EFFICIENT LEARNING OF DYNAMIC BAYESIAN NETWORKS FROM TIMED DATA

Ahmad Ahdab and Marc Le Goc

LSIS, UMR CNRS 6168, Université Paul Cézanne, Domaine Universitaire St Jérôme, 13397 Marseille cedex 20, France

Keywords: Machine Learning, Bayesian network, Stochastic representation, Data mining, Knowledge discovery.

Abstract: This paper addresses the problem of learning a Dynamic Bayesian network from timed data without prior knowledge to the system. One of the main problems of learning a Dynamic Bayesian network is building and orienting the edges of the network avoiding loops. The problem is more difficult when data are timed. This paper proposes an algorithm based on an adequate representation of a set of sequences of timed data and uses an information based measure of the relations between two edges. This algorithm is a part of the Timed Observation Mining for Learning (TOM4L) process that is based on the Theory of the Timed Observations. The paper illustrates the algorithm with an application on the Apache system of the Arcelor-Mittal Steel Group, a real world knowledge based system that diagnoses a galvanization bath.

1 INTRODUCTION

This paper describes the BJM4BN algorithm (BJ-Measure for Bayesian Networks) that learns a Dynamic Bayesian networks from timed data without prior knowledge to the process that generates the timed data. Most of the contributions to learn a bayesian network are based on un-timed data. The main difficulties to orient the edges of the resulting graph and building the conditional probability tables. These problems are more difficult when data are timed.

The BJM4BN algorithm proposes an efficient solution to solve these problems when data are timed. The solution is based on a particular representation of the timed data called the Stochastic Representation. This representation is the basis of the theory of Timed Observations (Le Goc, 2006). This theory defines a learning process called Timed Observation Mining for Learning (TOM4L) (Le Goc, 2005). The TOM4L process aims at discovering temporal knowledge about a set of time functions $x_i(t)$ considered as a dynamic system $X(t)=\{x_i(t)\}$ called a process. To this aim, the theory of Timed Observations defines an entropic measures called the the BJ-Measure of (Benayadi, 2008) that evaluates the flow of information between two nodes in a graph and provides so an efficient mean to orient the edges.

The next section presents a very short state of the art about learning Dynamic Bayesian networks (DBN). Section 3 introduces the basis of the Stochastic Representation of TOM4L and the BJ-Measure. Section 4 describes the BJM4BN algorithm and section 5 shows a real life application of the algorithm. Section 6 concludes the paper.

2 RELATED WORKS

A BN is a couple $<G, \theta>$ where $G$ denotes a Direct Acyclic Graph in which the nodes represent the variables and the edges represent the dependencies between the variables (Pearl, 1988), and $\theta$ is the Conditional Probabilities Tables (CP Tables) defining the conditional probability between the values of a variable given the values of the upstream variables of $G$. BN learning algorithms aims at discovering the couple $<G, \theta>$ from a given data base. BN learning algorithms fall into two main categories: “search and scoring” and “dependency analysis” algorithms. The “search and scoring” learning algorithms can be used when the knowledge of the edge orientation between the variables of the system is given (Cooper, 1992), (Heckerman, 1997). To avoid this problem, dependency analysis algorithms uses conditional independence tests (Cheng, 1997), (Cheeseman, 1995), (Friedman 1998), (Meyrs et al, 1999). But the number of test exponentially increases the computation time.
(Chickering, 1994).

For example, Cheng’s algorithm (Cheng, 1997) for learning BN is based on the d-separation concept of (Pearl, 1988) to infer the structure G of the Bayesian Network, and the mutual information I(X, Y) (eq. 1) to detect conditional independence relations.

$$I(X, Y) = \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$  \hspace{1cm} (1)

The mutual information I(X, Y) is used to evaluate all the potential pairs of variables (X, Y) and to producing a list L sorted in descending order: pairs of higher mutual information are supposed to be more related than those having low mutual information values. The List L is then pruned given an arbitrary parameter ε: each pair (X, Y) so that I(X, Y) < ε is eliminated of L. In real world applications, list L should be as small as possible using the ε parameter. The two main limitations such approaches is the need of defining the ε parameter and the exponential amount of Conditional Independence tests to orient the edges of the graph.

3 TOM4L FRAMEWORK

The BJM4BN algorithm provides a solution to these two problems that is based on the Stochastic Representation of the TOM4L framework (Le Goc, 2005), (Le Goc, 2006), (Le Goc, 2008).

In this framework, a message timed at $t_k$ contained in a database is an occurrence $C(k)$ of an observation class $C = \{ (x_i, \delta_i) \}$ which is an arbitrary set of couples $(x_i, \delta_i)$ where $\delta_i$ is one of the discrete value of a variable $x_i$. An observation class is often a singleton because in that case, two classes $C = \{ (x_i, \delta_i) \}$ and $C' = \{ (x_j, \delta_j) \}$ are only linked with the variables $x_i$ and $x_j$ when the constants $\delta_i$ and $\delta_j$ are independent (Le Goc, 2006).

The Stochastic Representation of a sequence $\omega = (\ldots, C(k), \ldots)$ of $m$ occurrences $C(k)$ defining a set $C = \{ C \}$ of $n$ timed observations is a set of matrix (Le Goc, 2005), (Bouché, 2005) from which the BJ-Measure is computed (Benayadi, 2008). This measure is based on the Kullback-Leibler distance $D(P(Y|Y=C)||P(Y))$ that evaluates the relation between the distribution of the conditional probability of $Y$ knowing that $X = C$ and the prior probability distribution of $Y$. The BJ-measure decomposes the Kullback-Leibler distance in two terms around the independence point $P(Y|Y=C)=P(Y)$ (i.e. $D(P(Y|Y=C)||P(Y))=0$).

The BJL-measure $BJL(C \rightarrow C')$ of binary relation $r(C \rightarrow C')$ is the right part of the Kullback-Leibler distance $D(P(Y|Y=C)||P(Y))$ so that:

- $P(Y=C|X=C)<P(Y=C) \Rightarrow BJL(C, C')=0$
- $P(Y=C|X=C)\geq P(Y=C) \Rightarrow BJL(C \rightarrow C')=D(P(Y|Y=C)||P(Y))$

The $BJL(C \rightarrow C')$ is not null when the observation $C(k)$ provides some information about the observation $C'(k)$. Symmetrically, when $BJL(C \rightarrow C')<0$, the observation $C(k)$ provides some information about $C'(k)$. The BJL-measure $BJL(C \rightarrow C')$ of a binary relation $r(C \rightarrow C')$ is then the left part of the Kullback-Leibler distance:

- $P(Y=C'|X=C)<P(Y=C') \Rightarrow BJL(C' \rightarrow C')=D(P(Y|Y=C')||P(Y))$
- $P(Y=C'|X=C)\geq P(Y=C') \Rightarrow BJL(C \rightarrow C')=0$

Consequently:

$$D(P(Y|Y=C)||P(Y))=BJL(C \rightarrow C')+BJL(C' \rightarrow C')$$  \hspace{1cm} (2)

Similarly, the BJW-measure evaluates the information distribution between the predecessors $(C(k))$ or $-C(k)$ of an observation $C(k+1)$ at time $t_{k+1}$:

$$D(P(Y|Y=C)||P(Y))=BJW(C \rightarrow C)+BJW(C' \rightarrow C')$$  \hspace{1cm} (3)

Because $(P(C|C')=P(C')=P(C'|C))$, the two measures are null at the same independence point and can be combined in a single measure called the BJW-measure which is the norm vector of $BJL(C \rightarrow C')$ and $BJW(C \rightarrow C')$:

$$M(C \rightarrow C') = \sqrt{BJL(C, C')^2 + BJW(C, C')^2}$$  \hspace{1cm} (4)

The BJ-Measure is no more justifiable when the $\theta_{ij}$ is greater of 4 or less than $\frac{1}{4}$ (Benayadi, 2008). This property is called the $\theta$ property.

When a BJ-measure $M(C \rightarrow C')$ is positive, the timed observation distribution of the $C'$ class bring information about the timed observation distribution of the $C$ class. So, considering the positive values only, the BJ-measure $M(C \rightarrow C')$ satisfies the three following properties:

1. Dissymmetry: $M(C \rightarrow C')=M(C' \rightarrow C)$ (generally)
2. Positivity: $\forall C$, $C'$, $M(C \rightarrow C') \geq 0$
3. Independence: $M(C \rightarrow C')=0$ if and only if $C$ and $C'$ are independant (i.e. $P(C|C')=P(C)$)
4. Triangular inequality: $M(C \rightarrow C)<M(C \rightarrow C') + M(C' \rightarrow C)$

This latter property can be used to reason with the BJ-measure to deduce the structure of a dynamic Bayesian network.
4 LEARNING PRINCIPLES

Let us consider a set $R=\{\ldots, r(C\rightarrow C'), \ldots\}$ of $n$ binary relations. The operation that remove a binary relation $r(C\rightarrow C')$ from the set $R$ is denoted $\text{Remove}(r(C\rightarrow C'))$: $R \leftarrow R - \{r(C\rightarrow C')\}$.

The positivity property leads to remove the $r(C\rightarrow C')$ relations having a negative value of the BJ-measure:

- Rule 1 ("Positivity rule"):
  $\forall r(C\rightarrow C') \in R, M(C\rightarrow C') \leq 0 \Rightarrow \text{Remove}(r(C\rightarrow C'))$

  The dissymmetry property allows deducing the orientation of a hypothetical relation between two timed observation classes $C^i$ and $C^j$:

- Rule 2 (Orientation rule):
  $\forall r(C\rightarrow C'), r(C\rightarrow C') \in R, M(C\rightarrow C') \cdot M(C'\rightarrow C) \Rightarrow \text{Remove}(r(C\rightarrow C'))$

![Figure 1: Loops.](image)

Now, let us consider a set $R=\{r(C\rightarrow C'^i), r(C\rightarrow C'^{i+1}), \ldots, r(C\rightarrow C'^{i+n})\}$, of $n+2$ binary relations defining a loop (Figure 2) where:

- $\forall r(C\rightarrow C') \in R, M(C\rightarrow C') > 0$

The problem of the set $R$ is that computing the distribution of a class $C^i$ requires knowing its distribution: loops must then be avoided. In other words, a relation $r(C\rightarrow C')$ must be removed from $R$ to break the loop. To solve this problem, the idea is to used the monotonous property of the BJ-measure: finding two of class $C^i$ and $C^j$ so that the BJ-measure of the relation $r(C\rightarrow C')$ is the lowest of the paths. To use this idea, all the paths leading to a particular $C^i$ class must be find in $R$. Let us suppose that $R$ contains $n$ paths $R_1 \subseteq R$, $R_2 \subseteq R$, \ldots, $R_n \subseteq R$ leading to the $C^i$ class (i.e. each $R_i$ is of the form $R_i=\{r(C\rightarrow C'^k), r(C'^k\rightarrow C'^{k+1}), \ldots, r(C'^{k+n}\rightarrow C', r(C'^{k+n}\rightarrow C', r(C^i\rightarrow C', r(C^i\rightarrow C')\}}$). The algorithm must find the $r(C\rightarrow C')$ relation with the lowest BJ-measure to remove it in $R$ ("Transitivity rule"):

- Rule 5 (Transitivity rule):
  $\forall r(C\rightarrow C') \in R, M(C\rightarrow C') > 0 \Rightarrow \text{Remove}(r(C\rightarrow C'))$

![Figure 2: Multiple Paths.](image)

These five rules are necessary (but no sufficient) to design an algorithm, but its efficiency depends mainly of the number of relation in the initial set $R$. The TOM4L framework provides the mathematical tools to remove the relations that can not play a significant role in the building of a naive Bayesian Network. Given the set $R=\{\ldots, r(C\rightarrow C'), \ldots\}$ of $n$ binary relations that can be build from a sequence $\omega$ of timed observation $C(k)$ defining a set $C=\{C^i\}$ of $N(C)$ classes $C^i$. The size of the Stochastic Representation matrix of the TOM4L framework is then $N(C) \cdot N(C)=N(C)^2$. This provides two ways to eliminate a relation $r(C\rightarrow C')$ having no interest for building a naive Bayesian Network:

- Test 1: $P(C|C') \cdot P(C', C) \leq 1/N(C)^2$
  $\Rightarrow \text{Remove}(r(C\rightarrow C'))$

  This first test compares $b_{ij}=P(C'|C') \cdot P(C', C)$ with the hazard of having an occurrence $C(k)$ of the $C^i$ class at time $t_i$ knowing that there is an occurrence $C(k+1)$ of the $C^i$ class occurring at time $t_{k+1}$. Because $\omega$ defines $N(C)$ classes, the $a \text{ priori}$ probability of having an occurrence $C(k)$ of a $C^i$ class followed by an occurrence $C(k+1)$ of the $C^i$ class is $P(C|C')=1/N(C)$ and the probability of reading a couple $(C(k), C(k+1))$ in $\omega$ is $P(C, C')=1/N(C)^2$. So each $b_{ij}$ value can be compared with the “absolute” hazard $1/(N(C) \cdot N(C)^2)$.

- Test 2: $P(C|C') \cdot P(C', C) \leq 1/(N(C) \cdot P(C) \cdot P(C'))$
  $\Rightarrow \text{Remove}(r(C\rightarrow C'))$

  This second test defines the hazard when supposing that $C^i$ and $C'^j$ classes are independent. In that case, the probability $P((C(k), C(k+1))$ in $\omega$ of having a couple $(C(k), C(k+1))$ in $\omega$ is $P(C) \cdot P(C')$ and having an occurrence $C(k)$ of a $C^i$ class, the hazard is to read any occurrence $C(k+1)$ of the $C^j$
class: $P(C|C)=1/N(C)$. So each $b_{ij}$ value can be compared with the “relative” hazard $1/(N(C)P(C)P(C))$.

The $θ$ property of the BJ-measure complete these two tests to eliminate the relation having no meaning according to the BJ-measure:

- Test 3: $θ_j>4 \Rightarrow Remove(r(C→C_i))$
- Within the TOM4L framework, these tree tests are implemented in the $F0/1=[f_j]$ matrix:
  - $(b_{ij}>1/N(C)) \land (b_{ij} \land 1/(N(C)P(C)P(C)) \land 1/4 \leq θ_j \leq 4) \Rightarrow f_j = 1$

So this lead to the rule number 6:

- Rule 6: $∀r(C→C_i)∈R, f_j = 0 \Rightarrow Remove(r(C→C_i))$

These six rules are used by the BJM4BN algorithm to build a naïve Bayesian Network.

5 THE BJM4BN ALGORITHM

The BJM4BN algorithm takes as inputs a sequence $ω$ of timed observation $C(κ)$ defining a set $\{C^i\}$ of $N(ω)$ classes $C^i$ and an output $C^j$ class that is the class for which the DBN is computed. It produces a set $G=\{r(C→C_i), ..., r(C→C_i)\}$ of $n$ binary relations that form the structure of a naïve Bayesian Network $(G, Θ)$.

The “BJM4BN” algorithm contains 5 stages. The first stage computes the Stochastic Representation of $Θ$ to produce the initial $M=[m_{ij}]$ matrix containing the BJ-measure values of the $N(ω)^2$ binary relations $r(C→C_i)$ defined by $Θ$ (line 1).

// Stage 1
1. Compute the $M=[m_{ij}]$ matrix
2. $∀i \neq 0, N(ω(1)), ∀j \neq 0, N(ω(1)), f_i = 0$
3. $∀i \neq 0, N(ω(1)), ∀j \neq 0, N(ω(1))$
   $∀(b_{ij}>1/N(ω(1)) \land (b_{ij} \land 1/(N(ω(1))P(C)P(C)) \land 1/4 \leq θ_j \leq 4) \Rightarrow f_j = 1$
4. $M=FO/1$
5. $∀i \neq 0, N(ω(1)), ∀j \neq 0, N(ω(1))$
   $m_{ij} = 0 \Rightarrow m_{ij} = 0$ // rule 1
6. $m_{ij} = 0 \Rightarrow m_{ij} = 0$ // rule 2

Next, the $F0/1=[f_{ij}]$ matrix is computed using test 4 (line 3) so that the new values $m_{ij}→m_{ij}:f_{ij}$ of matrix $M=[m_{ij}]$ with rule 6 (line 4). Finally, the $M$ matrix is normalized using rules 1 (line 5.1) and 4 (line 5.2).

// Stage 2
7. $L=[ϕ]\leftarrow[\{r(C→C^i), m_{ij}\}]$

Stage 2 computes the list $L$ from the normalized matrix $M$. The list $L=\{r(C→C_i), m_{ij}\}$ contains couples of the form $r(C→C_i), m_{ij}$ where $m_{ij}$ is the BJ-measure value of the associated $r(C→C_i)$ binary relation. This list will be used to find the relation $r(C→C_i)$ so that $m_{ij}$ is the minimal value of the BJ-measures contained in $L$.

// Stage 3
8. $C^i=C^i, G=\{ϕ\}$
9. $∀x(r(C→C^i))∈L \Rightarrow G=G+r(C→C^i)$
10. $Build(G, C^i)\leftarrow[\{r(C→C^i)\}] \Rightarrow G=G+r(C→C^i)$

Stage 3 builds recursively the initial $G$ graph from the $C_i$ class. This stage uses a recursive function called “Build(G, $C^i$)” where $C_i$ is the class the graph of which is to be build.

// Stage 4
11. $R=ϕ$
12. $∀R∈G, R=\{r(C→C^i), r(C→C^i), ..., r(C→C^i), r(C→C^i)\} \Rightarrow R=R+R(R_i)$
13. $∀R∈G, ∀x(r(C→C^i))∈R_i \Rightarrow R=R+R(R_i)$

// Stage 5
15. $R=ϕ$
16. $∀R∈G, R=\{r(C→C^i), r(C→C^i), ..., r(C→C^i), r(C→C^i)\} \Rightarrow R=R+R(R_i)$

Stage 4 finds and removes the loops in $G$ with Rule 3. This stage finds all the loops $R_i$ in $G$ of the form $R=\{r(C→C^i), r(C→C^i), ..., r(C→C^i), r(C→C^i)\}$ and put them in a set $R_i$ (line 12). Next, a new list $L_i$ is build containing all the relation $r(C→C^i)$ in $R_i$ with its associated $m_{ij}$ BJ-measure value (line 13). Every loops $R_i$ in $R$ are then removed using Rule 3 and the resulting $G$ graph is updated (line 14). At the end of this stage, $G$ contains no more loops. It is to note that the $L_i$ list being global (i.e. containing all the relations $r(C→C^i)$ participating in a loop), it is guaranty that the set of removed relation $r(C→C^i)$ is optimal: it is minimal and the removed relations are the smallest of the $G$ graph.

// Stage 5
18. $∀R∈G, ∀x(r(C→C^i))∈R_i, m_{ij} = Min(m_{ij}, L_i)$

229
Stage 5 removes the multiple paths in the $G$ graph with Rule 5. This stage proceeds as the stage 4, but the $R$ set contains only paths $R_i$ of the form $R = \{r(C \rightarrow C_{a},\ldots), r(C \rightarrow C_{a}), \ldots, r(C \rightarrow C_i)\}$ (line 16). At the end of this stage, $G$ contains no more multiple paths and it is guaranty that the set of removed relation $r(C \rightarrow C')$ is optimal.

Stage 6 computes the conditional probabilities tables for $G$ and finalizes the algorithm. The computing of the Conditional Probabilities Tables (CP Tables) is based on the numbering table $N^i[n_{ij}]$ of the Stochastic Representation of the $\omega$ sequence:

- $P(Y=C' | X=C) + P(Y=C | X=C') = 1$.

The computing of the CP tables uses this property (for simplicity, $P(C' | C)$ is rewritten $P(y|x)$).

For a root node $C'$:

- $P(x) = \sum_{n_{ij}} / \sum_{n_{ij}}$

For a single relation $r(C' \rightarrow C)$:

- $P(y|x) = n_{xy} / \sum_{n_{ij}}$
- $P(y|x) = \sum_{n_{ji}} = \sum_{n_{ij}}$

For a set $R = \{r(C' \rightarrow C')\}$ of two relations converging to the same $C'$ class:

- $P(y|x,z) = \sum_{n_{ij}} / \sum_{n_{ij}}$
- $P(y|x,z) = \sum_{n_{ij}} / \sum_{n_{ij}}$
- $P(y|x,z) = \sum_{n_{ij}} / \sum_{n_{ij}}$
- $P(y|x,z) = \sum_{n_{ij}} / \sum_{n_{ij}}$

The next section illustrates the computation of the CP tables with a real world process.

6 REAL WORLD APPLICATION

The Apache system is a clone Satchem, the knowledge based systems that The Arcelor Group, one of the most important steel companies in the world, has developed to monitor and diagnose its production tools (Le Goc, 2004). Apache aims at controlling a zinc bath, a hot bath containing a liquid mixture of aluminum and zinc continuously fed with aluminum and zinc ingots in which a hot steel strip is immerged. Apache monitors and diagnoses around 11 variables and is able to detect around 24 types of alarms. The analyzed sequence $\omega$ contains 687 events of 13 classes for 11 discrete variables. The counting matrix $N$ contains then 156 cells $n_{ij}$ (Table 1). The corresponding $M$ matrix is provided in Table 2, the $P/0/1$ matrix in Table 3 and the normalized $M$ matrix in Table 4. These matrices are computed in the first stage of the “BJM4BN” algorithm.

The stage 2 computes the $L$ list of Table 5. The node of interest being 1006, the initial $G$ graph resulting of stage 3 of the “BJM4BN” algorithm is given in Figure 4. This stage uses the normalized $M$ matrix and start with the 1006 column to add the relations $r(1001, 1006)$ and $r(1001, 1006)$ in the initial $G$ graph. Next, the Build($G$, 1006) function is executed to add the relation $r(1004, 1001)$ in $G$ before calling Build($G$, 1001) function and so on.

<table>
<thead>
<tr>
<th>Table 1: Counting matrix $N$ of $\omega$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega$</td>
</tr>
<tr>
<td>1004</td>
</tr>
<tr>
<td>1010</td>
</tr>
<tr>
<td>1015</td>
</tr>
<tr>
<td>1020</td>
</tr>
<tr>
<td>1025</td>
</tr>
</tbody>
</table>

The $G$ graph of Figure 3 having no loops, the stage 4 modify noting and stage 5 is executed with
this graph. Because each relation of the $G$ graph participates to the seven paths leading to the nodes 1006 and 1001, the $L_1$ list is equal to the $L$ list of Table 5. This table has been sorted so that the minimal value $m_2$ is at the end of the list. Its is then easy to see that the relation $(1020, 1031)$ is the first removed relation, before the relations $(1020, 1006)$ and $(1004, 1001)$. The elimination of these three relations is sufficient to build the final $G$ graph of Figure 4.

![Figure 3: Initial $G$ graph.](image1)

![Figure 4: Final $G$ graph.](image2)

The CPT tables are computed using the $N$ matrix (table 1). This matrix provides all the information’s to compute the probabilities of the root nodes of figure 4. A handmade graph of the experts of the Arcelor Group in 2003 can be find in (Bouché, 2005) and (Le Goc, 2005). The $G$ graph is all contained in the expert’s graph. But the expert’s graph does not contain the 1024 class: corresponding to an operator query for a chemical analysis, this class has been removed by experts.

7 CONCLUSIONS

This paper shows that the “BJT4BN” algorithm is efficient both in terms of pertinence, simplicity and speed. These properties come from the $BJ$-measure that provides an operational way to orient the edges of Bayesian Network without the exponential CI Tests of Cheng’s method. It is then an advantage of using the time of the data to learn a dynamic Bayesian network. Our current works are concerned with the combination of the Timed Data Mining techniques of the TOM4L framework with the “BJT4BN” algorithm to define a global validation of the TOM4L learning process.

REFERENCES


