival index is greater than 1*. The auxiliary variable *O* represents the other possible causes for the occurrence of periodontitis [Ramachandran and Mooney 1998].

The example assumes that the successive training bases $\mathbb{D}_1$, $\mathbb{D}_2$, $\mathbb{D}_3$ e $\mathbb{D}_4$ are complete and contains 200, 500, 750 and 1000 cases respectively. The objective of the experiment is to compare the convergence three implementations of the *Voting EM* algorithm. The implementation $\mathtt{A}_1$ specifies $\eta = 0,01$, the implementation $\mathtt{A}_2$ takes $\eta = 0,05$ and in the implementation $\mathtt{A}_3$ has $\eta = 0,3$. The data base training were simulated from subjective probability model specified and initial values of TPCs were assumed to come from regular distributions.

The results obtained with the experimental protocol of Section 3 are listed in Table 1.This table shows that $\Delta(\mathtt{A}_*)_*$ grows with the increase in the number of cases processed during training. This is an indication that the selected basis with $m \leq 1000$, no implementations of *Voting EM* algorithm converged adequately for the empirical distribution. This is evident in Figure 3 where it is possible be noted that the best results were obtained for $\eta = 0,01$.

Tabela 1. Results of Experiment.

Algorithm	$\mathtt{A}_1$	$\mathtt{A}_2$	$\mathtt{A}_3$
$\Delta(\mathtt{A}_l)_1$	0,81	1,78	6,96
$\Delta(\mathtt{A}_l)_2$	0,99	2,97	16,99
$\Delta(\mathtt{A}_l)_3$	1,15	3,39	19,78
$\Delta(\mathtt{A}_l)_4$	4,33	4,33	26,2

E_* is the prediction error of a model and *epsilon* an arbitrary value. The concept of *sample complexity*[3] is defined as the number of cases needed to generate, with a probability less than $\delta \leq \frac{1}{2}$, a model in which $E_* \leq \varepsilon$ [Russell and Norvig 1995]. Following [Dasgupta 1997] the complexity the sample for training the model used in the example is hit by $m \geq 4828$; assuming that (a) the algorithms used are non iterative (b) the database is complete, (c) $\delta = 0,05$, (d) $\varepsilon = 12$ and (e) the error is measured by the log-likelihood of the model in relation to data.

[3]This paper considers the learning scheme probably approximately correct models as described in the context computational learning theory.

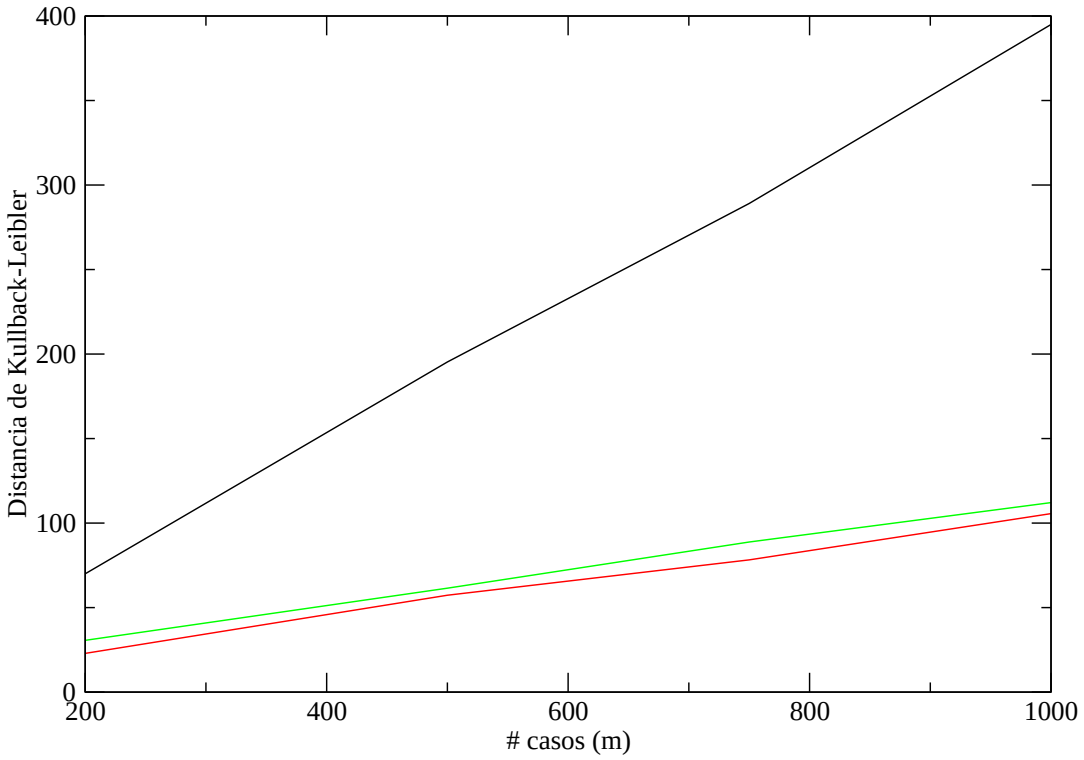

Figura 3. Results obtained for different sizes of base cases. Legend: A_1-dashed line; A_2-traço and point; A_3 full line.

Thus, when evaluating the results the experiment is important to consider that the *Voting EM* algorithm applies a myopic heuristic to perform the updates of the model (update basically depends on the update of the weight associated with the case being processed), then is reasonable to assume that it is more likely to converge to points that are not necessarily the minimum (local or global) regarding the extent of divergence used. This may partly explain the observed behavior.

5. Final Thoughts

This paper presented an experimental protocol to evaluate the convergence of iterative algorithms for training parameters of Bayesians networks.The importance of this work comes from the fact that the Bayesian networks comprise a formalism for uncertain knowledge representation and reasoning under uncertainty that has been used in many applications. This has motivated the implementation of algorithms for learning Bayesian networks integrated environments for many mining of data. However, this is not the case for iterative algorithms.

The protocol described here specifies the use of Kullback-Leible distance between the model represented by the Bayesian network and distribution the training data. This distance is calculated for different sizes of databases. After this processing, the results obtained by different algorithms can be compared using measures that assess the trend in a series of tests.

In future works we intend to extend the results presented here for the following cases

- imprecise probabilistic models [Levi 1980];
- models whose distributions are not regular at first [Russell and Norvig 1995];
- test the proposed protocol with other iterative algorithms;
- experiments with incomplete training bases.

Referências

Abbell, P., Koller, D., and Ng, A. (2006). Learning factor graphs in polynomial time and sample complexity. *Journal of Machine Learning Research*, 7:1743–1788.

Bauer, E., Koller, D., and Singer, Y. Update rules for parameter estimation of bayesian networks. In *11th International Joint Conference on Artificial Intelligence*, pages 2–13.

Cohen, I., Bronstein, A., and Cozman, F. (2001). Adaptive online learning of bayesian network parameters. Technical report, Hewlett Packard.

Dasgupta, S. (1997). The sample complexity of learning fixed-structure bayesian networks. *Machine Learning*, 29(2-3):165–180.

Duda, H. and Hart, J. (1987). *Pattern Recognition*. John Willey and Sons, New York.

Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163.

Heckerman, D. (1995). A tutorial on learning with bayesian networks.

Krause, J. P. (1998). Learning probabilistic networks. Technical report, http://citeseer.nj.nec.com/krause98learning.html. http://citeseer.nj.nec.com/krause98learning.html.

Levi, I. (1980). *The Enterprise of Knowledge*. MIT Press, Cambridge.

Loe, H. and Silness, J. (1963). Periodontal disease in pregnancyi.prevalence and severity. *Acta Odontol. Scand.*, 21(6):533–551.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill, New York.

Neapolitan, R. E. (1990). *Probabilistic Reasoning in Expert Systems*. Prentice Hall, Englewood Cliffs.

Parsaye, K. and Chignell, M. (1988). *Expert systems for experts*. John Wiley and Sons, New York.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco.

Provan, G. M. (1994). Tradeoffs on knowledge-based construcion of probabilistic models. *IEEE Transactions on Systems, Man and Cybernetics*, 24(11):1580–1592.

Ramachandran, S. and Mooney, R. (1998). Theory refinement for bayesian networks with hidden variables. In *In Machine Learning: Proceedingsof the International Conference*, pages 454–462. Morgan Kaufmann.

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A modern approach*. Prentice Hall, Upper Saddle River.

Zhang, N. L. and Poole, D. (1996). Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328.