Learning CB-decomposable Multi-dimensional Bayesian Network Classifiers

Hanen Borchani, Concha Bielza and Pedro Larrañaga
Departamento de Inteligencia Artificial, Facultad de Informática
Universidad Politécnica de Madrid, Boadilla del Monte, 28660, Madrid, Spain.
hanen.borchani@upm.es, {mcbielza, pedro.larranaga}@fi.upm.es

Abstract

Multi-dimensional Bayesian network classifiers (MBCs) have been recently introduced to deal with multi-dimensional classification problems where instances are assigned to multiple classes. MBCs have a restricted topology partitioning the set of class and feature variables into three different subgraphs: class subgraph, feature subgraph and bridge subgraph. In this paper, we propose a novel learning algorithm for class-bridge (CB) decomposable MBCs into maximal connected components. Basically, based on a wrapper greedy forward selection approach, the algorithm firstly learns the bridge and feature subgraphs. Then, while the number of components is greater than one and there is an accuracy improvement, it iteratively and sequentially merges together the components, and updates the bridge and feature subgraphs. By learning CB-decomposable MBCs, the computations of MPE are alleviated comparing to general MBCs. Experimental comparison with state-of-the-art algorithms are carried out using synthetic and real-world data sets. The obtained results show the merits of our proposed algorithm.

1 Introduction

Multi-dimensional classification (van der Gaag and de Waal, 2006) is an extension of the classical one-dimensional classification where each instance given by a vector of m features $\mathbf{x} = (x_1, ..., x_m)$ is associated, with not only a single class value, but with a set of d class values $\mathbf{c} = (c_1, ..., c_d)$. Multi-dimensional classification has been motivated by several application domains. For instance, in text categorization, a text document may be assigned to more than one topic; in scene classification, each semantic scene may be assigned to several classes, such as beach, sunset and mountain; in medical diagnosis, a patient may be suffering from multiple diseases, etc.

In recent years, the concept of multi-dimensionality has been introduced in Bayesian network classifiers (van der Gaag and de Waal, 2006; de Waal and van der Gaag, 2007; Rodríguez and Lozano, 2008; Bielza et al., 2010). In these probabilistic graphical models, known as multi-dimensional Bayesian network classifiers (MBCs), the graphical structure partitions the set of class and feature variables into three different subgraphs: class subgraph, feature subgraph and bridge subgraph, and the parameter set defines the conditional probability distribution of each variable given its parents.

One of the most challenging problems with MBC models involves the most probable explanation (MPE) computation, which is known to be NP-hard in general, and presents a significant complexity especially when the MBC has a large number of class variables.

In this paper, in order to alleviate the MPE computational burden, we consider the family of class-bridge decomposable multi-dimensional Bayesian network classifiers (CB-decomposable MBCs) introduced by Bielza et al. (2010). In fact, by decomposing class and bridge subgraphs of an MBC graphical structure into r maximal connected components, the maximization problem for MPE computation can

be transformed into r maximization problems operating in lower dimensional spaces. Moreover, using CB-decomposable MBCs may provide more insight about the domain and better interpretability of learned structures than large and complex MBCs which have no explicit representation for domain decomposability.

However, CB-decomposable MBCs merits have been only discussed and proved theoretically in (Bielza et al., 2010), and no learning approach nor an experimental study have been presented to empirically demonstrate the usefulness of this new family of MBCs.

In order to tackle these shortcomings, we propose in the present work a novel algorithm for learning CB-decomposable MBCs based on a wrapper greedy forward selection approach. Broadly speaking, in a first phase our algorithm learns a CB-decomposable MBC with a number of maximal connected components equal to the number of class variables. This is carried out by learning a selective naive Bayes (Langley and Sage, 1994) for each class variable C, then, removing their possible common children to have an initial bridge subgraph and the corresponding CB-decomposable MBC. In a second phase, a feature subgraph defining dependence relationships between the set of feature variables is learned. Finally, in a third phase, while the number of maximal connected components is greater than one and there is an accuracy improvement, the algorithm iteratively and sequentially merges together the components, then updates the bridge and feature subgraphs.

The remainder of this paper is organized as follows. In Section 2 we review the definitions of MBCs and CB-decomposable MBCs. In Section 3 we describe our algorithm for learning CB-decomposable MBCs from data. In Section 4 we present experimental set up and results. Finally, we round off the paper with some conclusions in Section 5.

2 Multi-dimensional Bayesian Network Classifiers

A Bayesian network over a set of discrete random variables $\mathbf{U} = \{X_1, ..., X_n\}, n \geq 1$, is a pair $\mathcal{B} = (\mathcal{G}, \mathbf{\Theta})$. $\mathcal{G} = (V, A)$ is a directed acyclic graph (DAG) whose vertices V correspond to variables $\mathbf{U}$, and whose arcs A represent direct dependencies between the vertices. $\mathbf{\Theta}$ is a set of conditional probability distributions such that $\theta_{x_i|\mathbf{pa}(x_i)} = p(x_i \mid \mathbf{pa}(x_i))$ defines the conditional probability of each possible value x_i of X_i given a set value $\mathbf{pa}(x_i)$ of $\mathbf{Pa}(X_i)$, where $\mathbf{Pa}(X_i)$ denotes the set of parents of X_i in $\mathcal{G}$.

A Bayesian network $\mathcal{B}$ represents a joint probability distribution over $\mathbf{U}$ factorized according to structure $\mathcal{G}$ as follows:

$$p(X_1, ..., X_n) = \prod_{i=1}^{n} p(X_i \mid \mathbf{Pa}(X_i)) \cdot \quad (1)$$

Definition 1. A *multi-dimensional Bayesian network classifier* (MBC) is a Bayesian network $\mathcal{B} = (\mathcal{G}, \mathbf{\Theta})$ where the structure $\mathcal{G} = (V, A)$ has a restricted topology. The set of vertices V is partitioned into two sets: $V_C = \{C_1, ..., C_d\}, d \geq 1$, of class variables and $V_X = \{X_1, ..., X_m\}, m \geq 1$, of feature variables ($d + m = n$). Moreover, the set of arcs A is partitioned into three sets A_C, A_X and A_{CX}, such that:

- $A_C \subseteq V_C \times V_C$ is composed of the arcs between the class variables having a subgraph $\mathcal{G}_\mathcal{C} = (V_C, A_C)$ -*class subgraph*- of $\mathcal{G}$ induced by V_C.
- $A_X \subseteq V_X \times V_X$ is composed of the arcs between the feature variables having a subgraph $\mathcal{G}_\mathcal{X} = (V_X, A_X)$ -*feature subgraph*- of $\mathcal{G}$ induced by V_X.
- $A_{CX} \subseteq V_C \times V_X$ is composed of the arcs from the class variables to the feature variables having a subgraph $\mathcal{G}_{\mathcal{CX}} = (V, A_{CX})$ -*bridge subgraph*- of $\mathcal{G}$ connecting class and feature variables.

Classification with an MBC under a 0-1 loss function amounts to solving the most probable explanation (MPE) problem, i.e. for a given evidence $\mathbf{x} = (x_1, ..., x_m)$ we have to get:

$$\begin{aligned} \mathbf{c}^* &= (c_1^*, ..., c_d^*) \\ &= arg \max_{c_1,...,c_d} p(C_1 = c_1, ..., C_d = c_d \mid \mathbf{x}) \cdot \end{aligned} \quad (2)$$

Example 1. Figure 1 shows an example of an MBC structure $\mathcal{G} = \mathcal{G}_{\mathcal{C}} \cup \mathcal{G}_{\mathcal{X}} \cup \mathcal{G}_{\mathcal{CX}}$ where the set of class variables $V_C = \{C_1, C_2, C_3, C_4\}$ and the set of feature variables $V_X = \{X_1, X_2, X_3, X_4, X_5, X_6\}$. We have:

$$\begin{aligned}
&\max_{c_1,...,c_4} p(C_1 = c_1, ..., C_4 = c_4 \mid \mathbf{x}) \\
&\propto \max_{c_1,...,c_4} p(c_1)p(c_2 \mid c_1)p(c_3)p(c_4) \\
&\cdot\, p(x_1 \mid c_1, x_2)p(x_2 \mid c_1, c_2)p(x_3 \mid c_3) \\
&\cdot\, p(x_4 \mid c_3)p(x_5 \mid c_4, x_1, x_6)p(x_6 \mid c_3, x_3)
\end{aligned}$$

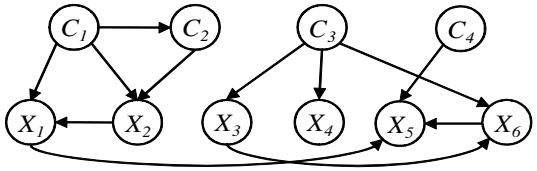

Figure 1: An example of an MBC structure.

Note that depending on the graphical structures of the class and feature subgraphs we can differentiate between several families of MBCs. Such families can be denoted as `class subgraph structure-feature subgraph structure` MBC where the possible structures of each subgraph may be: empty, tree, polytree, or DAG. MBC families used in the literature include `tree-tree` MBC (van der Gaag and de Waal, 2006), `polytree-polytree` MBC (de Waal and van der Gaag, 2007), `DAG-empty` MBC (Qazi et al., 2007), and `DAG-DAG` MBC (Rodríguez and Lozano, 2008). In this paper, we do not consider any restrictions on the learned MBC structures, i.e. any possible structure type is allowed for either class or feature subgraphs.

Definition 2. A *class-bridge decomposable multi-dimensional Bayesian network classifier* (CB-decomposable MBC) is an MBC $\mathcal{B} = (\mathcal{G}, \mathbf{\Theta})$ where the class subgraph $\mathcal{G}_{\mathcal{C}}$ and bridge subgraph $\mathcal{G}_{\mathcal{CX}}$ are decomposed into r maximal connected components, such that

1. $\mathcal{G}_C \cup \mathcal{G}_{(CX)} = \bigcup_{i=1}^{r} (\mathcal{G}_{C_i} \cup \mathcal{G}_{(CX)_i})$, where $\mathcal{G}_{C_i} \cup \mathcal{G}_{(CX)_i}$, with $i = 1, ..., r$, are its r *maximal connected components*, and
2. $Ch(V_{C_i}) \cap Ch(V_{C_j}) = \emptyset$, with $i, j = 1, ..., r$ and $i \neq j$, where $Ch(V_{C_i})$ denotes the children of all the variables in V_{C_i}, the subset of class variables in $\mathcal{G}_{C_i}$ (*non-shared children property*).

Bielza et al. (2010) proved that the MPE computation can be alleviated thanks to MBC class-bridge decomposability. In fact, maximizing over the set of all class variables amounts to maximizing over each class variable subset of the identified maximal connected components, i.e. maximizing over lower dimensional subspaces than originally.

Theorem 1. *Given a CB-decomposable MBC where $\mathcal{I}_i$ represents the sample space associated with V_{C_i}, then*

$$\begin{aligned}
&\max_{c_1,...,c_d} p(C_1 = c_1, ..., C_d = c_d \mid \boldsymbol{x}) \\
&\propto \prod_{i=1}^{r} \max_{\boldsymbol{c}^{\downarrow V_{C_i}} \in \mathcal{I}_i} \Bigg(\prod_{C \in V_{C_i}} p(c \mid \boldsymbol{pa}(c)) \\
&\qquad \cdot \prod_{X \in Ch(V_{C_i})} p(x \mid \boldsymbol{pa}_{V_C}(x), \boldsymbol{pa}_{V_X}(x)) \Bigg), \qquad (3)
\end{aligned}$$

where $\boldsymbol{c}^{\downarrow V_{C_i}}$ represents the projection of vector $\boldsymbol{c}$ to the coordinates found in V_{C_i}. $\boldsymbol{Pa}_{V_C}(X)$ and $\boldsymbol{Pa}_{V_X}(X)$ denote, respectively, the class parents and feature parents of X in $\mathcal{G}$. Obviously, for any class variable C, we have $\boldsymbol{Pa}_{V_X}(C) = \emptyset$ and $\boldsymbol{Pa}_{V_C}(C) = \boldsymbol{Pa}(C)$.

Given $\mathbf{x}$, each expression to be maximized in Equation (3) will be denoted as $\phi_i^{\mathbf{x}}(\mathbf{c}^{\downarrow V_{C_i}})$, $i = 1, ..., r$, i.e.

$$\begin{aligned}
\phi_i^{\mathbf{x}}(\mathbf{c}^{\downarrow V_{C_i}}) &= \prod_{C \in V_{C_i}} p(c \mid \mathbf{pa}(c)) \\
&\cdot \prod_{X \in Ch(V_{C_i})} p(x \mid \mathbf{pa}_{V_C}(x), \mathbf{pa}_{V_X}(x)) \cdot \qquad (4)
\end{aligned}$$

It holds that $\phi_i^{\mathbf{x}}(\mathbf{c}^{\downarrow V_{C_i}}) \propto p(\mathbf{C}^{\downarrow V_{C_i}} = \mathbf{c}^{\downarrow V_{C_i}} \mid \mathbf{x})$.

Example 2. Let us reconsider the MBC shown in Figure 1. It is a CB-decomposable MBC with $r = 3$. Its three maximal connected components are depicted in Figure 2. The first one is $\mathcal{G}_{C_1} \cup \mathcal{G}_{(CX)_1}$ with $V_{C_1} = \{C_1, C_2\}$ and

$Ch(V_{C_1}) = \{X_1, X_2\}$, the second is $\mathcal{G}_{C_2} \cup \mathcal{G}_{(CX)_2}$ with $V_{C_2} = \{C_3\}$ and $Ch(V_{C_2}) = \{X_3, X_4, X_6\}$, and the third is $\mathcal{G}_{C_3} \cup \mathcal{G}_{(CX)_3}$ with $V_{C_3} = \{C_4\}$ and $Ch(V_{C_3}) = \{X_5\}$. Note that $Ch(V_{C_1}) \cap Ch(V_{C_2}) = Ch(V_{C_1}) \cap Ch(V_{C_3}) = Ch(V_{C_2}) \cap Ch(V_{C_3}) = \emptyset$, as required. As a maximization problem we get:

$$\begin{aligned}
&\max_{c_1,...,c_4} p(C_1 = c_1, ..., C_4 = c_4 \mid \mathbf{x}) \\
&= \max_{c_1,c_2} p(c_1)p(c_2 \mid c_1)p(x_1 \mid c_1, x_2)p(x_2 \mid c_1, c_2) \\
&\quad \cdot \max_{c_3} p(c_3)p(x_3 \mid c_3)p(x_4 \mid c_3)p(x_6 \mid c_3, x_3) \\
&\quad \cdot \max_{c_4} p(c_4)p(x_5 \mid c_4, x_1, x_6) \\
&= \max_{c_1,c_2} \phi_1^{\mathbf{X}}(c_1, c_2) \cdot \max_{c_3} \phi_2^{\mathbf{X}}(c_3) \cdot \max_{c_4} \phi_3^{\mathbf{X}}(c_4) \cdot
\end{aligned}$$

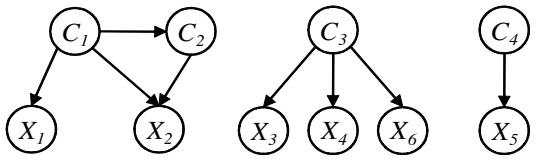

Figure 2: The three maximal connected components of the MBC example of Figure 1.

3 Learning CB-decomposable MBCs from Data

We describe in this section our proposed algorithm for learning CB-decomposable MBCs from data based on a wrapper greedy forward selection approach. Let $\mathcal{D}$ be a data set of N observations containing a value assignment for each variable $X_1, ..., X_m, C_1, ..., C_d$, i.e. $\mathcal{D} = \{(\mathbf{x}^{(1)}, \mathbf{c}^{(1)}), ..., (\mathbf{x}^{(N)}, \mathbf{c}^{(N)})\}$. As performance metrics, we use:

1. The *mean accuracy* over the d class variables:

$$Acc_m = \frac{1}{d} \sum_{i=1}^{d} \frac{1}{N} \sum_{l=1}^{N} \delta(c'_{li}, c_{li}), \qquad (5)$$

where $\delta(c'_{li}, c_{li}) = 1$ if $c'_{li} = c_{li}$, and 0 otherwise. Note that c'_{li} denotes the C_i class value outputted by the MBC for case l and c_{li} is its corresponding true value.

2. The *global accuracy* over the d-dimensional class variable:

$$Acc_g = \frac{1}{N} \sum_{l=1}^{N} \delta(\mathbf{c}'_l, \mathbf{c}_l), \qquad (6)$$

where $\delta(\mathbf{c}'_l, \mathbf{c}_l) = 1$ if $\mathbf{c}'_l = \mathbf{c}_l$, and 0 otherwise. That is, we call for a complete equality between all the components of the vector of predicted classes and the vector of real classes.

Our learning algorithm consists of three main phases, outlined by Algorithm 1, and detailed in what follows. Note that, in the different phases, the classifier accuracy is denoted Acc, which is equal to Acc_m or Acc_g depending on using the mean or the global accuracy.

3.1 Phase I: Learn bridge subgraph

Starting from an empty graphical structure, the first step in this phase is learning a selective naive Bayes (Langley and Sage, 1994) for each class variable C_i, $i = 1, ..., d$.

The d resulting selective naive Bayes models represent $r = d$ maximal connected components that may have common children. Thus, the next step is to check the non-shared children property in order to induce an initial CB-decomposable MBC. This is accomplished by removing, if necessary, all common children, based on two criteria, namely, the feature insertion rank and the accuracy.

Let $rank_j^i$ denotes the insertion rank of feature X_j in the selective naive Bayes NB_i for C_i, and $rank_j^k$ denotes the insertion rank of feature X_j in the selective naive Bayes NB_k for C_k. $rank_j^i < rank_j^k$ means that X_j is firstly selected by NB_i. Hence, in this case, X_j will be kept in NB_i and removed from NB_k. Otherwise, and in case that $rank_j^i = rank_j^k$, we proceed to compare the accuracies Acc^i and Acc^k, denoting respectively the accuracy of NB_i and NB_k when X_j was included in, then keep X_j in the NB presenting the highest accuracy and remove it from the other.

The result of this phase is a simple CB-decomposable MBC, denoted as $CB\text{-}MBC_r^b$, where only the bridge subgraph is defined and the class and feature subgraphs are still empty.

Algorithm 1

Input: $\mathcal{D}$, T
Output: $CB\text{-}MBC_r^{bfc}$
$\mathcal{G}_C = \emptyset$; $\mathcal{G}_{CX} = \emptyset$; $\mathcal{G}_X = \emptyset$; $r = d$.

[**Phase I: Learn bridge subgraph**]
Learn selective naive Bayes NB_i, $i = 1, ..., r$.
for Each NB_i, NB_k having a common feature X_j **do**
  if $rank_j^i < rank_j^k$ **then**
    Remove X_j from NB_k.
  else
    if $rank_j^i = rank_j^k$ **then**
      if $Acc^i > Acc^k$ **then**
        Remove X_j from NB_k.
      else
        Remove X_j from NB_i.
      end if
    end if
    Remove X_j from NB_i.
  end if
end for
Obtain $\mathcal{G}_{CX} = \bigcup_{i=1}^{r}(NB_i)$, that is, $CB\text{-}MBC_r^b$.

[**Phase II: Learn feature subgraph**]
for $TrialNumber = 1 : T$ **do**
  Add randomly one arc to $\mathcal{G}_X$.
  if No accuracy improvement **then**
    Discard the arc and do not consider it in subsequent iterations.
  end if
end for
Obtain $CB\text{-}MBC_r^{bf}$.

[**Phase III: Merge maximal connected components**]
$CB\text{-}MBC_r^{bfc} \longleftarrow CB\text{-}MBC_r^{bf}$.
Stop = False.
while $r > 1$ and *not Stop* **do**
  for Each class variables C_i, C_k pertaining to two different maximal connected components **do**
    Evaluate the arc insertion from C_i to C_k.
  end for
  Select the arc with the best accuracy Acc^{r-1}.
  if $Acc^{r-1} > Acc^r$ **then**
    Update $\mathcal{G}_C$.
    $Acc^r = Acc^{r-1}$.
    $CB\text{-}MBC_r^{bfc} \longleftarrow CB\text{-}MBC_{r-1}^{bfc}$.
    $r = r - 1$.
    while Accuracy improvement **do**
      Update $\mathcal{G}_{CX}$: add an arc from a class to a feature of the new merged component.
    end while
    while Accuracy improvement **do**
      Update $\mathcal{G}_X$: add an arc between feature variables.
    end while
  else
    Stop = True.
  end if
end while
return $CB\text{-}MBC_r^{bfc}$.

3.2 Phase II: Learn feature subgraph

This phase consists of learning the feature subgraph by introducing the dependence relationships between the feature variables. Since it may be impractical to consider all possible arc additions between the feature variables, especially if the number of features m is large, we will fix a parameter T as a maximum number of iterations.

In each iteration, an arc is selected at random between a pair of feature variables. If there is an accuracy improvement, the arc is added to $\mathcal{G}_X$, otherwise it is discarded and will not be considered in subsequent iterations. This phase ends when T is reached, and the induced MBC is denoted as $CB\text{-}MBC_r^{bf}$.

Note that, thanks to MBC decomposability, the classification accuracy associated with the arc addition in each iteration can be evaluated in a straightforward and local way. In fact, after adding an arc from X_i to X_j, only the term corresponding to variable X_j changes, that is, only the MPE computation of the maximal connected component to which X_j pertains, changes and needs to be reevaluated. The MPE computation over all remaining maximal connected components remains unchanged, which considerably reduces the computational burden.

3.3 Phase III: Merge maximal connected components

Taking as input the CB-decomposable MBC found in the previous phase, having r maximal connected components and a corresponding accuracy denoted Acc^r, the third phase consists of learning the class subgraph, which leads to merging the maximal connected components of the current CB-decomposable MBC, then updating the bridge and feature subgraphs.

As a first step, all possible arc additions between the class variables pertaining to different maximal connected components are evaluated. If there is an accuracy improvement, i.e. $Acc^{r-1} > Acc^r$, the subgraph $\mathcal{G}_C$ is updated by adding the arc improving the accuracy the most, and r is reduced to $r - 1$ maximal connected components.

Subsequently, a bridge update step is performed inside the new induced maximal connected component. Dependence relationships that may be added from class to feature variables of the corresponding component are greedily evaluated, and only when there is an accuracy improvement, the best one is added.

Note that, once again, the MBC decomposability plays a key role in alleviating the complexity of MPE computation since each possible arc addition between class variables or from class to feature variables is evaluated locally. Moreover, for bridge updating step, the MPE is only recomputed for the new merged component, which alleviates more the complexity.

The last step in this phase consists of updating the feature subgraph by inserting, one by one, additional arcs between feature variables while this improves the accuracy.

This phase iterates over these three steps, and terminates when no more component merging can improve the accuracy or until the condition $r = 1$ is reached. A CB-decomposable MBC denoted as $CB\text{-}MBC_r^{bfc}$ is returned.

4 Related Work

In this section, we briefly review the state-of-the-art on MBCs learning algorithms.

Van der Gaag and de Waal (2006) decompose the learning problem of `tree-tree` MBCs into two separate optimization problems: first learning the class subgraph using Chow and Liu's algorithm (1968), then, given a fixed bridge subgraph, learning the feature subgraph using also Chow and Liu's algorithm. The bridge subgraph is selected using a wrapper approach garanteeing a high classifier accuracy.

De Waal and van der Gaag (2007) present a theoretical approach for learning `polytree-polytree` MBCs where class and feature subgraphs are separately learnt based on Rebane and Pearl's algorithm (1989). Nevertheless, the induction of the bridge subgraph was not specified.

Moreover, Qazi et al. (2007) learn `DAG-empty` MBCs where the class subgraph is induced by standard Bayesian networks procedures, the bridge subgraph is learnt by adding dependence relationships form each class variable to a subset of selected features, and the feature subgraph is kept empty.

Rodríguez and Lozano (2008) use a multi-objective evolutionary approach to learn `DAG-DAG` MBCs. Each permitted MBC structure is coded as an individual with three substrings, one per subgraph. Based on different classification rules, joint and marginal, they define the objective functions as k-fold cross-validated estimators of each class classification error. The aim is to find non-dominated structures according to the objective functions.

More recently, Bielza et al. (2010) propose different learning algorithms, namely, pure filter (guided by the K2 algorithm), pure wrapper (guided by the classification accuracy) and hybrid algorithm (a combination of pure filter and pure wrapper), allowing any Bayesian network structure in the three MBC subgraphs.

Similarly as Bielza et al. (2010), we have no constraints about the subgraph structures of the generated MBCs. However, contrary to their leaning algorithms and contrary to other existing works, our proposal is to learn the new family of CB-decomposable MBCs instead of learning general MBCs.

5 Experiments

In order to evaluate our learning algorithm, we firstly perform experiments with a synthetic data set. We randomly generate an MBC, containing 6 class and 10 feature binary variables, decomposed into 3 maximal connected components. Then, we randomly sample a data set of size 1000 using the probabilistic logic sampling method (Henrion, 1988). We apply our algorithm denoted as `CB-MBC`, and other four algorithms, namely, `Tree-Tree` (van der Gaag and de Waal, 2006), `Polytree-Polytree` (de Waal and van der Gaag, 2007), `Pure Filter` (Bielza et al., 2010) and `Pure Wrapper` (Bielza et al., 2010), all starting from an empty structure.

We consider both the mean and the global accuracy to learn and then evaluate the performance of the classifiers. Furthermore, in order

to test the ability of the classifiers to recover the initial MBC structure, we compare each learned structure (LS) to the initial one (IS) using the following structural evaluation metrics:

- M1: percentage of arcs in LS that are present in IS, i.e. percentage of correctly-found arcs.
- M2: percentage of arcs in LS that are absent in IS, i.e. percentage of superfluous arcs.
- M3: percentage of arcs in IS that are oriented in an opposite direction in LS, i.e. percentage of badly-oriented arcs.
- M4: percentage of arcs in IS that are absent in LS, i.e. percentage of missing arcs.

Five-fold cross-validation experiments are run for each learning algorithm. Table 1 shows the average results over these runs.

Table 1: Experimental results over the synthetic data set.

Mean accuracy					
Classifier	Acc_m	M1	M2	M3	M4
`CB-MBC`	0.7182	26.66	37.55	11.88	61.44
`Tree-Tree`	0.7129	30.55	49.44	5.55	63.89
`Polytree-Polytree`	0.6330	30.10	15.10	6.21	63.66
`Pure Filter`	0.5351	7.55	14.66	8.88	83.55
`Pure Wrapper`	0.7098	22.22	42.88	17.77	60.00
Global accuracy					
Classifier	Acc_g	M1	M2	M3	M4
`CB-MBC`	0.2877	11.55	14.66	1.33	87.10
`Tree-Tree`	0.2838	25.00	53.88	8.33	66.66
`Polytree-Polytree`	0.2845	28.99	15.10	5.44	65.55
`Pure Filter`	0.2160	7.99	15.55	7.54	84.44
`Pure Wrapper`	0.2800	16.00	48.44	14.22	69.77

Our algorithm outperforms the state-of-the-art algorithms in terms of mean and global accuracy. For structural evaluation, `Tree-Tree` and `Polytree-Polytree` present the best percentages of correctly-found arcs (M1) while `Pure Filter` has the lowest one. `Tree-Tree` and `Pure Wrapper` induce the highest percentages of superfluous arcs (M2), and `Pure Wrapper` also induces a high percentage of badly-oriented arcs (M3) comparing to the rest of the algorithms. Moreover, we may observe that, with global accuracy, the learned structures are sparser, leading to more important percentages of missing arcs (M4) for all learning algorithms.

As additional experiments, we consider the real data set Emotions (Trohidis et al., 2008). It is about a multi-dimensional classification of music into emotions. It contains 72 music features for 593 songs categorized into one or more out of 6 classes of emotions: amazed-surprised, happy-pleased, relaxing-calm, quiet-still, sad-lonely, and angry-aggressive.

As previously, the accuracies of the considered learning algorithms are computed using 5-fold cross-validation. The results are summarized in Table 2. Note that, with this real data set, we do not have an initial MBC structure, so the structural evaluation is omitted in this set of experiments.

Table 2: Experimental results over Emotions data set.

Classifier	Acc_m	Acc_g
`CB-MBC`	0.8326	0.3639
`Tree-Tree`	0.8135	0.2977
`Polytree-Polytree`	0.8052	0.3422
`Pure Filter`	0.6733	0.2690
`Pure Wrapper`	0.8293	0.3650

From Table 2, we may conclude that our algorithm performs well. In fact, with the mean accuracy, `CB-MBC` presents the best accuracy, while with the global accuracy, `Pure Wrapper` slightly outperforms `CB-MBC`.

Finally, in Figure 3, we plot the computation learning time of the various learning algorithms, using the mean and global accuracy, for both synthetic and Emotions data sets.

Clearly, the algorithms using a filter approach require less computation than those using a wrapper approach. Moreover, the computation time of our algorithm is lower than the other wrapper approaches, mainly over the synthetic data set, which is basically due to the MBC CB-decomposability and the alleviation of MPE computation. Note also that the computation time with global accuracy is lower, since it is more difficult to improve the learned models, so that the algorithm ends in earlier iterations.

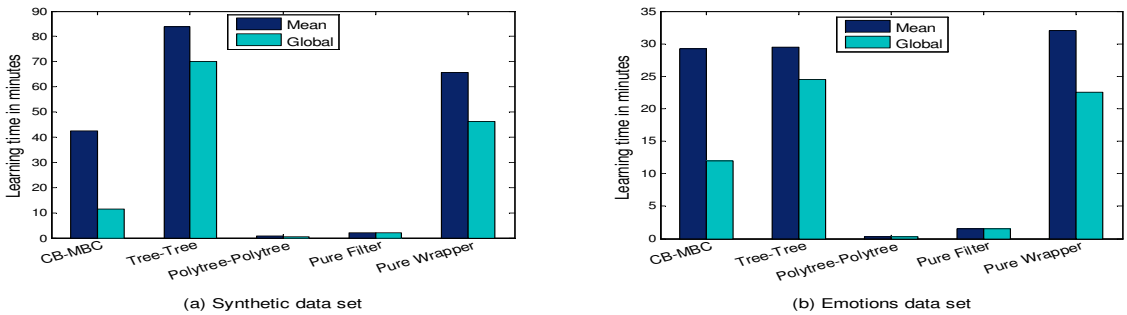

Figure 3: Computation learning times over (a) synthetic data set and (b) Emotions data set.

6 Conclusion

In this paper, we proposed a novel algorithm for learning CB-decomposable MBCs from data based on a wrapper forward selection approach. Indeed, CB-decomposability allows the alleviation of MPE computations. Experimental results with both synthetic and real-world data sets show that our algorithm performs well and requires less computation time than the state-of-the-art wrapper learning algorithms.

In the future, we intend to carry out additional experiments and investigate possible improvements of our algorithm. For instance, we intend to test the alternation between forward and backward selection techniques, and study the use of a filter approach mainly for feature subgraph learning in order to avoid the random arc additions between features. Furthermore, it would be interesting to extend our algorithm to deal with incremental learning from new incoming data, i.e. updating the current CB-decomposable MBC over time without a need to relearn it from scratch. In case of non-stationary domains, this may also require a detection mechanism to monitor the concept drift.

Acknowledgements

Work supported by projects TIN2007-62626 and Cajal Blue Brain (Spanish Ministry of Science and Innovation) and by project Dynamo (FONCICYT, European Union and Mexico).

References

C. Bielza, G. Li and P. Larrañaga. 2010. Multi-dimensional classification with Bayesian networks. Technical Report UPM-FI/DIA/2010-1, Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid.

C. Chow and C. Liu. 1968. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462-467.

P.R. de Waal and L.C. van der Gaag. 2007. Inference and learning in multi-dimensional Bayesian network classifiers. *In Proceedings of the Ninth European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty, Lecture Notes in Artificial Intelligence, Springer*, 4724:501–511.

M. Henrion. 1988. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. *In Proceedings of the Fourth Conference on the Uncertainty in Artificial Intelligence*, pages 149–163.

P. Langley and S. Sage. 1994. Induction of selective Bayesian classifiers. *In Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 399–406.

M. Qazi, G. Fung, S. Krishnan, R. Rosales, H. Steck, R.B. Rao, D. Poldermans and D. Chandrasekaran. 2007. Automated heart wall motion abnormality detection from ultrasound images using Bayesian networks. *In International Joint Conference on Artificial Intelligence*, pages 519–525.

G. Rebane and J. Pearl. 1989. The recovery of causal polytrees from statistical data. *In Proceedings of the Third Conference on Uncertainty in Artificial Intelligence*, pages 222–228.

J.D. Rodríguez and J.A. Lozano. 2008. Multi-objective learning of multi-dimensional Bayesian classifiers. *In Proceedings of the Eighth International Conference on Hybrid Intelligent Systems*, pages 501–506.

K. Trohidis, G. Tsoumakas, G. Kalliris and I. Vlahavas. 2008. Multilabel classification of music into emotions. *In Proceedings of the International Society for Music Information Retrieval Conference*, pages 325–330.

L.C. van der Gaag and P.R. de Waal. 2006. Muti-dimensional Bayesian network classifiers. *In Proceedings of the Third European Conference on Probabilistic Graphical Models*, pages 107–114.