International Journal of Approximate Reasoning 93 (2018) 74–87

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Tractability of most probable explanations in multidimensional Bayesian network classifiers ☆

Marco Benjumeda *, Concha Bielza, Pedro Larrañaga

Computational Intelligence Group, Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Spain

ARTICLE INFO

Article history:
Received 1 December 2016
Received in revised form 4 October 2017
Accepted 9 October 2017
Available online 27 October 2017

Keywords:
Multidimensional classification
Bayesian network classifiers
Most probable explanation complexity
Machine Learning

ABSTRACT

Multidimensional Bayesian network classifiers have gained popularity over the last few years due to their expressive power and their intuitive graphical representation. A drawback of this approach is that their use to perform multidimensional classification, a generalization of multi-label classification, can be very computationally demanding when there are a large number of class variables. Thus, a key challenge in this field is to ensure the tractability of these models during the learning process.

In this paper, we show how information about the most common queries of multidimensional Bayesian network classifiers affects the complexity of these models. We provide upper bounds for the complexity of the most probable explanations and marginals of class variables conditioned to an instantiation of all feature variables. We use these bounds to propose efficient strategies for bounding the complexity of multidimensional Bayesian network classifiers during the learning process, and provide a simple learning method with an order-based search that guarantees the tractability of the returned models. Experimental results show that our approach is competitive with other methods in the state of the art and also ensures the tractability of the learned models.

1. Introduction

Bayesian network classifiers [1] are one of the most widely used machine learning tools to address the problem of classification. Classification consists of assigning an instance to a class that is described by a set of features.

Multidimensional Bayesian network classifiers (MBCs) [2] extend Bayesian network classifiers to the problem of multidimensional classification. Multidimensional classification involves assigning an instance to a set of classes (instead of a single class) given the value of the set of features. This problem is common in several domains like text categorization (a text can be assigned to multiple topics), medicine (a patient may suffer from several diseases) or system monitoring (a system may break down from multiple failures).

MBCs are Bayesian networks (BN) with a restricted topology, where no arcs from feature variables to class variables are allowed. Each MBC is composed of a class subgraph, a bridge subgraph, and a feature subgraph (see Section 2). Inference in MBCs may have a high computational cost for some structures, even when the class and feature subgraphs are restricted to trees or polytrees.

☆ This paper is part of the Virtual special issue on the Eighth International Conference on Probabilistic Graphical Models, Edited by Giorgio Corani, Alessandro Antonucci, Cassio De Campos.

* Corresponding author.
E-mail addresses: marco.benjumeda.barquita@upm.es (M. Benjumeda), mcbielza@fi.upm.es (C. Bielza), pedro.larranaga@fi.upm.es (P. Larrañaga).

https://doi.org/10.1016/j.ijar.2017.10.024
0888-613X/

Although there is work in the literature addressing the problem of computational complexity in MBCs, the focus has not been on taking advantage of the most common type of queries of such models. In this paper, we study the computational complexity of most probable explanations (MPEs) and marginals of class variables in MBCs when an instantiation of the feature variables is given. The paper also provides upper bounds on the complexity of these models given additional restrictions on their structure that limit the treewidth of a transformation of it that we call the pruned graph.

Class-bridge (CB) decomposable MBCs [3] are capable of dividing the MPE problem into multiple simpler subproblems that can be computed independently in each of the MBC components. We prove that CB-decomposability can also be used to efficiently bound the complexity of MBCs during the learning process. We propose a learning method that uses these properties to search for tractable MBCs in the space of topological orderings.

This paper is an extended version of the work published in [4]. We extend the theoretical results to marginal computations, provide an alternative strategy for bounding the complexity of MBCs in the learning method, and extend the experiments using additional performance measures and real-world datasets. The rest of the paper is organized as follows. Section 2 describes MBCs, introduces CB-decomposability, and reviews previous work on inference complexity and learning in MBCs. Section 3 presents the new theoretical results with respect to the complexity of computations of MPEs and marginals in MBCs. Section 4 describes the method proposed for learning tractable MBCs. Section 5 reports the experimental results. Section 6 draws some conclusions and suggests future research lines.

2. Background

2.1. Multidimensional classification with Bayesian networks

A Bayesian network $\mathcal{B}$ represents a joint probability distribution over a set of random variables $\mathcal{V} = \{V_1, \ldots, V_n\}$. It is composed of a directed acyclic graph (DAG) $\mathcal{G}$ that represents the conditional dependences among the variables in $\mathcal{V}$, and a set of parameters $\Pr(V_i|\mathbf{Pa}_{\mathcal{G}}(V_i))$ (we use $\mathbf{Pa}_{\mathcal{G}}(V_i)$ to refer to the parents of V_i in $\mathcal{G}$) that represent the conditional probability distributions (CPDs) of each $V_i \in \mathcal{V}$ conditioned on its parents in $\mathcal{G}$. A joint probability distribution that satisfies the Markov condition with $\mathcal{G}$ is given by

$$\Pr(V_1, \ldots, V_n) = \prod_{i=1}^{n} \Pr(V_i|\mathbf{Pa}_{\mathcal{G}}(V_i)) \ . \tag{1}$$

Van der Gaag and de Waal [2] introduced *multidimensional Bayesian network classifiers* as an extension of Bayesian classifiers to multidimensional classification. MBCs are a special case of Bayesian networks with a restricted structure topology. They are defined as follows:

Definition 1. An MBC is a Bayesian network $\mathcal{B}$ over a set of variables $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$, where $\mathcal{V}$ is partitioned into two sets $\mathcal{C} = \{C_1, \ldots, C_d\}$, $d \geq 1$, of class variables and $\mathcal{F} = \{F_1, \ldots, F_m\}$, $m \geq 1$, of feature variables ($d + m = n$). The arcs in $\mathcal{G}$ are partitioned into three subsets, A_C, A_F, A_B, such that:

- $A_C \subseteq \mathcal{C} \times \mathcal{C}$ is composed of the arcs between the class variables having a subgraph $\mathcal{G}_C = (\mathcal{C}, A_C)$ – class subgraph – of $\mathcal{G}$ induced by $\mathcal{C}$.
- $A_F \subseteq \mathcal{F} \times \mathcal{F}$ is composed of the arcs between the feature variables having a subgraph $\mathcal{G}_F = (\mathcal{F}, A_F)$ – feature subgraph – of $\mathcal{G}$ induced by $\mathcal{F}$.
- $A_B \subseteq \mathcal{C} \times \mathcal{F}$ is composed of the arcs from the class variables to the feature variables having a subgraph $\mathcal{G}_B = (\mathcal{V}, A_B)$ – bridge subgraph – of $\mathcal{G}$ induced by $\mathcal{V}$ connecting class and feature variables.

Fig. 1 shows an example of the structure of an MBC and its corresponding subgraphs.

The problem of multidimensional classification in MBCs involves getting the most probable explanation (MPE) of the class variables given an instantiation of the feature variables, which is given by

$$\mathbf{c}^* = \arg\max_{\mathbf{c} \in \Omega_{\mathcal{C}}} \Pr(\mathbf{c}|\mathbf{f}) = \arg\max_{\mathbf{c} \in \Omega_{\mathcal{C}}} \Pr(\mathbf{c}, \mathbf{f}) \ , \tag{2}$$

where $\mathbf{f}$ is an instantiation of $\mathcal{F}$ and $\Omega_{\mathcal{C}}$ is the set containing all the possible configurations of $\mathcal{C}$.

2.2. Class-bridge decomposable multidimensional Bayesian network classifiers

An MBC is *class-bridge decomposable* [3] if it can be decomposed into multiple connected components, where each component is composed of all the nodes that are connected by an undirected path in $\mathcal{G}_C \cup \mathcal{G}_B$. Basically, the components of an MBC are the connected graphs obtained after removing the arcs of the feature subgraph from this MBC.

Definition 2. A CB-decomposable MBC is a BN $\mathcal{B}$ whose class subgraph and bridge subgraph are decomposed into r maximal components such that:

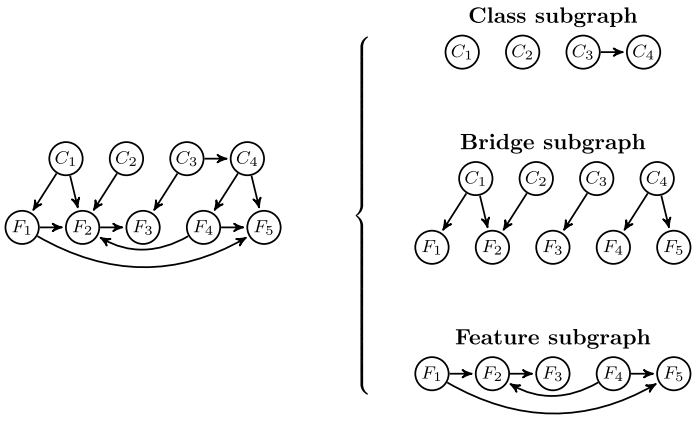
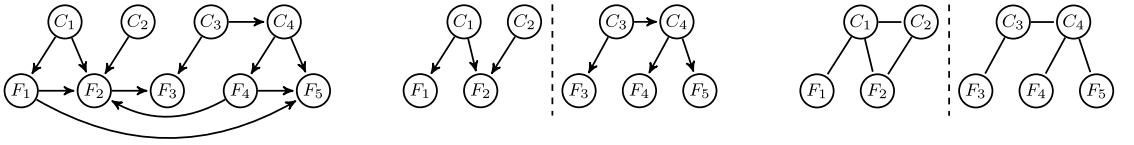


Fig. 1. MBC structure.



(a) MBC structure (b) Connected components (c) Moralized connected components

Fig. 2. MBC structure (a), connected components (b), and moralized connected components (c).

1. $\mathcal{G}_C \cup \mathcal{G}_B = \bigcup_{i=1}^{r} (\mathcal{G}_{C_i} \cup \mathcal{G}_{B_i})$, where $\mathcal{G}_{C_i} \cup \mathcal{G}_{B_i}$, $i = 1, \ldots, r$, are its maximal connected components.
2. $\mathbf{Ch}_{\mathcal{G}}(\mathcal{C}_i) \cap \mathbf{Ch}_{\mathcal{G}}(\mathcal{C}_j) = \emptyset$, with $i, j = 1, \ldots, r$ and $i \neq j$, where $\mathbf{Ch}_{\mathcal{G}}(\mathcal{C}_i)$ and $\mathbf{Ch}_{\mathcal{G}}(\mathcal{C}_j)$ denote the children of all variables in $\mathcal{C}_i$ and $\mathcal{C}_j$ respectively (the subsets of class variables in $\mathcal{G}_{C_i}$ and $\mathcal{G}_{C_j}$).

Bielza et al. [3] proved that exploiting the CB-decomposability of MBCs can reduce the number of computations required to perform multidimensional classification. Specifically, they showed that the MPE can be computed independently in each component, given that

$$\begin{aligned} \max_{\mathbf{c} \in \Omega_{\mathcal{C}}} \Pr(\mathbf{c}|\mathbf{f}) \propto \; & \prod_{i=1}^{r} \max_{\mathbf{c}_i \in \Omega_{\mathcal{C}_i}} \prod_{C_{ij} \in \mathcal{C}_i} \Pr(c_{ij}|\mathbf{Pa}_{\mathcal{G}_C}(C_{ij})) \\ & \prod_{F_{ij} \in \mathbf{Ch}_{\mathcal{G}}(\mathcal{C}_i)} \Pr(f_{ij}|\mathbf{Pa}_{\mathcal{G}_B}(F_{ij}), \mathbf{Pa}_{\mathcal{G}_F}(F_{ij})) \;, \end{aligned} \tag{3}$$

where $\mathcal{C}_i$ is the set containing the class variables that belong to component i, and $\Omega_{\mathcal{C}_i}$ is the set containing all the possible configurations of $\mathcal{C}_i$. This means that it is possible to maximize over each maximal connected component independently, therefore maximizing over lower dimensional spaces.

Let us consider the MBC shown in Fig. 2, which can be CB-decomposed in two connected components that contain nodes $\{C_1, C_2, F_1, F_2\}$ and $\{C_3, C_4, F_3, F_4, F_5\}$, respectively. To classify an instance $\mathbf{f} = (f_1, \ldots, f_5)$ we should get the MPE of $(C_1, \ldots, C_4)$ given $\mathbf{f}$. By Equation (3) we know that for any $\mathbf{c} = (c_1, \ldots, c_4)$,

$$\begin{aligned} \max_{\mathbf{c} \in \Omega_{\mathcal{C}}} \Pr(\mathbf{c}|\mathbf{f}) \propto \; & \Big(\max_{c_1, c_2} \Pr(c_1)\Pr(c_2)\Pr(f_1|c_1)\Pr(f_2|c_1, c_2, f_1, f_4), \\ & \max_{c_3, c_4} \Pr(c_3)\Pr(c_4|c_3)\Pr(f_3|c_3, f_2)\Pr(f_4|c_4) \\ & \Pr(f_5|c_4, f_1, f_4) \Big) \;. \end{aligned}$$

Thus, the MPE can be computed maximizing over (C_1, C_2) and (C_3, C_4) independently.

2.3. Complexity of most probable explanations in multidimensional Bayesian network classifiers

Assuming that all the feature variables are observed, performing multidimensional classification in an MBC $\mathcal{B}$ with class variables $\mathcal{C} = \{C_1, \ldots, C_d\}$ and feature variables $\mathcal{F} = \{F_1, \ldots F_m\}$ is equivalent to obtaining the MPE of the class variables conditioned on an instance $\mathbf{f}$ of the features. If there are unobserved feature variables, performing multidimensional classification in $\mathcal{B}$ is equivalent to obtaining not the MPE in $(C_1, \ldots, C_d)$ but the maximum a posteriori hypothesis (MAP). This can be intractable even if the treewidth of $\mathcal{B}$ is bounded [5].

Fig. 3 shows an example of both cases where multidimensional classification in an MBC is equivalent to obtaining the MPE and the MAP, respectively.

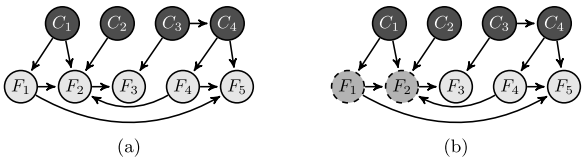


Fig. 3. Multidimensional classification with an MBC. In (a) the values $\boldsymbol{f} = (f_1, \ldots, f_5)$ of all the features are given, and multidimensional classification is equivalent to obtaining the MPE (i.e., $\arg\max_{\mathbf{c}\in\Omega_C} \Pr(\mathbf{c}|\mathbf{f})$). In (b) the values of F_1 and F_2 are missing, and multidimensional classification is equivalent to obtaining the MAP (i.e., $\arg\max_{\mathbf{c}\in\Omega_C} \Pr(\mathbf{c}|f_3, f_4, f_5)$).

Existing research addresses the complexity of multidimensional classification in MBCs as the complexity of computing the MPE. Thus, they implicitly assume that MPE queries will not contain missing values (i.e., the values of all the feature variables will be given). Otherwise, the resulting MPE would provide the most probable instantiation of $(C_1, \ldots, C_d, F_{m_1}, \ldots, F_{m_k})$, where $F_{m_1}, \ldots, F_{m_k}$ are the non-instantiated features. Note that the most probable instantiation of $(C_1, \ldots, C_d)$, that is equivalent to the MAP of the class variables given an instantiation of the observed features in this case, may differ from the most probable instantiation of $(C_1, \ldots, C_d, F_{m_1}, \ldots, F_{m_k})$.

In this paper, we also focus on the case where all the feature variables are observed. Hence, we consider that, to perform multidimensional classification, an MBC obtains $\arg\max_{\mathbf{c}\in\Omega_C} \Pr(\mathbf{c}|\mathbf{f})$.

MPE is generally NP-hard [6] and known exact methods for MPE computations in a BN $\mathcal{B}$ are exponential in the treewidth of $\mathcal{G}$, where $\mathcal{G}$ is the structure of $\mathcal{B}$. Nevertheless, MPE can be computed in polynomial time in $\mathcal{B}$ if the treewidth of $\mathcal{G}$ is bounded [7].

Given MBC structural constraints, further bounds on their inference complexity have been found. De Waal and van der Gaag [8] demonstrated that

$$\text{treewidth}(\mathcal{G}) \leq \text{treewidth}(\mathcal{G}_F) + d \ ,$$

where $\mathcal{G}_F$ is the feature subgraph and d is the number of class variables. This means that $\mathcal{B}$ could perform multidimensional classification in polynomial time if the addition of the treewidth of the feature subgraph and the number of class variables is bounded.

Furthermore, Kwisthout [6] showed that for any CB-decomposable MBC with structure $\mathcal{G}$

$$\text{treewidth}(\mathcal{G}) \leq \text{treewidth}(\mathcal{G}_F) + |d_{\max}| \ ,$$

where $|d_{\max}|$ is the number of class variables of the component with the maximum number of class variables. Hence, the MPE can be computed in polynomial time if the treewidth of $\mathcal{G}_F$ and the number of class variables of each component of $\mathcal{G}$ are bounded.

Pastink and van der Gaag [9] focused on MBCs with an empty feature subgraph. To bound the structure, they used

$$\text{treewidth}(\mathcal{G}_{\bar{F}}) < \text{treewidth}(\mathcal{G}') \ ,$$

where $\mathcal{G}_{\bar{F}}$ is the structure of an MBC with empty feature subgraph and $\mathcal{G}'$ is the graph obtained after moralizing $\mathcal{G}_{\bar{F}}$ and then removing all its feature nodes from the moralized graph.

When computing the MPE in a BN given an evidence $\mathbf{f}$, we can simplify the structure of the network by pruning every arc $V_i \to V_j$ such that V_i appears in $\mathbf{f}$. Pruning arc $V_i \to V_j$ for evidence $\mathbf{f}$ from a BN means removing arc $V_i \to V_j$ and the parameters of V_j that are not compatible with $\mathbf{f}$.

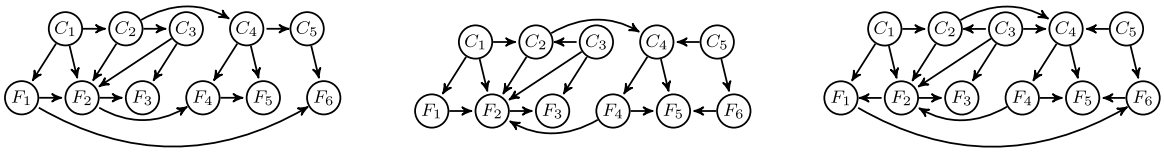
As mentioned above, previous research uses the treewidth of $\mathcal{G}$ to bound the inference complexity, exploiting the restrictions on the topology of $\mathcal{G}$, but without considering the known query-dependent information, that is, that all the feature variables are instantiated when we compute the MPE in $\mathcal{B}$. Here, we take advantage of the above to bound the complexity of multidimensional classification in MBCs.

2.4. Previous work on learning MBCs

The problem of learning MBCs from data has been addressed before. The literature contains methods for learning different families of MBCs, depending on the type of class and feature subgraphs that they can obtain (trees, forests, polytrees or DAGs). Here we denote the family of the MBC using <class subgraph> – <feature subgraph> (e.g., tree–DAG has a tree as the class subgraph and a DAG as the feature subgraph). Fig. 4 shows some of the most popular MBC families.

Methods have been proposed for learning tree–tree [2], polytree–polytree [8] and DAG–DAG [3] MBCs. These approaches do not explicitly consider the inference complexity of the learned models. Hence, they may lead to MBCs where the MPE cannot be solved efficiently, unless the number d of class variables is very small.

There are also other approaches in the literature that consider the complexity of the MBCs during the learning process. Corani et al. [10] proposed a method for learning sparse MBCs with a forest class subgraph and an empty feature subgraph, and Borchani et al. [11] introduced the first method to learn CB-decomposable MBCs. However, neither provides guarantees regarding the complexity of multidimensional classification in the models. Pastink and van der Gaag [9] proposed a method for learning tree–empty MBCs of bounded treewidth, providing an optional step to learn a forest feature subgraph, and



(a) Tree–tree (b) Polytree–polytree (c) DAG–DAG

Fig. 4. Widely used MBC families ordered from the least general (a) to the most general (c). Note that tree–tree is a special case of polytree–polytree, which is likewise a special case of DAG–DAG.

guaranteeing the tractability of the resulting models. The method computes the treewidth of each candidate and rejects any that exceeds the treewidth bound.

Markov random fields have also been used for multi-label classification. Ghamrawi and McCallum [12] proposed two pairwise models, CML and CMLF. CML learns a factor between each pair of class variables and between each pairwise combination of a class variable and a feature variable. CMLF also learns the latter factors, but instead of learning the former it learns a factor between each combination of two class variables and a feature variable, increasing the expressiveness of CML. Exact inference in these models requires computing a factor over all the possible configurations of the class variables. Hence, it is intractable when the number of class variables is not small. [13] proposes a method that learns an undirected graph between the class variables, and learns a base model (e.g., naïve Bayes) for each pair of connected class variables. The base model gives a factor over a pair of class variables given an instance of the feature variables. The drawback of this approach is that the number of base models is huge if the graph between the class variables is not sparsely connected.

In this paper we bound the complexity of (the most general) DAG–DAG MBCs by bounding the treewidth of a transformation of their structures (similar to the transformation used by Pastink and van der Gaag [9]). However, we do not bound the treewidth of their complete structures. Moreover, we propose an additional strategy that takes advantage of the CB-decomposability of MBCs to compute these bounds efficiently. We use these bounds to learn MBCs where multidimensional classification can be performed in polynomial time. We show that even high treewidth MBCs may be tractable given other structural constraints.

3. Theoretical results on MPE and marginal computations

In BNs with bounded treewidth, both evidence propagation and MPEs can be computed in polynomial time. In the case of MBCs (which are, in fact, also BNs), this is also true, but it is possible to exploit the restrictions on the network structure and the information about the queries sent to the MBCs. From the structure of MBCs, we know that there are no arcs from the feature to the class nodes. Also, the values of all the features should appear in the evidence.

As multidimensional classification in MBCs involves obtaining the MPE of the class variables given an instantiation of all the feature variables, we focus on finding bounds for this problem. Nevertheless, the results are extended to marginal computations because it is sometimes worth calculating the probability of a configuration of certain class variables given the value of all the features, and the extension is straightforward.

The complexity of inference in BNs is query dependent, given that the parameters of a network can be updated with the value of the evidence variables before performing inference.

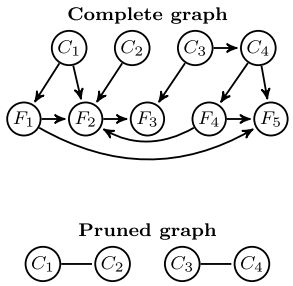
Definition 3. Let $\mathcal{G} = (\mathcal{C} \cup \mathcal{F}, \mathcal{A}_C \cup \mathcal{A}_B \cup \mathcal{A}_F)$ be the structure of an MBC $\mathcal{B}$. The pruned graph $\mathcal{G}'$ of $\mathcal{G}$ is the result of moralizing $\mathcal{G}$ and then removing the feature nodes from the resulting graph.

Theorem 1 states that MPE and marginal computations in an MBC are tractable if the treewidth of its pruned graph is bounded. This transformation was used by Pastink and van der Gaag [9] to bound the treewidth of tree–empty MBCs. Here, we extend it to bound the complexity of (the more general) DAG–DAG MBCs.

Theorem 1. *Let $\mathcal{G} = (\mathcal{C} \cup \mathcal{F}, \mathcal{A}_C \cup \mathcal{A}_B \cup \mathcal{A}_F)$ be the structure of an MBC $\mathcal{B}$, and* $\mathbf{f}$ *be an instantiation of $\mathcal{F}$. If the treewidth of its pruned graph $\mathcal{G}'$ and the number of parents of each node that belongs to $\mathcal{F}$ are bounded, $\mathcal{B}$ can compute MPEs and marginals in polynomial time given* $\mathbf{f}$.

Proof. Suppose that the CPD of each node $V_i \in \mathcal{C} \cup \mathcal{F}$ is represented by a potential ϕ_i. ϕ_i is updated with $\mathbf{f}$ by removing the entries that are not compatible with $\mathbf{f}$. This can be done in linear time in the size of ϕ_i, that is exponential in the number of parents of V_i in $\mathcal{G}$. Hence, the nodes in $\mathcal{F}$ can be updated with $\mathbf{f}$ in polynomial time if the number of parents of each node in $\mathcal{F}$ is bounded.

After updating $\mathcal{G}$ with $\mathbf{f}$, the domain of each potential ϕ_f of $V_f \in \mathcal{F}$ is $\mathbf{Pa}_{\mathcal{G}}(V_f) \cap \mathcal{C}$. There is an undirected link in $\mathcal{G}'$ between each node in $\mathbf{Pa}_{\mathcal{G}}(V_f) \cap \mathcal{C}$. It is evident that the width of the best elimination order for the resulting potentials is equal to the treewidth of $\mathcal{G}'$. As the width of the best elimination order bounds the complexity of MPE and marginal computations, if the treewidth of $\mathcal{G}'$ is bounded, $\mathcal{B}$ can compute MPEs and marginals in polynomial time given $\mathbf{f}$. □



Potential	Before	After
ϕ_{C_1}	$\{C_1\}$	$\{C_1\}$
ϕ_{C_2}	$\{C_2\}$	$\{C_2\}$
ϕ_{C_3}	$\{C_3\}$	$\{C_3\}$
ϕ_{C_4}	$\{C_3, C_4\}$	$\{C_3, C_4\}$
ϕ_{F_1}	$\{C_1, F_1\}$	$\{C_1\}$
ϕ_{F_2}	$\{C_1, C_2, F_1, F_2, F_4\}$	$\{C_1, C_2\}$
ϕ_{F_3}	$\{C_3, F_2, F_3\}$	$\{C_3\}$
ϕ_{F_4}	$\{C_4, F_4\}$	$\{C_4\}$
ϕ_{F_5}	$\{C_4, F_1, F_4\}$	$\{C_4\}$

Fig. 5. MBC structure and pruned graph (left), and domain of the potential of each node before and after they are updated with evidence $\mathbf{f} = (f_1, \ldots, f_5)$ (right). Note that the treewidth of the pruned graph is smaller than the number of class variables.

Fig. 5 shows an example of the structure of an MBC and its pruned graph. It also illustrates that all the variables belonging to the domain of the same potential $\phi_i \in \{\phi_{C_1}, \ldots, \phi_{C_4}, \phi_{F_1}, \ldots, \phi_{F_5}\}$ updated with an instance $\mathbf{f} = (f_1, \ldots, f_5)$ of the features are connected by a link in the pruned graph (and vice versa). This means that the treewidth of the pruned graph is equal to the width of the best elimination order in the updated potentials.

Although the computational cost of calculating the treewidth of the pruned graph $\mathcal{G}'$ is less than calculating the treewidth of the complete structure $\mathcal{G}$, the exact computation of the treewidth of a graph is an NP-complete problem [14].

There are multiple approaches that calculate the exact treewidth of a graph in exponential time [15,16], but they are mostly intractable in practice. We can compute whether the treewidth of a graph is less than or equal to a constant k in linear time if k is fixed, but obtaining the solution of this inequality is super-exponential in the treewidth [17]. Thus, it is intractable unless k is very small.

As the treewidth of $\mathcal{G}$ is equal to the width of the best elimination order for $\mathcal{G}$, heuristic methods for searching good elimination orders are often applied. Two popular heuristics eliminate, in each iteration, the node of smallest degree in the graph [18] or the node that produces the minimum number of fill-in edges [19].

Other approaches consist of applying graph recognition techniques [20,21], local search methods [22,23], or evolutionary methods [24]. Nevertheless, these methods are usually computationally demanding, especially if we aim to bound the treewidth of each candidate during the learning process.

Fortunately, Corollary 1 shows that if the number of class variables of an MBC $\mathcal{B}$ is bounded, then we can perform inference in $\mathcal{B}$ in polynomial time.

Corollary 1. *Let $\mathcal{G} = (\mathcal{C} \cup \mathcal{F}, \mathcal{A}_C \cup \mathcal{A}_B \cup \mathcal{A}_F)$ be the structure of an MBC $\mathcal{B}$, and $\mathbf{f}$ be an instantiation of $\mathcal{F}$. If the number of class variables d and the number of parents of each node in $\mathcal{F}$ are bounded, $\mathcal{B}$ can compute MPEs and marginals in polynomial time given $\mathbf{f}$.*

Proof. Let $\mathcal{G}'$ be the pruned graph of $\mathcal{G}$. As each node in $\mathcal{G}'$ belongs to $\mathcal{C}$, treewidth$(\mathcal{G}') \leq d$. Hence, from Theorem 1 we know that if the number of parents of each feature and d are bounded, $\mathcal{B}$ can compute MPEs and marginals in polynomial time given $\mathbf{f}$. □

As the pruned graph only contains class nodes, it is patent that its treewidth is always smaller than the number of class variables in the classifier. Let us consider the MBC shown in Fig. 5 and its pruned graph. The nodes in the pruned graph are $\{C_1, \ldots, C_4\}$, so its treewidth can never be greater than 3.

When the number d of class variables of $\mathcal{B}$ is not small, it is not so simple to decide whether $\mathcal{B}$ can perform multidimensional classification efficiently. Nevertheless, if the classifier is CB-decomposable, we can show that simply bounding the maximum number of class nodes per component also bounds the inference complexity of the MBCs, as shown in Corollary 2.

Corollary 2. *Let $\mathcal{G} = (\mathcal{C} \cup \mathcal{F}, \mathcal{A}_C \cup \mathcal{A}_B \cup \mathcal{A}_B)$ be the structure of a CB-decomposable MBC $\mathcal{B}$, and $\mathbf{f}$ be an instantiation of $\mathcal{F}$. If the number of class variables in each component of $\mathcal{G}$ and the number of parents of each node in $\mathcal{F}$ are bounded, $\mathcal{B}$ can compute MPEs and marginals in polynomial time given $\mathbf{f}$.*

Proof. Let $\mathcal{G}'$ be the pruned graph of $\mathcal{G}$. If $\mathcal{G}$ is CB-decomposable into r components $\mathcal{G}_1, \ldots, \mathcal{G}_r$, then $\mathcal{G}'$ is composed of r unconnected subgraphs $\mathcal{G}'_1, \ldots, \mathcal{G}'_r$, such that $\mathcal{V}'_i = \mathcal{V}_i \cap \mathcal{C}$, $i = 1, \ldots, r$, where $\mathcal{V}_i$ and $\mathcal{V}'_i$ are the nodes in $\mathcal{G}_i$ and $\mathcal{G}'_i$, respectively. As treewidth$(\mathcal{G}') = \max_i\{\text{treewidth}(\mathcal{G}'_i)\} < \max_i |\mathcal{V}'_i| = \max_i |\mathcal{V}_i \cap \mathcal{C}|$, we know from Theorem 1 that if the number of parents of each feature and the number of class variables in each component of $\mathcal{G}$ are bounded, $\mathcal{B}$ can compute MPEs and marginals in polynomial time given $\mathbf{f}$. □

Data: Dataset $\mathcal{D}$, ordering of class variables $\mathbf{O}_C = (O_{C_1}, \cdots, O_{C_d})$, ordering of feature variables $\mathbf{O}_F = (O_{F_1}, \cdots, O_{F_m})$, bound k
Result: MBC structure $\mathcal{G}$, best score S

1 $\mathcal{G} \leftarrow$ empty DAG;
2 **for** $V_i \in (O_{C_1}, \cdots, O_{C_d}, O_{F_1}, \cdots, O_{F_m})$ **do**
3 improve $\leftarrow$ **true**;
4 **while** improve **do**
5 Let V_j be the node that maximizes score$(\mathcal{D}, V_i, \mathbf{Pa}_{\mathcal{G}}(V_i) \cup \{V_j\})$, such that adding V_j to $\mathbf{Pa}_{\mathcal{G}}(V_i)$ does not exceed:
6 **a)** The bound k in the treewidth of the pruned graph of $\mathcal{G}$
7 **b)** The bound k of class variables per component in $\mathcal{G}$
8 **if** score$(\mathcal{D}, V_i, \mathbf{Pa}_{\mathcal{G}}(V_i) \cup \{V_j\}) >$ score$(\mathcal{D}, V_i, \mathbf{Pa}_{\mathcal{G}}(V_i))$ **then**
9 $\mathbf{Pa}_{\mathcal{G}}(V_i) \leftarrow \mathbf{Pa}_{\mathcal{G}}(V_i) \cup \{V_j\}$;
10 **else**
11 improve $\leftarrow$ **false**;
12 **end**
13 **end**
14 **end**
15 $S \leftarrow \sum_{i=1}^{n}$ score$(\mathcal{D}, V_i, \mathbf{Pa}_{\mathcal{G}}(V_i))$, with $n = d + m$;
16 **return** $\mathcal{G}$, S ;

Algorithm 1: Greedy search of tractable CB-decomposable MBCs (greedyMBC). Lines 6 and 7 correspond to the two alternatives proposed in this paper to bound MBC complexity.

Fig. 5 shows that the treewidth of the pruned graph is bounded by the maximum number of class variables per component (two in this case), given that there is no path from C_i to C_j in the pruned graph if two class nodes C_i and C_j are in two different connected components.

4. Learning tractable multidimensional Bayesian network classifiers

Given that inference in a BN is tractable if the treewidth of its structure is bounded, most existing algorithms for learning BNs with low inference complexity bound the treewidth of the networks during the learning process, rejecting any candidates that exceed the treewidth bound [25–27].

In the case of an MBC, instead of bounding the treewidth of its complete structure, we focus on bounding the treewidth of its pruned graph. We adapt order-based search (OBS) [28] to learn tractable MBCs with DAG–DAG structure. As the order of the variables in greedy search restricts the structure of the learned networks (i.e., a node can only be set as the parent of another node if it has been visited previously), OBS can be easily adapted to learn MBCs by considering only those orderings of the variables where the class variables precede the feature variables. In this manner, the parents of class variables must necessarily be other class variables. This is consistent with the MBC structure.

We use Algorithm 1 to learn the structure of MBCs given an ordering of the class ($\mathbf{O}_C$) and feature ($\mathbf{O}_F$) variables and a bound k on the treewidth of the pruned graph or, alternatively, on the maximum number of class variables per component. We do not specify a definite scoring function because any score used to evaluate BNs can be applied. We assume that the score is decomposable and must be maximized.

To bound the inference complexity, we provide two alternatives. The first approach (line 6 of Algorithm 1) involves computing the treewidth of the pruned graph (e.g., using the Min-Fill algorithm) of each MBC candidate and rejecting any that exceed a limit k that bounds the complexity of multidimensional classification (see Theorem 1). This alternative is more accurate than the second option, but may be very computationally demanding.

The second approach (line 7 of Algorithm 1) consists of using CB-decomposability to bound the complexity of multidimensional classification in the models (see Corollary 2). In this case, we limit the maximum number of class variables per component, rejecting any candidates that exceed this bound. The main benefit of this alternative is that the computational cost of computing this bound is negligible since it merely involves counting the number of class variables that belong to each component of the MBC.

An effective strategy used to learn BNs in the space of orderings is to perform a greedy process applying local changes among the orderings and picking the best change in each step [29]. A tabu list can also be used to reduce the computational cost, and random restarts can be useful for avoiding local optima. Algorithm 2 starts with a random ordering of the class and feature variables (line 1). In each iteration, it finds the swap between two consecutive nodes in the current ordering that maximizes the score and is not on the tabu list (line 6). The tabu list (line 10), that represents pairs of nodes that have been swapped recently, is used to prevent the algorithm from reversing recently applied swaps.

Table 1 compares the properties of our method with other popular MBC learning methods in the state of the art. Note that our approach is one of the few that provides theoretical guarantees with respect to the complexity of the models. Also, it allows highly expressive structures to be learned since it does not bound the treewidth of the complete graph.

5. Experimental results

To test the performance of our approach, we compared it with other state-of-the-art methods, including the tree–tree [2], polytree–polytree [8] and pure filter (DAG–DAG) [3] algorithms. We also compared it to a version of the method proposed by

Data: Dataset $\mathcal{D}$, class variables $\mathcal{C}$, feature variables $\mathcal{F}$, bound k, size of the tabu list t
Result: MBC structure $\mathcal{G}_{\text{best}}$
1 $\mathbf{O}_C, \mathbf{O}_F \leftarrow$ random permutation of $\mathcal{C}$ and $\mathcal{F}$;
2 $\mathcal{G}_{\text{best}}, S_{\text{best}} \leftarrow$ greedyMBC$(\mathcal{D}, \mathbf{O}_C, \mathbf{O}_F, k)$;
3 tabu $\leftarrow$ empty list;
4 improve $\leftarrow$ **true**;
5 **while** improve **do**
6 Let $\mathbf{O}'_C, \mathbf{O}'_F$ be the permutation of $\mathcal{C}$ and $\mathcal{F}$ obtained by applying a swap $V_i \leftrightarrow V_j$ in $\mathbf{O}_C$ or $\mathbf{O}_F$ that maximize S (with $\mathcal{G}, S =$ greedyMBC$(\mathcal{D}, \mathbf{O}'_C, \mathbf{O}'_F, k)$), such that $V_i \leftrightarrow V_j \notin$ tabu;
7 $\mathcal{G}, S \leftarrow$ greedyMBC$(\mathcal{D}, \mathbf{O}'_C, \mathbf{O}'_F, k)$;
8 **if** $S > S_{\text{best}}$ **then**
9 $\mathcal{G}_{\text{best}}, S_{\text{best}} \leftarrow \mathcal{G}, S$;
10 push $V_i \leftrightarrow V_j$ (best swap in line 6) into tabu;
11 $\mathbf{O}_C, \mathbf{O}_F \leftarrow \mathbf{O}'_C, \mathbf{O}'_F$
12 **else**
13 improve $\leftarrow$ **false**;
14 **end**
15 **if** size(tabu) $> t$ **then**
16 remove first element of tabu;
17 **end**
18 **end**
19 **return** $\mathcal{G}_{\text{best}}$;

Algorithm 2: Ordering-based search of tractable CB-decomposable MBCs (CB–OBS).

Table 1
Comparison of the properties of different MBC learning methods. For each approach, the table shows the family of MBC returned by the method, whether it addresses the problem of computational complexity of the learned models, whether it provides theoretical guarantees on the tractability of the models, and which part of the structure is bounded to ensure tractability. Note that Pastink and van der Gaag [9] provides an optional step to augment the empty feature subgraph to a forest.

	Family	Addresses complexity	Theoretical guarantees	Bound
[2]	Tree–tree			
[8]	Polytree–polytree			
[3]	DAG–DAG			
[9]	Tree–empty	x	x	Complete graph
[10]	Forest–empty	x		
[11]	DAG–DAG	x		
This work	DAG–DAG	x	x	Pruned graph

Pastink and van der Gaag [9] (small–tw). Instead of the branch and bound approach that they proposed to search the bridge subgraph, we used a greedy search process that picks the best parent set of each feature variable that does not exceed the treewidth bound in each iteration, given that the computational cost of the former is too high for this experimental framework. We used the Bayesian information criterion (BIC) as the scoring function for our method. CB–OBSp and CB–OBSc denote our approach when we bound the treewidth of the pruned graph and the number of class variables per component, respectively (options a) and b) in Algorithm 1). In all cases, we used a Bayesian estimation of the parameters with all hyperparameters equal to 0.05.

We generated a dataset of 5000 samples from three BNs. ANDES [30] is an intelligent tutoring system for Newtonian physics, MUNIN1 [31] is a network for the diagnosis of neuromuscular disorders, and PIGS [32] is a pedigree of breeding pigs. We also tested the different approaches on two real-world datasets: ENRON, a dataset for email classification [33], and MEDICAL, a dataset for medical text classification [34].

In the datasets generated from BNs, we selected one third of the variables at random as class variables. To reduce the dimensionality of all the datasets and remove uninformative features, we applied an information gain filter for each of the classes (i.e., we select the five features with highest information gain with respect to each class variable), generating a subset of selected features for each class variable. The definitive subset of features is the union of the subsets selected for each variable. The basic properties of the datasets are described in Table 2.

To test the performance of the methods, we used six different measures. The mean accuracy of the classifiers averages the accuracy values of all the class variables individually, as described below for N samples and d classes:

$$\text{acc}_M = \frac{1}{d \cdot N} \sum_{i=1}^{d} \sum_{j=1}^{N} \delta(c'_{ij}, c_{ij}) \ , \tag{4}$$

where c'_{ij} represents the predicted class label for variable C_j in instance i, c_{ij} is its true value, and $\delta(c'_{ij}, c_{ij}) = 1$ if $c'_{ij} = c_{ij}$, and 0 otherwise.

Table 2
Basic properties of the datasets. Number of classes, filtered features, and instances for each dataset.

Dataset	Classes	Features	Instances
ANDES	74	135	5000
MUNIN1	62	107	5000
PIGS	110	206	5000
ENRON	53	124	1702
MEDICAL	45	110	978

The global accuracy measures the fraction of instances in which the labels of all the classes were correctly assigned, and is given by:

$$\mathrm{acc}_G = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{c}'_i, \mathbf{c}_i) \ , \tag{5}$$

where $\delta(\mathbf{c}'_i, \mathbf{c}_i) = 1$ if $\mathbf{c}'_i = \mathbf{c}_i$, and 0 otherwise.

The log-likelihood of the model structure given the test dataset indicates how well the structure of each MBC fits the data.

The multi-class area under the ROC curve (AUC) proposed by Provost and Domingos [35] for a single class variable C is defined as:

$$\mathrm{AUC}_M = \sum_{c_i \in \Omega_C}^{N} \mathrm{AUC}(c_i) \cdot \Pr(c_i) \ , \tag{6}$$

where $\mathrm{AUC}(c_i)$ is the AUC for variable C using c_i as the target value against the rest of possible classes of C. To adapt the results to multidimensional classification, we use macro and micro averages [36] of AUC_M over all the class variables.

Additionally, we computed the marginals of the class variables given the value of the features in ANDES, MUNIN1, and PIGS. We give the mean square error (MSE) of the predictions obtained with the learned models with respect to the predictions obtained with the real BNs.

5.1. Results

In each dataset, we performed a 5-fold cross-validation to estimate the performance of each method. Table 3 compares the complexity of the learned models for each dataset. The complexity measures shown in each column are the treewidth of the pruned graph (τ_p), the treewidth of the complete graph (τ), computed using the Min-Fill algorithm, the size of the factors induced by variable elimination for solving the MPE, and the learning time (time_l) in seconds. The time complexity of variable elimination is given by the size of the induced factors. Table 4 compares the fitting and accuracy of the models. For each dataset and method, we show the mean accuracy (acc_M), the global accuracy (acc_G), and the log-likelihood (log–lik). Table 5 gives the micro-averaged AUC ($\mathrm{AUC}_{\mathrm{micro}}$), the macro-averaged AUC ($\mathrm{AUC}_{\mathrm{macro}}$), and the mean square error in the marginal computations (MSE).

In all tables, we give the mean value $\pm$ the standard deviation in 5-fold cross-validation for each measure. In Tables 4 and 5, we use – to denote that the complexity of a model did not allow us to compute MPEs or marginals because of time and space constraints, and we show the best results in bold.

In all cases, the bound on the number of class variables per component k in CB–OBSc was set to 15. Small values of k usually returned MBCs with a very low treewidth, which detracts from classification accuracy, while big values of k did not guarantee the tractability of the learned MBCs. The treewidth bound of the pruned graph τ_p for CB–OBSp, and the treewidth bound τ for small–tw were set to 5. Other small values of τ produced similar results.

In the experiments, the size of the models returned by CB–OBSp, CB–OBSc and small–tw was always small, and the treewidth of the pruned graph was low. On the other hand, some unrestricted methods (i.e., tree–tree, and polytree–polytree) produced huge models in some scenarios, where multidimensional classification was intractable. Unlike small–tw, the CB–OBSp and CB–OBSc methods were able to learn models with a high treewidth for their complete graph and with small treewidth for their pruned graph. This means that these methods help to output more expressive structures without affecting the complexity of multidimensional classification.

We compared the experimental results obtained for each measure in all the datasets using the Friedman test with $\alpha < 0.05$ and Holm's [37] and Shaffer's [38] procedures. Both Holm's and Shaffer's procedures associate pairwise comparisons with a set of hypotheses, and perform a step-down process with the corresponding set of ordered p-values to adjust the value of α. Garcia and Herrera [39] provided a review of statistical comparison procedures. The significant differences obtained for each measure are:

- acc_M: CB–OBSp was found to be significantly better than small–tw by both procedures, and significantly better than polytree–polytree by Holm's procedure.

Table 3
Complexity comparison.

	Method	τ_p	τ	size	time$_l$
ANDES	CB–OBSp	4.3 ± 0.5	11.0 ± 2.9	898.0 ± 56.2	1240.7 ± 12.6
	CB–OBSc	4.7 ± 0.5	11.7 ± 0.9	706.0 ± 111.4	146.0 ± 0.8
	Tree–tree	7.3 ± 0.9	24.0 ± 2.2	3278.0 ± 656.5	136.8 ± 1.6
	Polytree–polytree	5.0 ± 0.8	26.3 ± 2.1	1123.3 ± 195.0	46.6 ± 0.5
	DAG–DAG	8.8 ± 1.7	36.4 ± 3.0	8790.8 ± 6730.1	707.0 ± 94.5
	Small–tw	5.0 ± 0.0	5.0 ± 0.0	1444.7 ± 46.7	28.6 ± 0.1
MUNIN1	CB–OBSp	3.7 ± 0.5	10.3 ± 0.5	406.0 ± 62.1	350.4 ± 9.5
	CB–OBSc	3.3 ± 0.5	10.7 ± 2.5	348.0 ± 8.2	45.3 ± 4.1
	Tree–tree	7.0 ± 0.8	22.0 ± 0.8	1874.0 ± 520.0	40.6 ± 1.4
	Polytree–polytree	7.0 ± 0.0	24.0 ± 0.8	1807.3 ± 294.7	36.8 ± 0.7
	DAG–DAG	7.0 ± 0.8	28.3 ± 1.2	2270.0 ± 869.2	217.2 ± 6.2
	Small–tw	5.0 ± 0.0	5.0 ± 0.0	976.7 ± 8.2	9.4 ± 0.7
PIGS	CB–OBSp	5.0 ± 0.0	11.0 ± 0.8	7581.0 ± 820.6	4246.5 ± 27.9
	CB–OBSc	3.0 ± 0.0	10.7 ± 0.9	2012.0 ± 122.2	333.6 ± 14.2
	Tree–tree	12.0 ± 0.8	23.3 ± 2.5	$(4.42 \pm 2.55) \times 10^6$	484.4 ± 12.0
	Polytree–polytree	12.0 ± 0.0	37.7 ± 3.7	$(4.19 \pm 5.85) \times 10^6$	445.1 ± 1.2
	DAG–DAG	7.0 ± 0.8	29.5 ± 2.6	79244.2 ± 926.4	7197.2 ± 3.7
	Small–tw	5.0 ± 0.0	5.0 ± 0.0	12063.0 ± 404.4	103.6 ± 1.9
ENRON	CB–OBSp	4.2 ± 0.4	29.2 ± 2.6	378.8 ± 40	395.6 ± 13.1
	CB–OBSc	3.6 ± 0.5	27.4 ± 2.0	255.2 ± 22	26.4 ± 0.7
	Tree–tree	5.0 ± 0.0	11.0 ± 0.6	574.8 ± 55	16.9 ± 0.3
	Polytree–polytree	6.8 ± 1.2	13.2 ± 1.9	1106.0 ± 724	16.1 ± 0.6
	DAG–DAG	4.4 ± 0.5	25.6 ± 3.4	481.2 ± 71	14.2 ± 2.2
	Small–tw	5.0 ± 0.0	5.0 ± 0.0	574.8 ± 55	6.0 ± 0.1
MEDICAL	CB–OBSp	2.0 ± 0.0	5.2 ± 0.4	151.6 ± 1	110.8 ± 4.4
	CB–OBSc	2.4 ± 0.5	5.4 ± 0.5	147.6 ± 8	8.2 ± 0.2
	Tree–tree	2.4 ± 0.5	12.6 ± 1.5	230.4 ± 9	5.7 ± 0.3
	Polytree–polytree	13.2 ± 4.0	19.8 ± 3.2	$(5.27 \pm 8.10) \times 10^5$	5.7 ± 0.3
	DAG–DAG	2.4 ± 0.5	9.0 ± 0.6	207.6 ± 6	11.1 ± 0.7
	Small–tw	2.4 ± 0.5	2.4 ± 0.5	198.8 ± 4	3.3 ± 0.0

- acc$_G$: CB–OBSp was found to be significantly better than small–tw by both procedures.
- log–lik: CB–OBSp was found to be significantly better than small–tw by both procedures, and CB–OBSc was found significantly better than small–tw by Holm's procedure.
- AUC$_{\text{macro}}$: CB–OBSp was found to be significantly better than polytree–polytree by both procedures, and significantly better than small–tw by Holm's procedure.
- AUC$_{\text{micro}}$: CB–OBSp was found to be significantly better than polytree–polytree by both procedures.
- MSE: CB–OBSp was found to be significantly better than polytree–polytree by both procedures, and significantly better than small–tw by Holm's procedure.

Figs. 6 and 7 present graphically the results obtained with Holm's procedure for the experimental results shown in Tables 4 and 5 respectively. In the figures, groups of methods that are not significantly different are connected. We use the graphical representation proposed by Demšar [40].

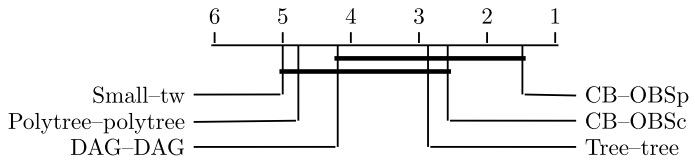
CB–OBSp performed the best in most cases, whereas the performance of CB–OBSc was similar to the performance of the unrestricted methods. The exception is the MEDICAL dataset, where tree–tree and small–tw obtained the best results in terms of both AUC measures. Apparently, models with simpler feature subgraphs discriminated better in this dataset. We can therefore conclude that limiting the number of class variables per component may degrade the fitness of the models compared to directly bounding the treewidth of the pruned graph. Nevertheless, as CB–OBSc yielded results that were similar to the results obtained with the unrestricted methods and given the negligible cost of computing this upper bound, CB–OBSc is an interesting option when the cost of CB–OBSp is too high.

6. Conclusions and future research

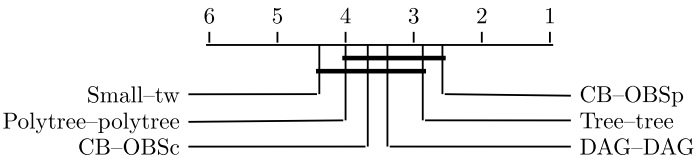
In this paper, we addressed the problem of the complexity of multidimensional classification in MBCs. We provided theoretical upper bounds for the complexity of these models, and we proved that the complexity of the queries that are usually performed in MBCs is bounded by the treewidth of the pruned graph. The treewidth of the pruned graph may be small even if the treewidth of the complete structure is high. We proposed a learning method that uses the above properties to ensure such tractability. We provided two alternatives for bounding the complexity of the methods. Directly bounding the treewidth of the pruned graph achieved a tighter bound, whereas limiting the number of class variables per component is more efficient.

Table 4
Accuracy and fitness comparison.

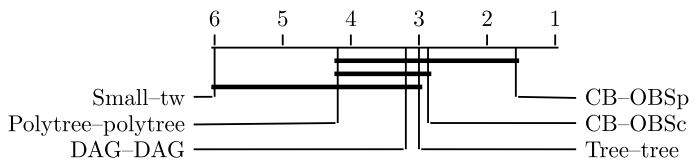
	Method	acc_M	acc_G	log–lik
ANDES	CB–OBSp	$\mathbf{0.821 \pm 0.000}$	0.000 ± 0.000	-708.157 ± 1.645
	CB–OBSc	0.820 ± 0.001	0.000 ± 0.000	-711.223 ± 1.834
	Tree–tree	0.817 ± 0.001	0.000 ± 0.000	-723.911 ± 2.242
	Polytree–polytree	0.812 ± 0.003	0.000 ± 0.000	-734.722 ± 3.528
	DAG–DAG	0.815 ± 0.002	0.000 ± 0.000	$\mathbf{-627.756 \pm 1.470}$
	Small–tw	0.805 ± 0.000	0.000 ± 0.000	-781.528 ± 1.052
MUNIN2	CB–OBSp	$\mathbf{0.794 \pm 0.001}$	0.000 ± 0.000	$\mathbf{-731.219 \pm 0.869}$
	CB–OBSc	0.793 ± 0.000	0.000 ± 0.000	-732.396 ± 0.158
	Tree–tree	0.793 ± 0.000	0.000 ± 0.000	-748.809 ± 1.120
	Polytree–polytree	0.788 ± 0.001	0.000 ± 0.000	-755.402 ± 1.865
	DAG–DAG	0.790 ± 0.001	0.000 ± 0.000	-737.428 ± 2.297
	Small–tw	0.775 ± 0.001	0.000 ± 0.000	-810.261 ± 0.906
PIGS	CB–OBSp	$\mathbf{0.630 \pm 0.003}$	0.000 ± 0.000	$\mathbf{-1239.132 \pm 1.283}$
	CB–OBSc	0.617 ± 0.002	0.000 ± 0.000	-1265.118 ± 1.926
	Tree–tree	–	–	-1243.061 ± 2.366
	Polytree–polytree	–	–	-1266.529 ± 1.410
	DAG–DAG	0.610 ± 0.003	0.000 ± 0.000	-1270.030 ± 3.671
	Small–tw	0.601 ± 0.001	0.000 ± 0.000	-1394.773 ± 1.748
ENRON	CB–OBSp	0.946 ± 0.007	$\mathbf{0.146 \pm 0.102}$	$\mathbf{-67.579 \pm 0.038}$
	CB–OBSc	0.943 ± 0.007	0.086 ± 0.042	-68.974 ± 0.038
	Tree–tree	$\mathbf{0.947 \pm 0.006}$	0.086 ± 0.031	-69.610 ± 0.035
	Polytree–polytree	0.946 ± 0.007	0.088 ± 0.032	-70.855 ± 0.037
	DAG–DAG	0.941 ± 0.007	0.130 ± 0.100	-68.595 ± 0.039
	Small–tw	0.944 ± 0.009	0.075 ± 0.044	-78.582 ± 0.042
MEDICAL	CB–OBSp	$\mathbf{0.988 \pm 0.002}$	$\mathbf{0.636 \pm 0.048}$	-15.410 ± 0.021
	CB–OBSc	0.988 ± 0.002	0.624 ± 0.049	-15.435 ± 0.020
	Tree–tree	0.988 ± 0.002	0.627 ± 0.045	$\mathbf{-14.832 \pm 0.018}$
	Polytree–polytree	–	–	-14.916 ± 0.012
	DAG–DAG	0.987 ± 0.002	0.617 ± 0.050	-15.564 ± 0.021
	Small–tw	0.986 ± 0.002	0.589 ± 0.045	-17.087 ± 0.022



(a) Comparison of acc_M with the Holm's test



(b) Comparison of acc_G with the Holm's test

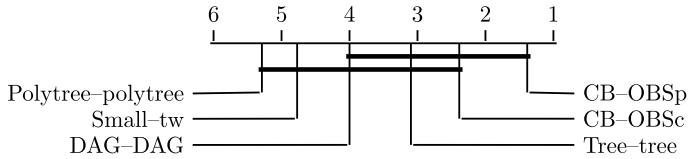


(c) Comparison of log–lik with the Holm's test

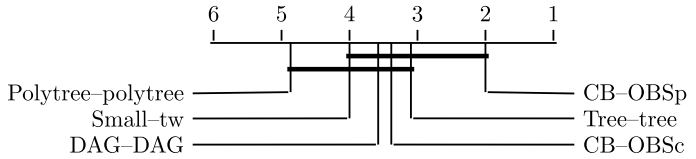
Fig. 6. Comparison of all classifiers against each other with the Holm's test for the experimental results shown in Table 4.

Table 5
AUC and MSE comparison. Note that marginals were only compared in datasets generated from BNs (i.e., ANDES, MUNIN1, and PIGS).

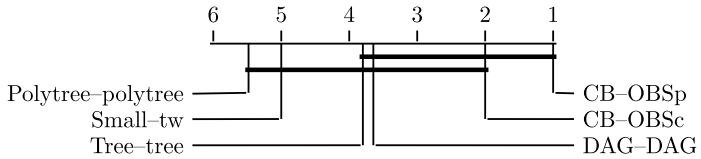
	Method	AUC_{macro}	AUC_{micro}	MSE
ANDES	CB–OBSp	$\mathbf{0.907 \pm 0.000}$	$\mathbf{0.834 \pm 0.001}$	$\mathbf{0.004 \pm 0.000}$
	CB–OBSc	0.906 ± 0.001	0.831 ± 0.002	0.005 ± 0.000
	Tree–tree	0.902 ± 0.001	0.827 ± 0.001	0.007 ± 0.000
	Polytree–polytree	0.896 ± 0.001	0.820 ± 0.001	0.013 ± 0.000
	DAG–DAG	0.900 ± 0.001	0.829 ± 0.002	0.009 ± 0.000
	Small–tw	0.886 ± 0.000	0.810 ± 0.001	0.017 ± 0.000
MUNIN1	CB–OBSp	$\mathbf{0.893 \pm 0.000}$	$\mathbf{0.812 \pm 0.001}$	$\mathbf{0.003 \pm 0.000}$
	CB–OBSc	0.892 ± 0.001	0.810 ± 0.001	0.004 ± 0.000
	Tree–tree	0.887 ± 0.000	0.804 ± 0.001	0.006 ± 0.000
	Polytree–polytree	0.884 ± 0.001	0.800 ± 0.002	0.009 ± 0.001
	DAG–DAG	0.887 ± 0.001	0.807 ± 0.001	0.007 ± 0.000
	Small–tw	0.868 ± 0.001	0.784 ± 0.003	0.017 ± 0.000
PIGS	CB–OBSp	$\mathbf{0.836 \pm 0.001}$	$\mathbf{0.815 \pm 0.001}$	$\mathbf{0.008 \pm 0.000}$
	CB–OBSc	0.821 ± 0.001	0.800 ± 0.001	0.014 ± 0.000
	Tree–tree	–	–	–
	Polytree–polytree	–	–	–
	DAG–DAG	0.818 ± 0.000	0.801 ± 0.000	0.016 ± 0.000
	Small–tw	0.805 ± 0.001	0.791 ± 0.002	0.022 ± 0.001
ENRON	CB–OBSp	$\mathbf{0.928 \pm 0.003}$	0.795 ± 0.008	–
	CB–OBSc	0.927 ± 0.005	0.782 ± 0.008	–
	Tree–tree	0.925 ± 0.005	$\mathbf{0.805 \pm 0.002}$	–
	Polytree–polytree	0.923 ± 0.007	0.798 ± 0.008	–
	DAG–DAG	0.924 ± 0.003	0.785 ± 0.004	–
	Small–tw	0.918 ± 0.008	0.804 ± 0.005	–
MEDICAL	CB–OBSp	0.979 ± 0.003	0.929 ± 0.009	–
	CB–OBSc	0.977 ± 0.003	0.926 ± 0.008	–
	Tree–tree	$\mathbf{0.980 \pm 0.002}$	0.942 ± 0.008	–
	Polytree–polytree	–	–	–
	DAG–DAG	0.975 ± 0.004	0.912 ± 0.004	–
	Small–tw	0.979 ± 0.003	$\mathbf{0.942 \pm 0.010}$	–



(a) Comparison of AUC_{micro} with the Holm's test



(b) Comparison of AUC_{macro} with the Holm's test



(c) Comparison of MSE with the Holm's test

Fig. 7. Comparison of all classifiers against each other with the Holm's test for the experimental results shown in Table 5.

Experimental results showed that the proposed method is competitive with other state-of-the-art methods in terms of accuracy, also ensuring that the learned MBCs can be solved efficiently. We also observed that some models remain tractable even with a large treewidth.

The upper bound provided by the number of class variables per component has the advantage of being able to be computed without increasing the computational cost of the learning process. However, there are MBCs that have a pruned graph with low treewidth and also have components with a high number of class variables. Thus, forcing the CB-decomposability of the models could lead to the rejection of some tractable models during the learning process. Although it is less expensive to calculate the treewidth of the pruned graph than the treewidth of the complete graph, its computational cost may be high. We are currently working on a method that searches in the space of elimination orders to learn bounded treewidth BNs. We intend to adapt this approach to learn MBCs where the treewidth of the pruned graph is efficiently bounded.

Most available scoring functions for learning MBCs are generative with the exception of wrapper methods, that are extremely computationally demanding when the number of variables in the network is high. Bayesian network classifiers that were learned using approximations of the conditional log-likelihood have shown a good performance compared to other Bayesian network classifiers learned using generative metrics [41,42]. It would be very interesting to extend these strategies to multidimensional classification and MBCs.

Also, we are interested in addressing the problem of the complexity of MPE computations in MBCs where there are uninstantiated feature variables.

Finally, one of the main problems with models with latent variables is that exact inference usually has to be performed during the learning process to complete the values of the hidden variables (e.g., structural expectation-maximization). We are interested in adapting the ideas described here to reduce the learning complexity of these models without restricting their structure to trees or polytrees.

Code availability Source code is available upon request from the authors.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the Cajal Blue Brain (C080020-09; the Spanish partner of the Blue Brain initiative from EPFL) and TIN2016-79684-P projects, by the Regional Government of Madrid through the S2013/ICE-2845-CASI-CAM-CM project, and by Fundación BBVA grants to Scientific Research Teams in Big Data 2016. This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 720270 (HBP SGA1). M. Benjumeda is supported by a predoctoral contract for the formation of doctors from the Spanish Ministry of Economy and Competitiveness (BES-2014-068637).

References

[1] C. Bielza, P. Larrañaga, Discrete Bayesian network classifiers: a survey, ACM Comput. Surv. 47 (1) (2014) 5.
[2] L.C. van der Gaag, P.R. de Waal, Multi-dimensional Bayesian network classifiers, in: Proceedings of the 3rd European Workshop on Probabilistic Graphical Models, 2006, pp. 107–114.
[3] C. Bielza, G. Li, P. Larrañaga, Multi-dimensional classification with Bayesian networks, Int. J. Approx. Reason. 52 (6) (2011) 705–727.
[4] M. Benjumeda, C. Bielza, P. Larrañaga, Learning tractable multidimensional Bayesian network classifiers, in: Proceedings of the 8th International Conference on Probabilistic Graphical Models, vol. 52, 2016, pp. 25–32.
[5] J.D. Park, MAP complexity results and approximation methods, in: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 2002, pp. 388–396.
[6] J. Kwisthout, Most probable explanations in Bayesian networks: complexity and tractability, Int. J. Approx. Reason. 52 (9) (2011) 1452–1469.
[7] B.K. Sy, Reasoning MPE to multiply connected belief networks using message passing, in: Proceedings of the 10th National Conference on Artificial Intelligence, AAAI Press, 1992, pp. 570–576.
[8] P.R. de Waal, L.C. van der Gaag, Inference and learning in multi-dimensional Bayesian network classifiers, in: Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Springer, 2007, pp. 501–511.
[9] A. Pastink, L.C. van der Gaag, Multi-classifiers of small treewidth, in: Proceedings of the 13th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 2015, pp. 199–209.
[10] G. Corani, A. Antonucci, D.D. Mauá, S. Gabaglio, Trading off speed and accuracy in multilabel classification, in: Proceedings of the 7th European Workshop on Probabilistic Graphical Models, 2014, pp. 145–159.
[11] H. Borchani, C. Bielza, P. Larrañaga, Learning CB-decomposable multi-dimensional Bayesian network classifiers, in: Proceedings of the 5th European Workshop on Probabilistic Graphical Models, 2010, pp. 25–32.
[12] N. Ghamrawi, A. McCallum, Collective multi-label classification, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, ACM, 2005, pp. 195–200.
[13] J. Arias, J.A. Gamez, T.D. Nielsen, J.M. Puerta, A scalable pairwise class interaction framework for multidimensional classification, Int. J. Approx. Reason. 68 (2016) 194–210.
[14] S. Arnborg, D.G. Corneil, A. Proskurowski, Complexity of finding embeddings in a k-tree, SIAM J. Algebraic Discrete Methods 8 (2) (1987) 277–284.
[15] F.V. Fomin, Y. Villanger, Treewidth computation and extremal combinatorics, Combinatorica 32 (3) (2012) 289–308.
[16] H.L. Bodlaender, F.V. Fomin, A.M.C.A. Koster, D. Kratsch, D.M. Thilikos, On exact algorithms for treewidth, ACM Trans. Algorithms 9 (1) (2012) 12–23.
[17] H.L. Bodlaender, A linear time algorithm for finding tree-decompositions of small treewidth, in: Proceedings of the 25th Annual ACM Symposium on Theory of Computing, ACM, 1993, pp. 226–234.
[18] H.M. Markowitz, The elimination form of the inverse and its application to linear programming, Manag. Sci. 3 (3) (1957) 255–269.
[19] U.B. Kjærulff, Triangulation of Graphs-Algorithms Giving Small Total State Space, Tech. Rep. 1990.
[20] D.J. Rose, R.E. Tarjan, G.S. Lueker, Algorithmic aspects of vertex elimination on graphs, SIAM J. Comput. 5 (2) (1976) 266–283.

[21] A. Berry, J.R. Blair, P. Heggernes, B.W. Peyton, Maximum cardinality search for computing minimal triangulations of graphs, Algorithmica 39 (4) (2004) 287–298.
[22] F. Clautiaux, A. Moukrim, S. Nègre, J. Carlier, Heuristic and metaheuristic methods for computing graph treewidth, RAIRO. Rech. Opér. 38 (1) (2004) 13–26.
[23] U. Kjærulff, Optimal decomposition of probabilistic networks by simulated annealing, Stat. Comput. 2 (1) (1992) 7–17.
[24] P. Larrañaga, C.M. Kuijpers, M. Poza, R.H. Murga, Decomposing Bayesian networks: triangulation of the moral graph with genetic algorithms, Stat. Comput. 7 (1) (1997) 19–34.
[25] F.R. Bach, M.I. Jordan, Thin junction trees, in: Advances in Neural Information Processing Systems, 2001, pp. 569–576.
[26] A. Chechetka, C. Guestrin, Efficient principled learning of thin junction trees, in: Advances in Neural Information Processing Systems, 2008, pp. 273–280.
[27] G. Elidan, S. Gould, Learning bounded treewidth Bayesian networks, in: Advances in Neural Information Processing Systems, 2009, pp. 417–424.
[28] R.R. Bouckaert, Optimizing causal orderings for generating DAGs from data, in: Proceedings of the 8th International Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 1992, pp. 9–16.
[29] M. Teyssier, D. Koller, Ordering-based search: a simple and effective algorithm for learning Bayesian networks, in: Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2005, pp. 584–590.
[30] C. Conati, A.S. Gertner, K. VanLehn, M.J. Druzdzel, On-line student modeling for coached problem solving using Bayesian networks, in: Proceedings of the 6th International Conference on User Modeling, Springer, 1997, pp. 231–242.
[31] S. Andreassen, F.V. Jensen, S.K. Andersen, B. Falck, U.B. Kjærulff, M. Woldbye, A.R. Sørensen, A. Rosenfalck, F. Jensen, MUNIN – an expert EMG assistant, in: Computer-Aided Electromyography and Expert Systems, vol. 21, 1989, pp. 247–256.
[32] C.S. Jensen, U. Kjærulff, A. Kong, Blocking Gibbs sampling in very large probabilistic expert systems, Int. J. Hum.-Comput. Stud. 42 (6) (1995) 647–666.
[33] B. Klimt, Y. Yang, The Enron corpus: a new dataset for email classification research, in: Proceedings of the 15th European Conference on Machine Learning, Springer, 2004, pp. 217–226.
[34] J.P. Pestian, C. Brew, P. Matykiewicz, D.J. Hovermale, N. Johnson, K.B. Cohen, W. Duch, A shared task involving multi-label classification of clinical free text, in: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, Association for Computational Linguistics, 2007, pp. 97–104.
[35] F. Provost, P. Domingos, Improving probability estimation trees, Mach. Learn. 52 (3) (2000) 199–215.
[36] M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2014) 1819–1837.
[37] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. (1979) 65–70.
[38] J.P. Shaffer, Modified sequentially rejective multiple test procedures, J. Am. Stat. Assoc. 81 (395) (1986) 826–831.
[39] S. Garcia, F. Herrera, An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons, J. Mach. Learn. Res. 9 (Dec. 2008) 2677–2694.
[40] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[41] A.M. Carvalho, T. Roos, A.L. Oliveira, P. Myllymäki, Discriminative learning of Bayesian networks via factorized conditional log-likelihood, J. Mach. Learn. Res. 12 (Jul. 2011) 2181–2210.
[42] D.J. Grossman, P.M. Del Domingos, Learning Bayesian network classifiers by maximizing conditional likelihood, in: Proceedings, 21st International Conference on Machine Learning, ICML 2004, 2004, p. 46.