# Multi-dimensional Bayesian Network Classifier Trees

Santiago Gil-Begue(✉), Pedro Larrañaga, and Concha Bielza

Universidad Politécnica de Madrid, Boadilla del Monte, Madrid, Spain
{sgil,pedro.larranaga,mcbielza}@fi.upm.es

**Abstract.** Multi-dimensional Bayesian network classifiers (MBCs) are probabilistic graphical models tailored to solving multi-dimensional classification problems, where an instance has to be assigned to multiple class variables. In this paper, we propose a novel multi-dimensional classifier that consists of a classification tree with MBCs in the leaves. We present a wrapper approach for learning this classifier from data. An experimental study carried out on randomly generated synthetic data sets shows encouraging results in terms of predictive accuracy.

## 1 Introduction

In this paper we are interested in classification problems where there are multiple class variables. Multi-dimensional Bayesian network classifiers (MBCs) are probabilistic graphical models tailored to solving this kind of classification problem, which arises in many application domains. For example, MBCs have been used in the literature to estimate the health-related quality of life of Parkinson's disease patients [3], for sentiment analysis [13], to assist in the treatment of multiple sclerosis [16] and to predict human immunodeficiency virus inhibitors [4], among other applications.

Meta-classifiers combine different models before making a final decision motivated by the fact that there is no a learning algorithm that always induces the most accurate classifier [18]. A hybrid classifier is a meta-classifier that is induced taking into account two or more paradigms. Following this idea, we propose a novel multi-dimensional classifier that combines both well-known classification trees and MBCs. To the best of our knowledge, the *multi-dimensional Bayesian network classifier tree* (MBCTree) is the first hybrid model proposed in the context of multi-dimensional classification. We also present a wrapper approach for learning MBCTrees from data. As a wrapper strategy, we review existing performance evaluation measures to assess multi-dimensional classifiers.

The remainder of this article is organized as follows. Section 2 reviews the fundamentals of MBCs. Section 3 describes some performance measures suitable

for evaluating multi-dimensional classifiers. Section 4 describes our new multi-dimensional classifier and a wrapper algorithm for learning this classifier from data. Section 5 contains experimental results with synthetic data sets. Section 6 finally completes the article with a discussion and future work.

## 2 Fundamentals

### 2.1 Multi-dimensional Classification

We are interested in classification problems where there are multiple class variables $C_1, \ldots, C_d$. The *multi-dimensional classification* problem consists of finding a function $h$ that assigns a vector of $d$ class values $\mathbf{c} = (c_1, \ldots, c_d)$ to each instance given by a vector of $m$ features $\mathbf{x} = (x_1, \ldots, x_m)$:

$$\begin{aligned} h : \Omega_{X_1} \times \cdots \times \Omega_{X_m} &\rightarrow \Omega_{C_1} \times \cdots \times \Omega_{C_d} \\ (x_1, \ldots, x_m) &\mapsto (c_1, \ldots, c_d) \end{aligned}$$

We assume that $C_j$ is a discrete variable, for all $j \in \{1, \ldots, d\}$, with $\Omega_{C_j}$ denoting its sample space and $I = \Omega_{C_1} \times \cdots \times \Omega_{C_d}$, the space of joint configurations of the class variables. Analogously, $\Omega_{X_i}$ is the sample space of the discrete feature variable $X_i$, for all $i \in \{1, \ldots, m\}$. When all the classes are binary, i.e., $\left|\Omega_{C_j}\right| = 2$ for all $j \in \{1, \ldots, d\}$, the problem is called *multi-label classification*. Note that multi-label classification is just a particular setting of multi-dimensional classification. There are a lot of contributions to this multi-label paradigm. Two up-to-date reviews of the main proposals presented during the latest years are [7] and [19].

As stated by [2], multi-dimensional classification is a more difficult problem than the best-known single-class case. The main problem is that there is a large number of possible class label combinations, $|I|$, and a usual sparseness of available data. In a typical scenario where an instance $\mathbf{x}$ is assigned to the most likely combination of classes (0−1 loss function), the aim is to compute $\arg\max_{c_1,\ldots,c_d} p(C_1 = c_1, \ldots, C_d = c_d|\mathbf{x})$. It holds that $p(C_1 = c_1, \ldots, C_d = c_d|\mathbf{x}) \propto p(C_1 = c_1, \ldots, C_d = c_d, \mathbf{x})$, which requires $|I| \cdot |\Omega_{X_1} \times \cdots \times \Omega_{X_m}|$ parameters to be assigned. In the single-class, $C$, case, $|I|$ is just $|\Omega_C|$ rather than $|\Omega_{C_1} \times \cdots \times \Omega_{C_d}|$. Besides it having a high cardinality, it is also hard to estimate the required parameters from a (sparse) data set in this $d$-dimensional space $I$. The factorization of this joint probability distribution when using a Bayesian network can somehow reduce this number of parameters, which has been studied with the MBCs.

### 2.2 Multi-dimensional Bayesian Network Classifiers

A Bayesian network [11,14] over a set of discrete random variables $\{Z_1, \ldots, Z_n\}$, $n \geq 1$, is a pair $\mathcal{B} = (G, \boldsymbol{\Theta})$. $G = (V, A)$ is a directed acyclic graph whose vertices $V$ correspond to variables $Z_i$ and whose arcs $A$ represent direct probabilistic dependencies between the vertices. $\boldsymbol{\Theta}$ is a vector of parameters such that $\theta_{z_i|\mathbf{pa}(z_i)} = p(z_i|\mathbf{pa}(z_i))$ defines the conditional probability of each possible value

$z_i$ of $Z_i$ given a vector value $\mathbf{pa}(z_i)$ of the parents of $Z_i$ in $G$. $\mathcal{B}$ represents a joint probability distribution $p_{\mathcal{B}}$ over the set of random variables factorized according to its structure $G$:

$$p_{\mathcal{B}}(z_1, \ldots, z_n) = \prod_{i=1}^{n} p(z_i|\mathbf{pa}(z_i)).$$

Bayesian network classifiers [1] are Bayesian networks of restricted topology tailored to solving classification problems in which instances described by a number of features have to be classified in one of several distinct predefined classes. The finite set of vertices $V$ of a Bayesian network classifier is partitioned into a set $V_X = \{X_1, \ldots, X_m\}$, $m \geq 1$, of feature variables and a singleton set $V_C = \{C\}$ that corresponds to the class variable (i.e., $n = m + 1$).

An MBC is a Bayesian network specially designed to solve multi-dimensional classification problems. The graph $G = (V, A)$ of an MBC has the set $V$ of vertices also partitioned into two sets $V_C = \{C_1, \ldots, C_d\}$, $d \geq 1$, of class variables and $V_X = \{X_1, \ldots, X_m\}$, $m \geq 1$, of feature variables (i.e., $n = m + d$). Note that Bayesian network classifiers are a particular setting ($d = 1$) of MBCs. The graph has also a restricted topology in which the set of arcs $A$ is partitioned into three sets $A_C$, $A_X$ and $A_{CX}$. The first time MBCs were proposed by [6], the three sets of arcs had the following properties:

1. The set $A_C \subseteq V_C \times V_C$ is composed of the arcs between the class variables having a subgraph $G_C = (V_C, A_C)$, *class subgraph*, of $G$ induced by $V_C$;
2. The set $A_X \subseteq V_X \times V_X$ is composed of the arcs between the feature variables having a subgraph $G_X = (V_X, A_X)$, *feature subgraph*, of $G$ induced by $V_X$;
3. The set $A_{CX} \subseteq V_C \times V_X$ is composed of the arcs from the class variables to the feature variables having a subgraph $G_{CX} = (V, A_{CX})$, *feature selection subgraph*, of $G$ induced by $V$, such that for each $X_i \in V_X$, there is a $C_j \in V_C$ with the arc $(C_j, X_i) \in A_{CX}$ and for each $C_j \in V_C$, there is an $X_i \in V_X$ with the arc $(C_j, X_i) \in A_{CX}$.

MBCs were later extended by [2], such that the two conditions of the set of arcs $A_{CX}$ were removed, and the resulting subgraph was renamed to another term:

3. The set $A_{CX} \subseteq V_C \times V_X$ is composed of the arcs from the class variables to the feature variables having a subgraph $G_{CX} = (V, A_{CX})$, *bridge subgraph*, of $G$ induced by $V$.

This last definition has been mainly adopted in the literature. Figure 1 shows an example of an MBC structure and its three subgraphs. Note that the initial definition by [6] does not recognize this structure as an MBC, because for $X_3 \in V_X$, there is no $C_j \in V_C$ with $(C_j, X_3) \in A_{CX}$. The extension of [2] can be seen as a more general definition.

## 3 Performance Evaluation Measures for Multi-dimensional Classifiers

The evaluation of models in a multi-dimensional context needs a special approach because the simultaneous performance over all class variables should be taken

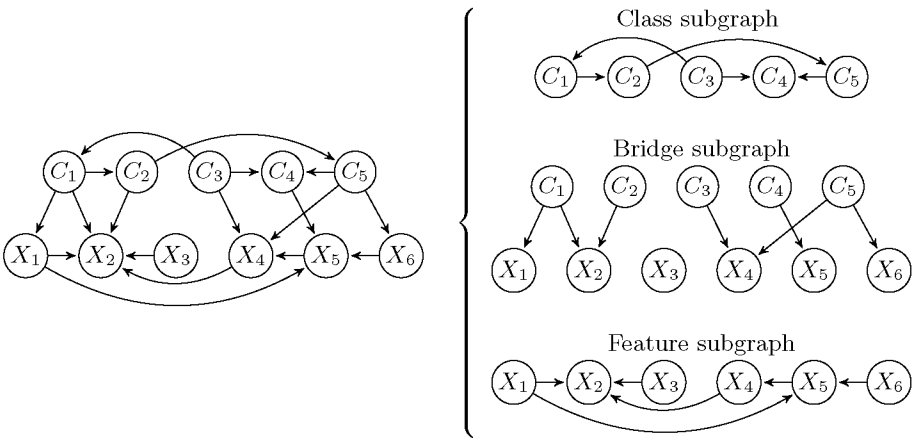


**Fig. 1.** An example of an MBC structure with its three subgraphs.

into account. Several performance evaluation measures have been extended to the particular multi-label setting, but only few extensions to the more general multi-dimensional classification problem are found in the literature. The most frequent measures for multi-label classification are summarized in [7].

[2] proposed the following multi-dimensional performance measures that extend those in the multi-label domain:

- *Global* or *joint accuracy* over the $d$-dimensional class variable, which extends the *multi-label 0/1 subset accuracy* [21] by computing the fraction of correctly classified examples, i.e., those whose all predicted class values are exactly the same as their corresponding true values. It is a very strict evaluation measure, especially when the size of the class space, $|I|$, is large. Let $\mathbf{c}'_i$ be the $d$-dimensional binary prediction for case $i$ in the test data set of $N$ cases, $\mathbf{c}_i$ its corresponding true value, and $\delta(\mathbf{c}'_i, \mathbf{c}_i) = 1$ if $\mathbf{c}'_i = \mathbf{c}_i$ and 0 otherwise, then the global accuracy is defined as:

$$Acc = \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{c}'_i, \mathbf{c}_i). \tag{1}$$

- *Mean* or *average accuracy* over the $d$ class variables, which evaluates the fraction of correctly classified example-class pairs. Let $c'_{ij}$ be the $C_j$ class value predicted by the model for case $i$ in the test data set, $c_{ij}$ its corresponding true value, and $\delta(c'_{ij}, c_{ij}) = 1$ if $c'_{ij} = c_{ij}$ and 0 otherwise. Then, the mean accuracy is defined as:

$$\overline{Acc} = \frac{1}{d} \sum_{j=1}^{d} Acc_j = \frac{1}{d} \sum_{j=1}^{d} \frac{1}{N} \sum_{i=1}^{N} \delta(c'_{ij}, c_{ij}).$$

This measure is the complementary of the multi-label *Hamming loss* [17], i.e., $mean\ accuracy + Hamming\ loss = 1$, but extended to the multi-dimensional paradigm.

## 4 Multi-dimensional Bayesian Network Classifier Trees

Meta-classifiers combine different models motivated by the *no free-lunch theorem* [18], which states that there is no a learning algorithm that in any domain always induces the most accurate classifier. A hybrid classifier is a meta-classifier that is induced taking into account two or more paradigms. An example of a hybrid classifier is the naive Bayes tree (NBTree) of [10], which deploys a naive Bayes model on each leaf node of a classification tree. Following a similar idea, [12] proposed the logistic model tree, with logistic regressions instead of naive Bayes classifiers in the leaves of the classification tree. Another example is the lazy Bayesian rules proposed by [20], which builds a most appropriate rule for each test instance with a local naive Bayesian classifier as its consequent.

In this paper we propose a hybrid of *classification trees* [5] and MBCs. To the best of our knowledge, this is the first hybrid model proposed in the context of multi-dimensional classification. An MBCTree is a classification tree with MBCs in the leaves (Fig. 2):

- An internal node of an MBCTree corresponds to a feature variable $X_i$ as in standard classification trees, and has a labelled branch to a child for each of its possible values in $\Omega_{X_i}$. The labels are the possible values of $X_i$.
- A leaf node of an MBCTree corresponds to an MBC over all the class variables and those feature variables not present in the path from the root to the leaf. An MBCTree may be asymmetric, so the MBCs at the leaves may have different feature variables. A new instance will be classified by sorting down the tree from the root to some MBC leaf node according to the outcome of the tests along the path.

### 4.1 A Wrapper Approach for Learning MBCTrees from Data

We propose a wrapper approach guided by the global accuracy (Eq. (1)) for learning MBCTrees from data, although any other multi-dimensional measure could be used. The proposed approach is detailed in Algorithm 1.

The MBCTree is learned by recursively choosing as internal node the variable $X_{best}$ that best splits the data, i.e., that achieves the highest global accuracy [steps 1–11], until splitting no longer adds value to the predictions [steps 12–14]. The proposed approach is seen as a greedy top-down algorithm [15]. It starts at the root node of the tree and computes the global accuracy, $Acc_{MBCTree_i}$, of each feature variable $X_i$ to split on [steps 2–8]. For this, an MBC is learned for each possible value of $X_i$ with the corresponding portion of data [steps 3–6]. The accuracy of the split is computed by sorting the test data set to the learned

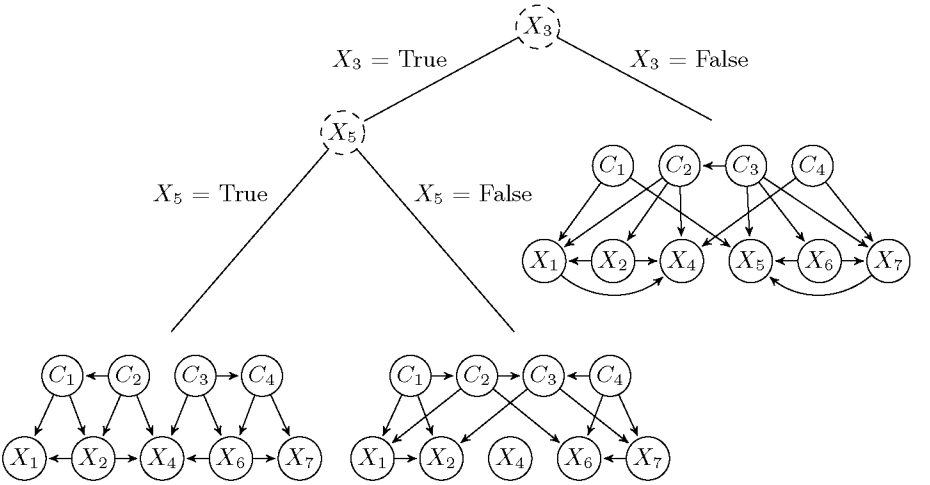


**Fig. 2.** An example of an MBCTree structure.

MBCs testing $X_i$ [step 7]. We have chosen a wrapper strategy guided by the global accuracy to learn the MBCs as in [2]. This process is repeated on each derived subset of the best split in a recursive manner [step 11], until there is no split that improves the global accuracy achieved by an MBC learned with the data that reach the current node [step 1]. In that case, this MBC is placed as a leaf node [step 13]. A leaf node must be also created if there is only one feature variable left to split on, as there could not be an MBC at the leaf with no feature variables in its structure, and also if there is no enough data to keep growing the tree, i.e., to learn split MBCs. To avoid overfitting the training set, at least a minimum improvement can be required in [step 10] as a prune strategy, such that the recursion ends in [step 13] if no significant improvement is achieved.

## 5 Experimental Results

In order to evaluate our proposed approach we performed an experimental study on synthetic data sets. For the benefit of the community, we make the source code public[1]. The study follows the steps shown in Fig. 3:

- First, a random MBCTree with fixed depth is generated. The feature variables associated to the internal nodes are randomly chosen. The MBCs leaf are also randomly generated, such that the class and feature subgraphs are uniformly distributed samples of directed acyclic graphs [9] and each class variable $C_j$ is connected to each feature variable $X_i$ with a probability $p$. We chose $p = 0.5$ in order to uniformly sample from the MBC structure space. The parameters of MBCs are forced to be extreme, i.e., lower than 0.3 and greater than 0.7.

[1] Code available at https://github.com/ComputationalIntelligenceGroup/MBCTree.

**Algorithm 1.** Wrapper algorithm for learning MBCTrees from data.

Input: A labelled data set $D$
Output: An MBCTree learned with the data set $D$

1: Learn an MBC and compute its global accuracy, $Acc_{MBC}$, with the data set $D$
2: **for each** feature variable $X_i$ **do**
3: Split $D$ into $|\Omega_{X_i}|$ subsets $D_{ij}$ based on the possible values $j$ of $X_i$
4: **for each** subset $D_{ij}$ **do**
5: Learn an $\text{MBC}_{ij}$ with the subset $D_{ij}$
6: **end for**
7: Compute the global accuracy, $Acc_{MBCTree_i}$, of the split on $X_i$
8: **end for**
9: $best = \arg\max_i Acc_{MBCTree_i}$
10: **if** $Acc_{MBCTree_{best}} > Acc_{MBC}$ **then**
11: Create an internal node $X_{best}$. For each child node under a branch with label $j \in \Omega_{X_{best}}$, call the algorithm recursively, from step 1, on the data subset $D_{best\ j}$
12: **else**
13: Create a leaf node with the MBC of step 1
14: **end if**

- Second, a data set is simulated from the MBCTree. For this, a data subset of random size is simulated for each MBC leaf by using probabilistic logic sampling [8]. It is imposed that each subset contributes at least a fixed percentage to the whole data set.
- Third, an MBC and an MBCTree are learned following the wrapper approach in [2] and Algorithm 1, respectively. Then, they are compared in terms of predictive accuracy with the simulated data set, which is divided in a training set containing 80% of the instances and a test set with 20%. The training set is also divided in the same percentages for learning the MBCTree: 80% of the instances are used for learning the MBCs and 20% for evaluating them so that the best split can be computed. Cross-validation could have been used in both cases, but we followed a train and test strategy together with a larger simulated data set because of computational efficiency.

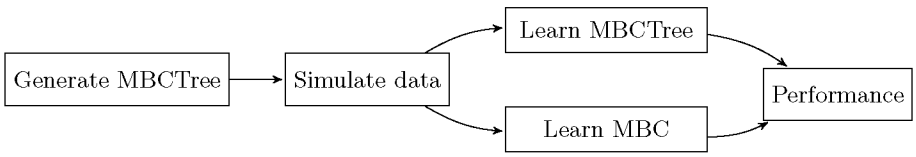


**Fig. 3.** Steps of the experimental study performed with synthetic data.

We established two different configurations following the aforementioned scheme. Each configuration was executed ten times, leading to the results shown in Table 1.

- Configuration A: an MBCTree of depth 1 with 11 feature variables in the MBC leaf (11 + 1 features altogether) and 4 class variables to be predicted. All the variables are binary. The simulated data set consists of 100,000 instances, with at least 20% in each branch.
- Configuration B: an MBCTree of depth 2 with 10 feature variables in the MBC leaf (10 + 2 features altogether) and 4 class variables to be predicted. All the variables are binary. The simulated data set consists of 100,000 instances, with at least 20% in each branch in a recursive manner.

**Table 1.** Comparison in terms of global accuracy of the MBCs and MBCTrees.

| Configuration A | | | | Configuration B | | | |
|---|---|---|---|---|---|---|---|
| MBC | MBCTree | Diff. | Learned | MBC | MBCTree | Diff. | Learned |
| 0.6649 | 0.6941 | +0.0292 | 101 | 0.7903 | 0.8113 | +0.0210 | 177 |
| 0.7719 | 0.7972 | +0.0253 | 101 | 0.7696 | 0.8005 | +0.0309 | 139 |
| 0.7898 | 0.8053 | +0.0155 | 59 | 0.7444 | 0.7866 | +0.0422 | 177 |
| 0.8098 | 0.8216 | +0.0118 | 37 | 0.7876 | 0.8183 | +0.0307 | 101 |
| 0.7280 | 0.7382 | +0.0102 | 79 | 0.8422 | 0.8692 | +0.0270 | 157 |
| 0.7796 | 0.8070 | +0.0274 | 81 | 0.7892 | 0.8350 | +0.0458 | 179 |
| 0.8461 | 0.8630 | +0.0169 | 59 | 0.7772 | 0.7948 | +0.0176 | 159 |
| 0.8584 | 0.8664 | +0.0080 | 37 | 0.7764 | 0.8066 | +0.0302 | 141 |
| 0.8367 | 0.8474 | +0.0107 | 121 | 0.7051 | 0.7383 | +0.0332 | 231 |
| 0.8314 | 0.8577 | +0.0264 | 37 | 0.7352 | 0.7762 | +0.0410 | 177 |
| *Average* | | **+0.0181** | **71** | *Average* | | **+0.0320** | **164** |

As an encouraging result of the experimental study performed, our MBCTree model has achieved higher global accuracies than MBCs in all the executions for both configurations. Similar profits were obtained by the NBTree with respect to standard naive Bayes classifiers [10]. The improvement is more noticeable in Configuration B executions because the simulated data set comes from more different probability distributions, and the MBCTree is able to discover this data partition. Following this idea, using a configuration C with a random MBCTree of depth 3 has improved the results even more, with an average improvement of 0.0364 over ten executions. In all three cases, the accuracy improvement is statistically significant with a $p$-value = 0.001953 when using the Wilcoxon signed-rank test.

As a positive aspect of the proposed learning approach, all internal nodes of an initial randomly generated MBCTree have been always recovered following the same structure in the learned MBCTree, for both Configurations A and B. In addition, the algorithm has often included other extra internal nodes besides this main structure. Most internal nodes in Configuration C have been also recovered, but in a different order. For example, the root node of the initial model has been sometimes included at the second level. This is explained because of the greedy nature of the proposed learning algorithm.

In contrast, the computational burden for learning an MBCTree has to be remarked. The critical point is to compute at each node which feature variable best splits the data, as an MBC is learned for each possible value of each feature variable. On average, 71 and 164 MBCs have been needed to be learned for inducing the MBCTrees for Configuration A and B, respectively. Our method is then adding two orders of magnitude in this particular problem. This complexity could be alleviated if the MBCs were learned in parallel. Another drawback of our model is that an MBCTree needs a larger training data set because the recursive partitioning of the data makes the MBCs leaf to be learned with fewer data.

## 6 Conclusions and Future Research

In this paper we have proposed a novel multi-dimensional classifier that consists of a classification tree with MBCs in the leaves. An improvement in terms of predictive accuracy has been shown on randomly generated synthetic data sets, but at the expense of adding more complexity to the model learning.

As future work, we would intend to carry out a more extensive experimental study using real data sets. We have seen that learning MBCTrees is high data demanding, so a thoughtful choice of a real multi-dimensional problem is required. Moreover, it will be interesting to extend MBCTrees to deal with the challenging task of multi-dimensional classification for concept-drifting data streams. Finally, we would like to alleviate the computational burden of our proposed wrapper algorithm, for what we plan (1) to learn the MBCs for each feature value in parallel since they are independent processes, as well as to execute a parallel recursion for the branches of a split node, (2) to use a filter learning technique, and also (3) to investigate a model based on the random selection of the features to split on, within an ensemble approach.

**Acknowledgements.** This work has been partially supported by the Spanish Ministry of Economy, Industry 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.

## References

1. Bielza, C., Larrañaga, P.: Discrete Bayesian network classifiers: A survey. ACM Comput. Surv. **47**(1), 5 (2014). https://doi.org/10.1145/2576868
2. Bielza, C., Li, G., Larrañaga, P.: Multi-dimensional classification with Bayesian networks. Int. J. Approx. Reason. **52**(6), 705–727 (2011). https://doi.org/10.1016/j.ijar.2011.01.007
3. Borchani, H., Bielza, C., Martínez-Martí, P., Larrañaga, P.: Markov blanket-based approach for learning multi-dimensional Bayesian network classifiers: An application to predict the European quality of life-5 dimensions (EQ-5D) from the 39-item Parkinson's disease questionnaire (PDQ-39). J. Biomed. Inform. **45**(6), 1175–1184 (2012). https://doi.org/10.1016/j.jbi.2012.07.010

4. Borchani, H., Bielza, C., Toro, C., Larrañaga, P.: Predicting human immunodeficiency virus inhibitors using multi-dimensional Bayesian network classifiers. Artif. Intell. Med. **57**(3), 219–229 (2013). https://doi.org/10.1016/j.artmed.2012.12.005
5. Breiman, L., Friedman, J.H., Olshen, R., Stone, C.J.: Classification and Regression Trees. Wadsworth, California (1984)
6. van der Gaag, L.C., de Waal, P.R.: Muti-dimensional Bayesian network classifiers. In: Proceedings of the 3rd European Workshop in Probabilistic Graphical Models, pp. 107–114 (2006)
7. Gibaja, E., Ventura, S.: A tutorial on multi-label learning. ACM Comput. Surv. **47**(3), 52 (2015). https://doi.org/10.1145/2716262
8. Henrion, M.: Propagating uncertainty in Bayesian networks by probabilistic logic sampling. Mach. Intell. Pattern Recognit. **5**, 149–163 (1988). https://doi.org/10.1016/B978-0-444-70396-5.50019-4
9. Ide, J.S., Cozman, F.G.: Random generation of Bayesian networks. In: Bittencourt, G., Ramalho, G.L. (eds.) SBIA 2002. LNCS (LNAI), vol. 2507, pp. 366–376. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36127-8_35
10. Kohavi, R.: Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, vol. 96, pp. 202–207 (1996)
11. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. The MIT Press, Cambridge (2009)
12. Landwehr, N., Hall, M., Frank, E.: Logistic model trees. Mach. Learn. **59**(1–2), 161–205 (2005). https://doi.org/10.1007/s10994-005-0466-3
13. Ortigosa-Hernández, J., Rodríguez, J.D., Alzate, L., Lucania, M., Inza, I., Lozano, J.A.: Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers. Neurocomputing **92**, 98–115 (2012). https://doi.org/10.1016/j.neucom.2012.01.030
14. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Burlington (1988)
15. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986). https://doi.org/10.1023/A:1022643204877
16. Rodríguez, J.D., Pérez, A., Arteta, D., Tejedor, D., Lozano, J.A.: Using multidimensional Bayesian network classifiers to assist the treatment of multiple sclerosis. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **42**(6), 1705–1715 (2012). https://doi.org/10.1109/TSMCC.2012.2217326
17. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Mach. Learn. **37**(3), 297–336 (1999). https://doi.org/10.1023/A:1007614523901
18. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997). https://doi.org/10.1109/4235.585893
19. Zhang, M., Zhou, Z.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. **26**(8), 1819–1837 (2014). https://doi.org/10.1109/TKDE.2013.39
20. Zheng, Z., Webb, G.I.: Lazy learning of Bayesian rules. Mach. Learn. **41**(1), 53–84 (2000). https://doi.org/10.1023/A:1007613203719
21. Zhu, S., Ji, X., Xu, W., Gong, Y.: Multi-labelled classification using maximum entropy method. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 274–281. ACM (2005). https://doi.org/10.1145/1076034.1076082