Check for updates

Received: 30 April 2021 | Revised: 18 July 2021 | Accepted: 29 July 2021

DOI: 10.1002/int.22611

RESEARCH ARTICLE

WILEY

Multidimensional continuous time Bayesian network classifiers

Carlos Villa-Blanco | **Pedro Larrañaga** | **Concha Bielza**

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

Correspondence
Carlos Villa-Blanco, Computational Intelligence Group, Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Boadilla del Monte, Madrid 28660, Spain.
Email: carlos.villa@upm.es

Funding information
Universidad Politécnica de Madrid, Grant/Award Number: Predoctoral contract for the formation of doctors; Ministerio de Ciencia, Innovación y Universidades, Grant/Award Number: PID2019-109247GB-I00; Ministerio de Ciencia, Innovación y Universidades, Grant/Award Number: RTC2019-006871-7; Fundación BBVA, Grant/Award Number: BAYES-CLIMA-NEURO

Abstract

The multidimensional classification of multivariate time series deals with the assignment of multiple classes to time-ordered data described by a set of feature variables. Although this challenging task has received almost no attention in the literature, it is present in a wide variety of domains, such as medicine, finance or industry. The complexity of this problem lies in two nontrivial tasks, the learning with multivariate time series in continuous time and the simultaneous classification of multiple class variables that may show dependencies between them. These can be addressed with different strategies, but most of them may involve a difficult preprocessing of the data, high space and classification complexity or ignoring useful interclass dependencies. Additionally, no attention has been given to the development of new multidimensional classifiers of time series based on probabilistic graphical models, even though transparent models can facilitate further understanding of the domain. In this paper, a novel probabilistic graphical model is proposed, which is able to classify a discrete multivariate temporal sequence into multiple class variables while modeling their dependencies. This model extends continuous time Bayesian networks to the multidimensional classification problem, which are able to explicitly represent the behavior of time series that

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2021 The Authors. *International Journal of Intelligent Systems* published by Wiley Periodicals LLC

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

evolve over continuous time. Different methods for the learning of the parameters and structure of the model are presented, and numerical experiments on synthetic and real-world data show encouraging results in terms of performance and learning time with respect to independent classifiers, the current alternative approach under the continuous time Bayesian network paradigm.

KEYWORDS

Bayesian network classifiers, learning from data, multidimensional classification, multivariate time series, probabilistic graphical models

1 | INTRODUCTION

Many classification problems imply the analysis of trends or dynamics that occur in sequences of time-ordered data to perform accurate predictions. Typical examples are found in finance, medicine, signal processing or industry, but more applications are emerging in virtually any domain.[1–4] This article focuses on the complex scenario of multidimensional classification over multivariate time series, presenting a novel model for the task and a real-world problem where it is applied.

The multidimensional classification problem deals with the simultaneous classification of multiple class variables, that is, it requires the definition of a mapping function that determines the output of several multiclass class variables based on a given input data. This learning problem is included in the more general multioutput paradigm, which also covers supervised learning problems with outputs of different data types, such as real-valued or ordinal. The reader is referred to the comprehensive review of Xu et al.[5] for a more in-depth reading about the multioutput learning paradigm.

Traditional classification algorithms are limited to the prediction of a unique variable, so they cannot be directly applied in the studied multidimensional context. Two simple approaches are commonly used to avoid this limitation: the definition of a compound class variable that collects all combinations of class values (label powerset method) and the learning of independent classifiers for each class variable (binary relevance method). Nonetheless, both solutions involve a number of drawbacks, being a common inconvenience the impossibility of modeling the dependencies between class variables. This independence assumption can be avoided with methods such as chain classifiers,[6] which iteratively train a set of classifiers (one per class variable), whose feature spaces are extended with the ground-truth classes of their predecessors. However, this approach is really dependent on the order in which classifiers are applied, requiring to explore an intractable number of chain orders to find a suitable one. The performance of all these solutions could be improved by considering them along with frameworks such as that from Jia and Zhang,[7] which enriches our original data by extracting new features that encode information about the class variables.

The above methods transform the multidimensional problem into one or more one-dimensional classifications. Rather, we will study the adaption of existing algorithms, which

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

tackle the problem more directly without requiring this preprocessing. Learning algorithms such as decision trees,[8] support vector machines,[9] *k*-nearest neighbors,[10] or Bayesian networks classifiers[11] have already been extended to perform simultaneous classification of multiple class variables. However, some of these proposals focus on the multilabel classification subproblem, where all class variables are binary.[12]

Apart from the multidimensional facet, the data studied in this article present a temporal dimension that cannot be ignored. To apply most of the above models over temporal data, it is common to follow a similar preprocessing strategy to that for static multidimensional data, called feature-based approach.[13] It transforms our original data set by extracting new feature variables that summarize the dynamics of time series during a time window. In this way, any traditional static classifier can be applied. Nonetheless, this implies the costly extraction of a probable large number of variables, which, in the end, may lose significant information. The adaptation of multidimensional classifiers to temporal data has not been extensively studied and few models can be directly applied in this context. Some of these algorithms include the multilabel *k*-nearest neighbors with dynamic time warping[10,14] and the long-short term memory recurrent neural networks.[15,16] Of these, the method of multilabel *k*-nearest neighbors stands out due to its simplicity and great performance.[17] Nonetheless, since it is a lazy classifier, it may imply a high space and classification complexity.

To the best of our knowledge, the direct application of probabilistic graphical models to the multidimensional classification of time series has received no attention, although they provide interesting characteristics such as an intuitive representation of variable dependencies. Dynamic Bayesian networks[18,19] are widely studied in the literature, but they only model discrete time, later extended to continuous time Bayesian networks.[20] For one-dimensional classification, the continuous time Bayesian network classifiers were introduced by Stella and Amer.[21] Here we extend them to the multidimensional problem, as well as different methods for its learning from data. We believe this is the first probabilistic graphical model able to perform time series multidimensional classification while explicitly modeling continuous time. More specifically, the main contributions of this study are the following:

- A novel multidimensional continuous time Bayesian network classifier that is able to model discrete state multivariate time series data and classify it into multiple class variables. This proposal explicitly represents the temporal dynamics of feature variables in continuous time and seeks to improve its predictions by modeling the dependencies between class variables.
- The introduction of algorithms for the learning of the parameters and structure of the presented model from data, as well as different structure constraints to adapt the model and its learning to the characteristics and demands of a certain problem.
- A comprehensive comparative study with 50 synthetic and a real-world Industry 4.0 data sets to validate, under different conditions, the proposed model's effectiveness and its performance improvements with respect to continuous time Bayesian network classifiers using the binary relevance strategy.
- The development of a software tool to allow the application of the presented model in other studies and the sampling of synthetic discrete state multivariate time series data sets with multiple class variables.

The remainder of this article is organized as follows. Section 2 reviews some fundamental concepts on which the presented model is based. Section 3 introduces the multidimensional continuous time Bayesian network classifier, while Sections 4 and 5 explain some methods to

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 List of acronyms

Acronym	Meaning
BDe	Bayesian Dirichlet equivalent
BIC	Bayesian information criterion
BN	Bayesian network
CLL	Conditional log-likelihood
CTBN	Continuous time Bayesian network
CTBNC	Continuous time Bayesian network classifier
CTNBC	Continuous time naive Bayes classifier
DAG	Directed acyclic graph
LL	Log-likelihood
MBC	Multidimensional Bayesian network classifier
Multi-CTBNC	Multidimensional continuous time Bayesian network classifier

learn its parameters and structure from data, respectively. Subsequently, Section 6 describes how the model performs multidimensional classification of unseen sequences. Section 7 presents synthetic and real-world experiments, and discusses the results that the proposed model obtains. Finally, Section 8 concludes the article and highlights future lines of research. For the sake of clarity, Table 1 provides the list of acronyms used in this study.

2 | FUNDAMENTALS

A Bayesian network (BN) is a probabilistic graphical model that encodes conditional independence assumptions over some random variables to obtain a factorized version of their joint probability distribution. These models have been widely used in a variety of domains[22–25] and their learning from data with different algorithms constitutes an important active research area. Some reasons for their success are the graphical representation of uncertainty, from which we could even learn causal relationships under certain conditions, that they can handle incomplete data, which are common in real-world problems, or that they can combine expert and data-extracted knowledge.[26,27]

Definition 1 (Bayesian network)**.** A BN $\mathcal{B} = (\mathcal{G}, \mathbf{B})$ over a set of random variables $\mathcal{X} = \{X_1, ..., X_m\}$ consists of a directed acyclic graph (DAG) $\mathcal{G}$ encoding conditional independence assumptions among the variables and a set of parameters $\mathbf{B}$ defining the statistical dependencies between them. This allows to factorize the joint probability distribution over $\mathcal{X}$ as

$$p(X_1, ..., X_m|\mathcal{G}, \mathbf{B}) = \prod_{i=1}^{n} p\left(X_i|Pa(X_i), \mathbf{B}_{X_i}^{Pa(X_i)}\right),$$

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where $Pa(X_i)$ are the parent variables of X_i in $\mathcal{G}$, that is, variables pointing at X_i in $\mathcal{G}$, and $\mathbf{B}_{X_i}^{Pa(X_i)}$ represents the conditional probability (discrete or continuous) distribution of X_i given $Pa(X_i)$.

The classical definition of a BN is limited to model static data, that is, samples that are assumed to be independent from each other, which is an inconvenience when data is dynamic. This article considers dynamic data with a temporal dimension, which is commonly known as time series data.

Definition 2 (Time series data set)**.** A multivariate time series data set $\mathcal{D} = \{\mathcal{S}_1, ..., \mathcal{S}_N\}$ consists of multiple multivariate temporal sequences or trajectories $\mathcal{S}_l = \{\mathbf{x}_l^{t_1}, ..., \mathbf{x}_l^{t_{T_l}}\}$ $(l = 1, ..., N)$*, which are time-ordered sets of T_l observations $\mathbf{x}_l^{t_j} = \{x_{l1}^{t_j}, ..., x_{lm}^{t_j}\}$ ($T_l - 1$ transitions) over some variables $\mathcal{X} = \{X_1, ..., X_m\}$. Each transition is represented by two pairs $\mathbf{x}_l^{t_j}$ and $\mathbf{x}_l^{t_{j+1}}$, where $t_j, t_{j+1} \in \mathbb{R}_{\geq 0}$ and $t_j < t_{j+1}$, such that variables have value $\mathbf{x}_l^{t_j}$ from time t_j to t_{j+1} and value $\mathbf{x}_l^{t_{j+1}}$ from t_{j+1} to t_{j+2}, where $j + 2 \leq T_l$.

Dynamic Bayesian networks are the best known extension of BNs to model temporal data and they have been successfully used in real-world problems.[28–30] Nonetheless, they are based on discretizing time, forcing us to define a uniform time granularity even when the described processes evolve at different rates. Continuous time Bayesian networks (CTBNs)[20] avoid this problem by describing the dynamics of each variable as a finite-state, continuous-time, homogeneous Markov process. Therefore, CTBNs can explicitly represent time, while still keeping the interesting graphical properties of BNs.

Definition 3 (Continuous time Bayesian network)**.** A CTBN $\mathcal{N} = (\mathcal{G}, \mathcal{Q}, P_{\mathcal{X}}^0)$ over a set of discrete random variables $\mathcal{X} = \{X_1, ..., X_m\}$, where each variable X_i has a sample space $\Omega_{X_i} = \{x_1, ..., x_k\}$,† consists of:

- A continuous transition model specified by a directed graph $\mathcal{G}$ and a conditional intensity matrix (CIM) $\mathcal{Q}_{X_i}^{Pa(X_i)}$ for each X_i that describes its temporal dependencies. A CIM can be seen as a set of intensity matrices $\mathcal{Q}_{X_i}^{pa(X_i)}$, each encoding the dynamics of X_i given an instantiation $pa(X_i)$ of its parents $Pa(X_i)$ in $\mathcal{G}$:

$$\mathcal{Q}_{X_i}^{pa(X_i)} = \begin{bmatrix} -q_{x_1}^{pa(X_i)} & q_{x_1,x_2}^{pa(X_i)} & \cdots & q_{x_1,x_k}^{pa(X_i)} \\ q_{x_2,x_1}^{pa(X_i)} & -q_{x_2}^{pa(X_i)} & \cdots & q_{x_2,x_k}^{pa(X_i)} \\ \vdots & \cdots & \ddots & \vdots \\ q_{x_k,x_1}^{pa(X_i)} & q_{x_k,x_2}^{pa(X_i)} & \cdots & q_{x_k,x_k}^{pa(X_i)} \end{bmatrix},$$

  where $q_{x_j}^{pa(X_i)} = \sum_{x_z \neq x_j} q_{x_j,x_z}^{pa(X_i)}$ is the intensity of X_i leaving state x_j and $q_{x_j,x_z}^{pa(X_i)}$ is proportional to the probability of X_i transitioning from state x_j to x_z, both when its parents have value $pa(X_i)$. CTBNs use this continuous transition model to define two distributions for each variable. An exponential distribution over the time a variable remains in a certain state and a multinomial distribution over which state a variable will transition to when its current state is known to change.
- An initial distribution $P_{\mathcal{X}}^0$, specified as a BN, that represents the initial state of a temporal process.

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Unlike a BN, the graph $\mathcal{G}$ of a CTBN can be cyclic since its arcs represent the dependencies between variables across time, that is, the state to which a variable will transition depends on the current state of other variables.

As with BNs, the interest in CTBNs is not solely motivated by knowledge discovery, but they can also be applied to the classification of unseen sequences.

Definition 4 (Time series classification)**.** This task consists on training a classification model with a (multivariate) time series data set, whose sequences $\mathcal{S}_l = \{\mathbf{x}_l^{t_1}, ..., \mathbf{x}_l^{t_{T_l}}, c_l\}$ describe the transitions of m, time dependent, feature variables $\mathcal{X}$ and have assigned a unique state c_l for a, time independent, class variable C. The objective is to use the model to predict the state of C on a previously unseen sequence $\mathcal{S}_p = \{\mathbf{x}_p^{t_1}, ..., \mathbf{x}_p^{t_{T_p}}\}$, where class variable information is missing, by analyzing the state transitions of the feature variables.

The family of classification models known as continuous time Bayesian network classifiers (CTBNCs)[21] are able to perform classification over the data introduced in Definition 4 by incorporating a new, time independent, class variable node to the CTBNs.

Definition 5 (Continuous time Bayesian network classifier)**.** A CTBNC is a pair $\mathcal{C} = \{\mathcal{N}, P(C)\}$, where $\mathcal{N}$ is a CTBN over a set of feature variables $\mathcal{X} = \{X_1, ..., X_m\}$ and a time independent class variable C, with sample space $\Omega_C = \{c_1, ..., c_k\}$, that is fully specified by the marginal probability $P(C)$. The CTBNC graph has the same properties of that of a CTBN, but includes a class variable node with no parents, that is, $Pa(C) = \varnothing$.

3 | MULTIDIMENSIONAL CONTINUOUS TIME BAYESIAN NETWORK CLASSIFIER

The existence of multidimensional classification data provoked the appearance of multidimensional Bayesian network classifiers (MBCs),[11] but in static settings. However, there are real-world problems where we need to classify temporal sequences into multiple class variables, that is, a sequence $\mathcal{S}_l = \{\mathbf{x}_l^{t_1}, ..., \mathbf{x}_l^{t_{T_l}}, \mathbf{c}_l\}$ now includes the state of d class variables $\mathbf{c}_l = \{c_{l1}, ..., c_{ld}\}$. Take as an example the problem presented in Section 7.1.2, which requires to identify the power consumption state (discretized as high, low or inactive) of elements of an industrial machine based on the energy data it produces. Surely, several individual models (one per element) can be used to predict each of them. However, this would not identify inter-element dependencies, which could provide valuable, or even crucial, information to classify certain class variables. For example, it is known that some elements always work together, while others cannot be active at the same time. The multidimensional continuous time Bayesian network classifier (Multi-CTBNC) that we introduce here seeks to extend CTBNCs to this more complex scenario, allowing to model the interclass interactions by capturing the probabilistic relationships of class variables with a BN.

Definition 6 (Multidimensional continuous time Bayesian network classifier)**.** A Multi-CTBNC $\mathcal{M} = (\mathcal{G}, \mathbf{B}, \mathcal{Q}, P_{\mathcal{V}}^0)$ over a set of discrete variables $\mathcal{V} = \{X_1, ..., X_m, C_1, ..., C_d\}$ is formed by:

- A directed (possibly cyclic) graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where vertices $\mathcal{V}$ are partitioned into those for feature variables $\mathcal{V}_{\mathcal{X}} = \{X_1, ..., X_m\}$, $m \geq 1$, and for class variables

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

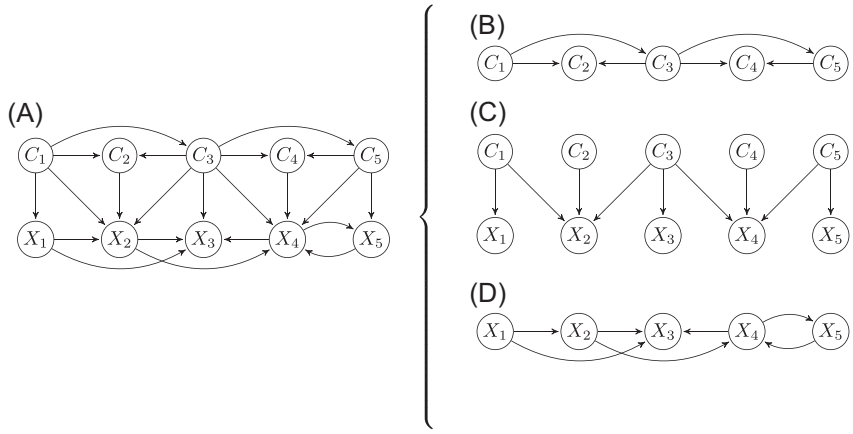

FIGURE 1 A Multi-CTBNC graph (A) and its class (B), bridge (C), and feature (D) subgraphs. CTBNC, continuous time Bayesian network classifier

$\mathcal{V}_\mathcal{C} = \{C_1, ..., C_d\}$, $d \geq 1$, and arcs $\mathcal{A} = \{\mathcal{A}_\mathcal{C}, \mathcal{A}_\mathcal{X}, \mathcal{A}_{\mathcal{C}\mathcal{X}}\}$ are divided into those between class variables $\mathcal{A}_\mathcal{C} \subseteq \mathcal{V}_\mathcal{C} \times \mathcal{V}_\mathcal{C}$, feature variables $\mathcal{A}_\mathcal{X} \subseteq \mathcal{V}_\mathcal{X} \times \mathcal{V}_\mathcal{X}$ and from class to feature variables $\mathcal{A}_{\mathcal{C}\mathcal{X}} \subseteq \mathcal{V}_\mathcal{C} \times \mathcal{V}_\mathcal{X}$.

- Class variable parameters $\mathbf{B}$, which form conditional probability tables (CPTs).
- A set of CIMs $\mathcal{Q}$, one for each feature variable X_i.
- An initial distribution $P^0_\mathcal{V}$ to represent the initial state of a sequence. As more than one class variable is present, $P^0_\mathcal{V}$ is specified as an MBC.

As the class variables do not depend on time, a Multi-CTBNC is based on capturing their probabilistic relationships with BNs. Thus, this model can be decomposed into a BN and a CTBN, which divide $\mathcal{G}$ into three subgraphs:

1. The *class subgraph* $\mathcal{G}_\mathcal{C} = (\mathcal{V}_\mathcal{C}, \mathcal{A}_\mathcal{C})$: is defined by a BN that models the dependencies between class variables. This subgraph must be a DAG.
2. The *feature subgraph* $\mathcal{G}_\mathcal{X} = (\mathcal{V}_\mathcal{X}, \mathcal{A}_\mathcal{X})$: is defined by a CTBN that models the dependencies between feature variables over time and, therefore, cycles may appear.
3. The *bridge subgraph* $\mathcal{G}_{\mathcal{C}\mathcal{X}} = (\mathcal{V}, \mathcal{A}_{\mathcal{C}\mathcal{X}})$: represents the dependencies of features on class variables and, therefore, it is defined by the same CTBN as the feature subgraph. It is also known as feature selection subgraph[11] since it specifies which feature variables are relevant for classification.

Figure 1 shows a Multi-CTBNC graph and its constituent subgraphs. Note that the structure of this model shares many similarities with that of an MBC, despite the multiple differences in their underlying paradigms, but the Multi-CTBNC allows the appearance of cycles in its feature subgraph.

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4 | PARAMETER LEARNING

The parameters of a Multi-CTBNC are those of a BN and a CTBN. As we are assuming all variables to be discrete, Multi-CTBNC nodes would contain either CPTs (for class variables) with parameters:

- $\beta_{c_j}^{pa(C_y)}$: probability of class variable C_y taking state c_j given the parents' state $pa(C_y)$. These are the parameters for the multinomial distribution over class variables' state;

or CIMs (for feature variables), which are summarized by two types of parameters:

- $q_{x_j}^{pa(X_f)}$: intensity of feature variable X_f leaving state x_j when the parents' state is $pa(X_f)$. This is the parameter for the exponential distribution over the time a feature variable remains in a certain state.
- $\theta_{x_j,x_z}^{pa(X_f)} = \frac{q_{x_j,x_z}^{pa(X_f)}}{q_{x_j}^{pa(X_f)}}$: probability of X_f transitioning from state x_j to x_z, where $x_j \neq x_z$, when a transition is known to occur and the parents' state is $pa(X_f)$. These are the parameters for the multinomial distribution over which state a feature variable will transition to.

Therefore, the parameters for each class variable, $\mathbf{B}_{C_y}^{Pa(C_y)} = \{\beta_{c_j}^{pa(C_y)} : c_j \in \Omega_{C_y}\}$, and each feature variable, $\mathbf{q}_{X_f}^{Pa(X_f)} = \{q_{x_j}^{pa(X_f)} : x_j \in \Omega_{X_f}\}$ and $\mathbf{\Theta}_{X_f}^{Pa(X_f)} = \{\theta_{x_j,x_z}^{pa(X_f)} : x_j, x_z \in \Omega_{X_f}, x_j \neq x_z\}$, define the parameters $\mathbf{B} = \{\mathbf{B}_{C_y}^{Pa(C_y)} : C_y \in \mathcal{V}_C\}$, $\mathbf{q} = \{\mathbf{q}_{X_f}^{Pa(X_f)} : X_f \in \mathcal{V}_X\}$ and $\mathbf{\Theta} = \{\mathbf{\Theta}_{X_f}^{Pa(X_f)} : X_f \in \mathcal{V}_X\}$ of a Multi-CTBNC. To estimate these parameters, some sufficient statistics that summarize all observable data are recorded:

- $N_{c_j}^{pa(C_i)}$: number of sequences where class variable C_i takes state c_j while the parents' state is $pa(C_i)$.
- $N^{pa(C_i)} = \sum_{c_j} N_{c_j}^{pa(C_i)}$: number of sequences where parents of C_i have state $pa(C_i)$ independently of the state of C_i.
- $M_{x_j,x_z}^{pa(X_i)}$: number of transitions of feature variable X_i from state x_j to x_z when the parents' state is $pa(X_i)$.
- $M_{x_j}^{pa(X_i)} = \sum_{x_z \neq x_j} M_{x_j,x_z}^{pa(X_i)}$: number of transitions of X_i from state x_j to any other when the parents' state is $pa(X_i)$.
- $T_{x_j}^{pa(X_i)}$: time of X_i spent in state x_j when the parents' state is $pa(X_i)$.

Given the structure of a Multi-CTBNC, its parameters can be estimated with methods like maximum likelihood estimation or Bayesian estimation. The first approach assumes that the parameters are constants, seeking those values that maximize the probability of the observable data, that is, the maximum likelihood estimates. Parameters are then estimated as follows:[31,32]

$$\hat{\beta}_{c_j}^{pa(C_i)} = \frac{N_{c_j}^{pa(C_i)}}{N^{pa(C_i)}}, \quad \hat{q}_{x_j}^{pa(X_i)} = \frac{M_{x_j}^{pa(X_i)}}{T_{x_j}^{pa(X_i)}}, \quad \text{and} \quad \hat{\theta}_{x_j,x_z}^{pa(X_i)} = \frac{M_{x_j,x_z}^{pa(X_i)}}{M_{x_j}^{pa(X_i)}}.$$

In the case of the Bayesian estimation, the parameters are considered random variables and a prior distribution is defined over them. This is a more interesting approach since we can add prior expert knowledge and avoid the zero-count problem.[33] Conjugate prior distributions are defined for the two types of distributions used by the Multi-CTBNC, the multinomial and

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

exponential distributions. The most common prior distribution for the multinomial parameters is the Dirichlet distribution,[34] while for the exponential parameter, the Gamma distribution is an appropriate choice.[32] As conjugate priors are used, the posterior distribution of the parameters given the observed data follows the same distribution and, therefore, it can be obtained analytically. Then, the parameters can be estimated using, for example, their expected values, in the same way as the maximum likelihood estimation, but including the hyperparameters of the Dirichlet prior distributions $\lambda_{c_j}^{pa(C_i)}$ and $\alpha_{x_j,x_z}^{pa(X_i)}$, and of the Gamma prior distribution $\alpha_{x_j}^{pa(X_i)}$ and $\tau_{x_j}^{pa(X_i)}$:[31,32]

$$\hat{\beta}_{c_j}^{pa(C_i)} = \frac{N_{c_j}^{pa(C_i)} + \lambda_{c_j}^{pa(C_i)}}{N^{pa(C_i)} + \sum_{c_z} \lambda_{c_z}^{pa(C_i)}}, \quad \hat{q}_{x_j}^{pa(X_i)} = \frac{M_{x_j}^{pa(X_i)} + \alpha_{x_j}^{pa(X_i)}}{T_{x_j}^{pa(X_i)} + \tau_{x_j}^{pa(X_i)}},$$

and

$$\hat{\theta}_{x_j,x_z}^{pa(X_i)} = \frac{M_{x_j,x_z}^{pa(X_i)} + \alpha_{x_j,x_z}^{pa(X_i)}}{M_{x_j}^{pa(X_i)} + \alpha_{x_j}^{pa(X_i)}}.$$

These hyperparameters can be seen as imaginary counts of the sufficient statistics that occur before any data is observed, that is, $\lambda_{c_j}^{pa(C_i)}$ is the number of times a class variable C_i takes state c_j, $\alpha_{x_j,x_z}^{pa(X_i)}$ is the number of transitions of a feature variable X_i from state x_j to state x_z, $\alpha_{x_j}^{pa(X_i)} = \sum_{x_z \neq x_j} \alpha_{x_j,x_z}^{pa(X_i)}$ is the number of transitions from state x_j to any other different state and $\tau_{x_j}^{pa(X_i)}$ is the time X_i remains in state x_j, all before a data set $\mathcal{D}$ is considered. The hyperparameters can be defined by using expert knowledge and/or optimization techniques,[35] such as random search[36] or Bayesian optimization.[37]

5 | STRUCTURE LEARNING

The problem of learning the structure of a CTBN has been traditionally approached as an optimization problem,[32,38,39] where a score assigned to structures is maximized. Therefore, this section adapts some common scores to the multidimensional classification problem with a Multi-CTBNC.

Score-based algorithms define a score to evaluate how well a model fits the observed data and a space of candidate structures the model can take. The components of these algorithms are:

- An optimization algorithm: some examples are greedy hill climbing, tabu search, simulated annealing or genetic algorithms.[32,34,40,41]
- A hypothesis space of structures: in the case of a Multi-CTBNC, the class subgraph must be acyclic and the bridge subgraph only contains arcs from class variables to features. The feature subgraph has no restrictions.
- A score metric: these are commonly divided into those based on the likelihood function and the Bayesian scores. The next sections will introduce three different scores for learning the structure of Multi-CTBNCs.

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

5.1 | Log-likelihood

The simplest approach to learn the structure of a Multi-CTBNC is to maximize the likelihood of the observed data given the model, $P(\mathcal{D}|\mathcal{M})$. The likelihood function of a Multi-CTBNC is obtained by incorporating the appearance and dependencies of class variables to the likelihood function of a CTBN:[32]

$$\mathrm{L}(\mathcal{M}:\mathcal{D}) = \prod_{y=1}^{d} \prod_{pa(C_y)} \prod_{c_j} \left(\hat{\beta}_{c_j}^{pa(C_y)}\right)^{N_{c_j}^{pa(C_y)}} \tag{1a}$$

$$\prod_{f=1}^{m} \prod_{pa(X_f)} \prod_{x_j} \left(\hat{q}_{x_j}^{pa(X_f)}\right)^{M_{x_j}^{pa(X_f)}} \exp\left(-\hat{q}_{x_j}^{pa(X_f)} T_{x_j}^{pa(X_f)}\right) \prod_{x_z \neq x_j} \left(\hat{\theta}_{x_j,x_z}^{pa(X_f)}\right)^{M_{x_j,x_z}^{pa(X_f)}}. \tag{1b}$$

The equation above shows that the likelihood for a Multi-CTBNC is decomposed into those for a BN (1a) and a CTBN (1b). This means that the learning of the class subgraph structure (BN) and the feature and bridge subgraphs (CTBN) of a Multi-CTBNC can be performed separately since they do not influence each other. In the case of a BN, the search space is limited to directed acyclic graphs, while the search space of a CTBN is simpler, since the graph can be cyclic. As the bridge subgraph encodes the dependencies of the features on the class variables, it is also defined during the learning of the CTBN. As explained above, its structure has to be restricted to only allow arcs from class variables to features. This is the same restriction found in a CTBNC, but here extended to more than one class variable.

In practice, instead of using the likelihood function, a better approach is to maximize the log-likelihood (LL), which is monotonically related.[31] The reason for this is that the LL is much easier to maximize and it helps to prevent underflows and overflows caused by the multiplication of small numbers and the exponential function:

$$\mathrm{LL}(\mathcal{M}:\mathcal{D}) = \sum_{y=1}^{d} \sum_{pa(C_y)} \sum_{c_j} N_{c_j}^{pa(C_y)} \log\left(\hat{\beta}_{c_j}^{pa(C_y)}\right) \tag{2a}$$

$$+ \sum_{f=1}^{m} \sum_{pa(X_f)} \sum_{x_j} \left[M_{x_j}^{pa(X_f)} \log\left(\hat{q}_{x_j}^{pa(X_f)}\right) - \hat{q}_{x_j}^{pa(X_f)} T_{x_j}^{pa(X_f)} + \sum_{x_z \neq x_j} M_{x_j,x_z}^{pa(X_f)} \log\left(\hat{\theta}_{x_j,x_z}^{pa(X_f)}\right) \right]. \tag{2b}$$

This score tends to overfit the data by favoring densely connected networks. Therefore, a penalization factor over the complexity of the network, that is, the number of parameters, can be included. Widely known penalization functions such as the Bayesian information criterion (BIC)[42] or the Akaike information criterion[43] can be applied over both the LLs of the BN and the CTBN. For example, the LL of a Multi-CTBNC with BIC penalization is:

$$BIC(\mathcal{M}:\mathcal{D}) = \mathrm{LL}(\mathcal{M}:\mathcal{D}) - \frac{dim(\mathcal{G}_{\mathcal{C}})}{2}\log(N) - \frac{dim(\mathcal{G}_{\mathcal{X}} \cup \mathcal{G}_{\mathcal{CX}})}{2}\log(N),$$

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where $dim(\mathcal{G}_{\mathcal{C}}) = \sum_{y=1}^{d} r_{C_y}(|\Omega_{C_y}| - 1)$ and $dim(\mathcal{G}_{\mathcal{X}} \cup \mathcal{G}_{\mathcal{C}\mathcal{X}}) = \sum_{f=1}^{m} r_{X_f} z_{X_f}$ are the dimension (number of independent parameters) of the BN and CTBN, respectively, r_{C_y} is the number of possible instantiations of $Pa(C_y)$ and $z_{X_f} = (|\Omega_{X_f}| - 1)|\Omega_{X_f}|$ is the number of possible transitions from each state of X_f to any other.

5.2 | Conditional log-likelihood

When facing a multidimensional classification problem, the LL can be defined as:

$$\mathrm{LL}(\mathcal{M} : \mathcal{D}) = \sum_{l=1}^{N} \log P\left(\mathbf{c}_l | \mathbf{x}_l^{t_1}, ..., \mathbf{x}_l^{t_{T_l}}\right) + \sum_{l=1}^{N} \log P\left(\mathbf{x}_l^{t_1}, ..., \mathbf{x}_l^{t_{T_l}}\right). \tag{3}$$

Equation (3) divides the LL into a first term representing the model's ability to classify a sequence and a second term describing the dependencies among feature variables. If the number of features is large, the LL is dominated by the second term, which may negatively impact the performance of the classifier. Consequently, Friedman et al.[44] proposed to only focus on the first term, which is the conditional log-likelihood (CLL), thereby following a discriminative approach. The CLL function was previously proposed by Codecasa and Stella[38] for the learning of CTBNCs, and here we extend it for Multi-CTBNCs.

The idea is to specialize the LL of the CTBN (Equation 2b) in the classification task since it defines the bridge subgraph and, therefore, the feature variables that are relevant for classification. The LL of the BN (Equation 2a) remains unchanged, as interclass dependencies may be relevant for the classification. This results in the following score:

$$\mathrm{CLL}(\mathcal{M} : \mathcal{D}) = \sum_{l=1}^{N} \log P(\mathbf{c}_l) + \sum_{l=1}^{N} \log P\left(\mathbf{c}_l | \mathbf{x}_l^{t_1}, ..., \mathbf{x}_l^{t_{T_l}}\right),$$

where

$$\begin{aligned} \textstyle\sum_{l=1}^{N} \log P\left(\mathbf{c}_l | \mathbf{x}_l^{t_1}, ..., \mathbf{x}_l^{t_{T_l}}\right) &= \textstyle\sum_{l=1}^{N} \log \left(\frac{P\left(\mathbf{x}_l^{t_1}, \ldots, \mathbf{x}_l^{t_{T_l}} \mid \mathbf{c}_l\right) P(\mathbf{c}_l)}{P\left(\mathbf{x}_l^{t_1}, \ldots, \mathbf{x}_l^{t_{T_l}}\right)} \right) \\ &= \textstyle\sum_{l=1}^{N} \Big[\log P\left(\mathbf{x}_l^{t_1}, ..., \mathbf{x}_l^{t_{T_l}} | \mathbf{c}_l\right) + \log P(\mathbf{c}_l) \\ &\quad - \log\left(\textstyle\sum_{\mathbf{c}'} P\left(\mathbf{x}_l^{t_1}, ..., \mathbf{x}_l^{t_{T_l}} | \mathbf{c}'\right) P(\mathbf{c}') \right) \Big]. \end{aligned} \tag{4}$$

Equation (4) includes a normalization factor (denominator term) which is what differentiates its result from that given by the LL (Equation 3). Most notable differences with respect to the CLL for a CTBNC are found in the class probability and denominator terms, which take into account the multiple class variables and their dependencies. These are estimated, respectively, as follows:

$$\sum_{l=1}^{N} \log P(\mathbf{c}_l) = \sum_{y=1}^{d} \sum_{pa(C_y)} \sum_{c_j} N_{c_j}^{pa(C_y)} \log\left(\hat{\beta}_{c_j}^{pa(C_y)} \right),$$

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

and

$$\sum_{l=1}^{N} \log\left(\sum_{\mathbf{c}'} P\left(\mathbf{x}_l^{t_l}, ..., \mathbf{x}_l^{t_{T_l}} | \mathbf{c}'\right) P(\mathbf{c}')\right) = \log\left(\sum_{\mathbf{c}'} \prod_{y=1}^{d} \hat{\beta}_{c'_y}^{pa(C_y)} \prod_{f=1}^{m} \prod_{pa(X_f)} \prod_{x_j} \left(\hat{q}_{x_j}^{pa(X_f)}\right)^{M_{x_j}^{pa(X_f)}} \exp\left(-\hat{q}_{x_j}^{pa(X_f)} T_{x_j}^{pa(X_f)}\right) \prod_{x_z \neq x_j} \left(\hat{\theta}_{x_j,x_z}^{pa(X_f)}\right)^{M_{x_j,x_z}^{pa(X_f)}}\right). \tag{5}$$

In the case of the likelihood term $P(\mathbf{x}_l^{t_l}, ..., \mathbf{x}_l^{t_{T_l}} | \mathbf{c}_l)$, the only difference is that parameters and sufficient statistics are defined based on the state of, potentially, multiple class variables that are parents of the features. Unfortunately, the denominator term cannot be further decomposed, as the logarithm is applied to a sum over all class configurations. Therefore, the log-sum-exp trick may be required in practice to prevent underflows and overflows.[33]

5.3 | Bayesian Dirichlet equivalent score

This section presents a Bayesian score function for learning Multi-CTBNCs, the Bayesian Dirichlet equivalent (BDe) score, which was first presented for CTBNs by Nodelman et al.[32] Bayesian scores are defined as

$$\mathrm{BS}(\mathcal{G} : \mathcal{D}) = \log P(\mathcal{D}|\mathcal{G}) + \log P(\mathcal{G}),$$

which are derived from the logarithm of the Bayes' rule to obtain the model structure with the highest probability given the data:

$$P(\mathcal{G}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{G})P(\mathcal{G})}{P(\mathcal{D})} \propto P(\mathcal{D}|\mathcal{G})P(\mathcal{G}).$$

They incorporate a prior probability over the model structures, $P(\mathcal{G})$, which is maximized together with the marginal likelihood of the data given the structure, $P(\mathcal{D}|\mathcal{G})$, which, in contrast to the likelihood, does not consider a specific assignment of the parameters. It rather adds uncertainty about them by integrating over all their possible values for $\mathcal{G}$:

$$P(\mathcal{D}|\mathcal{G}) = \int_{\mathbf{B},\mathbf{q},\boldsymbol{\Theta}} P(\mathcal{D}|\mathbf{B}, \mathbf{q}, \boldsymbol{\Theta}, \mathcal{G}) P(\mathbf{B}, \mathbf{q}, \boldsymbol{\Theta}|\mathcal{G}) d\mathbf{B} d\mathbf{q} d\boldsymbol{\Theta}.$$

This has the advantage over the previous scores of hopefully reducing overfitting by evaluating over all possible values of the parameters.

Making the common assumptions of global and local parameter independence,[34] the log-marginal-likelihood of the data given the structure (i.e., $\log P(\mathcal{D}|\mathcal{G})$) can be decomposed into the sum of local log-marginal-likelihoods for each type of parameter and variable (LML):

$$\log P(\mathcal{D}|\mathcal{G}) = \sum_{y=1}^{d} \mathrm{LML}\left(\mathbf{B}_{C_y}^{Pa(C_y)} : \mathcal{D}\right) + \sum_{f=1}^{m} \left[\mathrm{LML}\left(\mathbf{q}_{X_f}^{Pa(X_f)} : \mathcal{D}\right) + \mathrm{LML}\left(\boldsymbol{\Theta}_{X_f}^{Pa(X_f)} : \mathcal{D}\right)\right].$$

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Given that the BDe score assumes a Dirichlet distribution over the parameter priors $P(\mathbf{B}|\mathcal{G})$ and $P(\boldsymbol{\Theta}|\mathcal{G})$ (with hyperparameters of Section 4), the log-marginal-likelihoods for the parameters $\mathbf{B}_{C_y}^{Pa(C_y)}$ and $\boldsymbol{\Theta}_{X_f}^{Pa(X_f)}$ can be decomposed, respectively, as follows (for a derivation see [32,45]):

$$\mathrm{LML}\left(\mathbf{B}_{C_y}^{Pa(C_y)} : \mathcal{D}\right) = \log\left(\prod_{pa(C_y)} \frac{\Gamma\left(\sum_{c_j} \lambda_{c_j}^{pa(C_y)}\right)}{\Gamma\left(\sum_{c_j} \lambda_{c_j}^{pa(C_y)} + N_{c_j}^{pa(C_y)}\right)} \prod_{c_j} \frac{\Gamma\left(\lambda_{c_j}^{pa(C_y)} + N_{c_j}^{pa(C_y)}\right)}{\Gamma\left(\lambda_{c_j}^{pa(C_y)}\right)}\right),$$

and

$$\mathrm{LML}(\boldsymbol{\Theta}_{X_f}^{Pa(X_f)} : \mathcal{D}) = \log\left(\prod_{pa(X_f)} \prod_{x_f} \frac{\Gamma\left(\alpha_{x_f}^{pa(X_f)}\right)}{\Gamma\left(\alpha_{x_f}^{pa(X_f)} + M_{x_f}^{pa(X_f)}\right)} \prod_{x_z \neq x_f} \frac{\Gamma\left(\alpha_{x_f,x_z}^{pa(X_f)} + M_{x_f,x_z}^{pa(X_f)}\right)}{\Gamma\left(\alpha_{x_f,x_z}^{pa(X_f)}\right)}\right),$$

where $\Gamma(\cdot)$ is the gamma function.

In the case of $P(\mathbf{q}|\mathcal{G})$, a Gamma distribution is assumed (with hyperparameters of Section 4). Then, the log-marginal-likelihood for $\mathbf{q}_{X_f}^{Pa(X_f)}$ can be estimated with the following closed formula (for a derivation see [32]):

$$\mathrm{LML}(\mathbf{q}_{X_f}^{Pa(X_f)} : \mathcal{D}) = \log\left(\prod_{pa(X_f)} \prod_{x_j} \frac{\Gamma\left(\alpha_{x_j}^{pa(X_f)} + M_{x_j}^{pa(X_f)} + 1\right)\left(\tau_{x_j}^{pa(X_f)}\right)^{\alpha_{xj}^{pa(X_f)}+1}}{\Gamma\left(\alpha_{x_j}^{pa(X_f)} + 1\right)\left(\tau_{x_j}^{pa(X_f)} + T_{x_j}^{pa(X_f)}\right)^{\alpha_{xj}^{pa(X_f)}+M_{xj}^{pa(X_f)}+1}}\right).$$

Finally, if a uniform prior over the structures $P(\mathcal{G})$ is considered, the BDe score simply maximizes the log-marginal-likelihood of the observed data given a Multi-CTBNC:

$$\mathrm{BDe}(\mathcal{G} : \mathcal{D}) = \sum_{i=1}^{d} \mathrm{LML}\left(\mathbf{B}_{C_i}^{pa(C_i)} : \mathcal{D}\right) + \sum_{f=1}^{m} \left[\mathrm{LML}\left(\mathbf{q}_{X_f}^{pa(X_f)} : \mathcal{D}\right) + \mathrm{LML}\left(\boldsymbol{\Theta}_{X_f}^{pa(X_f)} : \mathcal{D}\right)\right].$$

To further penalize complex structures, a nonuniform structure prior $P(\mathcal{G})$, such as a Binomial prior distribution,[46] can be considered.

5.4 | Structure constraints

Due to the complexity to find and learn all possible Multi-CTBNCs, some assumptions can be made about the structure, that is, the hypothesis space can be reduced. Furthermore, these assumptions could help to learn models with better performance since they can prevent overfitting the data.

Likewise for MBCs, a variety of Multi-CTBNC families can be proposed considering different search spaces for the class and feature subgraphs. For example, they can be

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

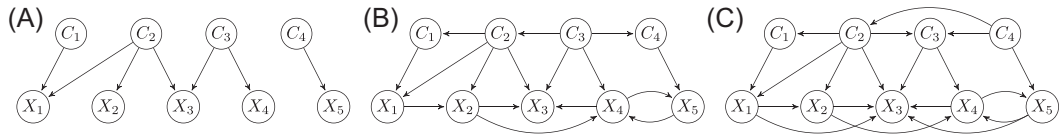

FIGURE 2 Examples of structures from different Multi-CTBNC families. (A) Empty-empty Multi-CTBNC, (B) tree-max2 Multi-CTBNC, and (C) DAG-digraph Multi-CTBNC. CTBNC, continuous time Bayesian network classifier; DAG, directed acyclic graph

limited to be empty, a tree, a forest, a polytree, a max$K$ (nodes have K parents at most), a DAG or, in the case of the feature subgraph, a directed graph (digraph). Following the notation proposed by Bielza et al.,[47] the different families are denoted as {class subgraph structure} − {feature subgraph structure} Multi-CTBNC. Some examples like the empty-empty Multi-CTBNC, tree-max$K$ Multi-CTBNC and DAG-digraph Multi-CTBNC are shown in Figure 2.

A well-known subclass of classifiers is the naive Bayes family, which assumes conditional independence between features given the class variables. In the case of the Multi-CTBNC, a fully naive model is an empty-empty Multi-CTBNC (see Figure 2A) with a complete bridge subgraph. The benefit of this model over independent continuous time naive Bayes classifiers (CTNBCs)[21] is that the number of parameters can be drastically reduced for certain data sets. Take as example the parameters that would be learned from a data set with eight ternary variables, three of them class variables. If three CTNBCs are built, the total number of independent parameters would be 276 since there would be 45 intensity matrices (three for each of the 15 feature nodes), each of them with six degrees of freedom, plus three CPTs with a degree of freedom of two. However, if there are exclusively five possible class configurations, a fully naive Multi-CTBNC would only require 156 parameters since the number of intensity matrices is reduced to 25 (five for each of the five feature nodes). Therefore, if the number of possible states of the class variables is large, but the number of class configurations is relatively small, a fully naive Multi-CTBNC would need a potentially much smaller number of parameters.

The *-max$K$ Multi-CTBNC family is of special interest since, for fixed K, the learning of a CTBN can be performed in polynomial time depending on the number of variables and data set size.[32] This is possible since the parent set of each CTBN feature can be optimized individually without having to worry about avoiding cycles in the resulting structure. Unfortunately, complexity for learning the bridge and feature subgraphs increases rapidly with the inclusion of more class variables. A *-max$K$ Multi-CTBNC does not limit the number of class variables that can be parents of the features, in the same way as a *-max$K$ MBC does, so the total number of parents could potentially be $K + d$ for each feature. Evidently, this problem could be alleviated by imposing restrictions on the number of class variables.[48]

Regarding the learning of the class subgraph, if a general DAG or even a max2 are considered, finding its optimal structure would be NP-hard due to the acyclicity constraint.[49] If the number of class variables is relatively large, a tree structure may be a better option since polynomial-time learning algorithms can be applied.[34,44,50]

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

6 | CLASSIFICATION

Given a sequence $\mathcal{S}_p = \{\mathbf{x}_p^{t_1}, ..., \mathbf{x}_p^{t_{T_p}}\}$, whose transitions are fully observed, classification is performed by choosing the class configuration that maximizes the posterior probability, that is, the maximum a posteriori estimate of

$$P(\mathbf{c}|\mathcal{S}_p) = \frac{P(\mathcal{S}_p|\mathbf{c})P(\mathbf{c})}{P(\mathcal{S}_p)} \propto P(\mathcal{S}_p|\mathbf{c})P(\mathbf{c}) = \prod_{j=1}^{T_p-1}\prod_{f=1}^{m} P\left(x_{pf}^{t_j}|\mathbf{c}\right)P\left(x_{pf}^{t_{j+1}}|x_{pf}^{t_j}, \mathbf{c}\right)\prod_{y=1}^{d} P(c_y|pa(C_y)). \tag{6}$$

The term $P(x_{pf}^{t_j}|\mathbf{c})$ is the probability that feature variable X_f stays in state $x_{pf}^{t_j}$ during the time interval of length $\delta_j = t_{j+1} - t_j$ given the class configuration $\mathbf{c}$:

$$P\left(x_{pf}^{t_j}|\mathbf{c}\right) = \exp\left(-q_{x_{pf}^{t_j}}^{pa(X_f)}\delta_j\right),$$

while $P(x_{pf}^{t_{j+1}}|x_{pf}^{t_j}, \mathbf{c})$ is the probability that X_f transitions from state $x_{pf}^{t_j}$ to state $x_{pf}^{t_{j+1}}$ when it is known that a transition occurs and given the class configuration $\mathbf{c}$:

$$P\left(x_{pf}^{t_{j+1}}|x_{pf}^{t_j}, \mathbf{c}\right) = \begin{cases} 1 - \exp\left(-q_{x_{pf}^{t_j}}^{pa(X_f)}\theta_{x_{pf}^{t_j}, x_{pf}^{t_{j+1}}}^{pa(X_f)}\epsilon\right), & \text{if } x_{pf}^{t_j} \neq x_{pf}^{t_{j+1}} \\ 1 & \text{otherwise}, \end{cases}$$

where ϵ is a small positive number. Finally, $P(c_y|pa(C_y))$ represents the probability of class variable C_y taking state c_y given the parent's state $pa(C_y)$:

$$P(c_y|pa(C_y)) = \beta_{c_y}^{pa(C_y)}.$$

Therefore, the predicted class configuration $\mathbf{c}^*$ for a sequence $\mathcal{S}_p$ is

$$\mathbf{c}^* = \underset{\mathbf{c}}{\operatorname{argmax}} \prod_{j=1}^{T_p-1}\prod_{f=1}^{m}\exp\left(-q_{x_{pf}^{t_j}}^{pa(X_f)}\delta_j\right)P\left(x_{pf}^{t_{j+1}}|x_{pf}^{t_j}, \mathbf{c}\right)\prod_{y=1}^{d}\beta_{c_y}^{pa(C_y)}. \tag{7}$$

As it happened before, the logarithm of the argmax argument in Equation (7) is used instead for convenience.

If estimating the a posteriori probabilities of each class configuration is needed, the marginal likelihood of the sequence $\mathcal{S}_p$ must be computed, that is, the denominator term of the posterior probability (Equation 6) cannot be ignored:

$$P(\mathcal{S}_p) = \sum_{\mathbf{c}'} P(\mathcal{S}_p|\mathbf{c}')P(\mathbf{c}').$$

Unfortunately, as shown in Equation (5), this term cannot be further decomposed.

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

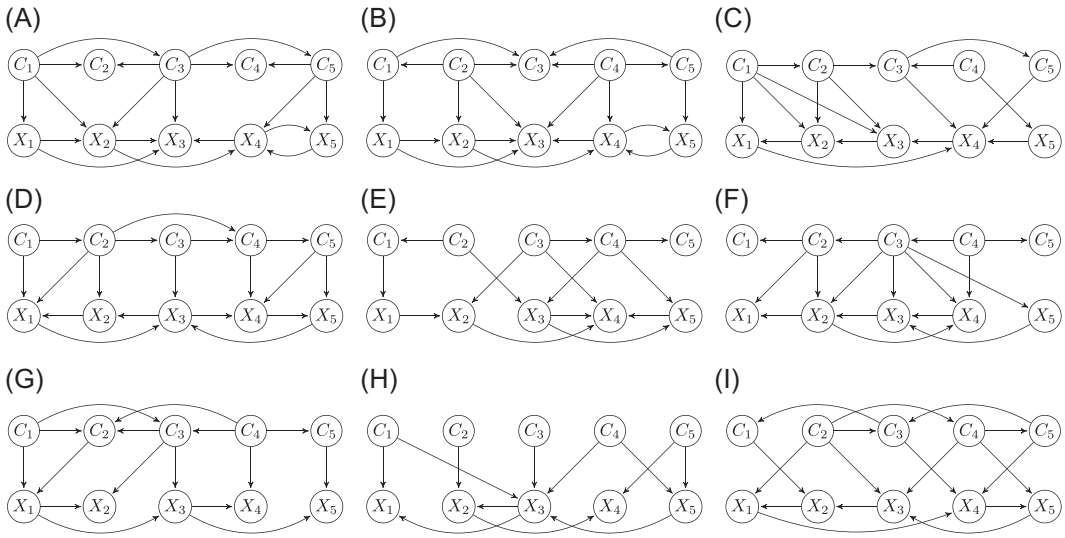

FIGURE 3 Structures from which synthetic data sets were generated

7 | EXPERIMENTS

This section will empirically compare the performance of the Multi-CTBNC with multiple independent CTBNCs for the multidimensional classification of synthetic and real-world time series data. The assessment will be performed with several performance evaluation metrics estimated with a fivefold cross-validation scheme to guarantee an honest and fair comparison. The learning of the model structures will be done by hill-climbing optimization, using the aforementioned scores and an empty initial structure. Parameters will be learned with Bayesian estimation using hyperparameters $\lambda_{c_j}^{pa(C_i)} = 1$, $\alpha_{x_j,x_z}^{pa(X_i)} = 1$ and $\tau_{x_j}^{pa(X_i)} = 0.001$, which were found to provide interesting results.

All experiments were run on an Intel Core i7-7700K at 4.20GHz with 32 GB of RAM using Windows 10 operating system. The Multi-CTBNC and CTBNC were developed in Java and the software and data sets are freely available at https://github.com/carlvilla/Multi-CTBNCs.

7.1 | Data sets

The proposed model will be first evaluated over randomly generated synthetic data. Then, a real-world data set from an industrial machine (from now on referred to as energy data set) is used to prove the usefulness of the model in a real-world scenario.

7.1.1 | Synthetic data sets

Fifty synthetic time series data sets were randomly generated from the structures of Figures 1A and 3A–I (five data sets per structure). The objective is to compare the performance of the Multi-CTBNC with independent CTBNCs when the data is obtained under diverse conditions. The class variables (C_1, C_2, C_3, C_4, and C_5) are assumed ternary, while the feature variables (X_1, X_2, X_3, X_4, and X_5) can take four values, so they have three possible transitions from a certain state.

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The data sets are sampled via probabilistic logic sampling[51] from random CPTs and CIMs. To sample a sequence, a class configuration obtained from a BN and an initial observation are first defined for the sequence. From this we can sample the time that each feature variable will remain in its current state. Therefore, this time gives the order in which the transitions of the features will occur, obtaining a new observation for the sequence after the transition of only one of them. The new state to which the feature variable transitions is sampled by taking into account the current state of its parent features, as well as the classes of the parent class variables. Once the transition is done, a duration time for the new state of the feature is sampled and the above process will be carried on until a sequence of a predefined duration is obtained. In our case, each data set contains 10,000 sequences that last a bit more than 10 time units (an average of 164 transitions).

7.1.2 | Energy data set

The energy data set contains electrical measurements extracted in collaboration with a partner company from an industrial machine working in a real environment. These variables include intensity (I), voltage (V), active power (P), reactive power (Q), and apparent power (S), which were observed at a sampling frequency of 500 Hz and discretized into 30 states with an equal width discretization. The industrial machine is composed of different three-phase motors and, therefore, for each energy variable a measure in each of the three phases (A, B, and C) was obtained. In total, the data set has 15 feature variables.

The task to perform with the energy data set is to classify the power consumption state (high, low, or inactive) of six motors (M1, M2, M3, M4, M5, and M6), which constitute six class variables, by using the energy consumption of the machine as a whole. It is important to note that these motors are related to each other, as M1 and M2, as well as M5 and M6, work together on very similar tasks. At the same time, M3 and M4 work synchronously with the motor pairs M1/M2 and M5/M6, respectively.

As in a real application we cannot obtain sequences where the motors have a unique state for all transitions, the extracted training sequences have a fixed duration of 0.3 s, determined based on the needs of the company, and the consumption state assigned for the motors is the one that occurs the most.

7.2 | Performance evaluation metrics

Evaluation metrics for multidimensional classifiers should consider, simultaneously, the performance of the model on multiple class variables. Although the literature on this topic is limited, several evaluation metrics have been already proposed for this context:

- Global accuracy:[47] ratio of sequences that were correctly classified for all class variables, that is, a partially correct classification is considered as an error:

$$Acc = \frac{1}{N}\sum_{l=1}^{N} \delta\left(\mathbf{c}'_l, \mathbf{c}_l\right),$$

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where $\mathbf{c}'_l$ and $\mathbf{c}_l$ are the predicted and actual classes of sequence l, respectively, and $\delta(\cdot,\cdot)$ is the Kronecker's delta function, so $\delta(\mathbf{c}'_l, \mathbf{c}_l) = 1$ if $\mathbf{c}'_l = \mathbf{c}_l$ and 0 otherwise.

- Mean accuracy:[47] mean of the accuracies obtained for each class variable separately:

$$\overline{Acc} = \frac{1}{d}\sum_{y=1}^{d} Acc_y = \frac{1}{d}\sum_{y=1}^{d}\frac{1}{N}\sum_{l=1}^{N}\delta\left(c'_{ly}c_{ly}\right),$$

where Acc_y is the accuracy for class variable C_y.

- Global Brier score:[52] it measures the accuracy of probabilistic classifiers by considering the probability that assigns to multidimensional predictions:

$$Bs = \frac{1}{N}\sum_{l=1}^{N}\sum_{g=1}^{|\mathcal{I}|}\left(p\left(\mathbf{C} = \mathbf{c}_g|\mathbf{x}_l^{t_1}, ..., \mathbf{x}_l^{t_{\eta_l}}\right) - \delta(\mathbf{c}_g, \mathbf{c}_l)\right)^2,$$

where $\mathcal{I} = \Omega_{C_1} \times \cdots \times \Omega_{C_d}$ is the space of joint configurations of the class variables.

- F_1 score: harmonic mean of the precision (P) and recall (R) on a class c_j:

$$F_1 = 2\frac{PR}{P+R} = 2\frac{tp_{c_j}}{2tp_{c_j} + fp_{c_j} + fn_{c_j}},$$

where tp_{c_j}, fp_{c_j}, and fn_{c_j} are the counts for true positives, false positives and false negatives, respectively, for class c_j.

Traditional equations for precision, recall and, therefore, F_1 score can only be used for a unique binary class variable. However, Gil-Begue et al.[53] extended them for multiple, possibly nonbinary, class variables. Let B be a function that computes any of these evaluation metrics by receiving a confusion matrix, then the metric scores are obtained with macro- and micro-averaging as follows:

- Macro-averaging: averages the scores of each class variable:

$$B_{macro} = \frac{1}{d}\sum_{y}^{d} B_{C_y}, \quad \text{where } B_{C_y} = \begin{cases} \frac{1}{|\Omega_{C_y}|}\sum_{c_j} B(tp_{c_j}, fp_{c_j}, tn_{c_j}, fn_{c_j}), & \text{if } |\Omega_{C_y}| > 2 \\ B(tp_{C_y}, fp_{C_y}, tn_{C_y}, fn_{C_y}) & \text{otherwise.} \end{cases}$$

If the class variable C_y is binary, only the confusion matrix for one of its classes $(tp_{C_y}, fp_{C_y}, tn_{C_y}, fn_{C_y})$ is considered.

- Micro-averaging: aggregates the confusion matrices of each class variable:

$$B_{micro} = B\left(\sum_{y=1}^{d} TP_{C_y}, \sum_{y=1}^{d} FP_{C_y}, \sum_{y=1}^{d} TN_{C_y}, \sum_{y=1}^{d} FN_{C_y}\right),$$

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where

$$\{TP_{C_y}, FP_{C_y}, TN_{C_y}, FN_{C_y}\} = \begin{cases} \frac{1}{|\Omega_{C_y}|} \sum_{c_j} \{tp_{c_j}, fp_{c_j}, tn_{c_j}, fn_{c_j}\}, & \text{if } |\Omega_{C_y}| > 2 \\ \{tp_{C_y}, fp_{C_y}, tn_{C_y}, fn_{C_y}\}, & \text{otherwise.} \end{cases}$$

Only the macro-averaging approach will be considered in our experiments to calculate the F_1 score, as the micro-averaging is equivalent to the mean accuracy when the cardinality of all class variables is the same and greater than two. As a false positive for a certain class is, at same time, a false negative for another when a multiclass class variable C_y is considered, that is, $\sum_{c_j} fp_{c_j} = \sum_{c_j} fn_{c_j}$, both the micro-averaged precision and recall for C_y are equivalent and, therefore, equal to the micro-averaged F_1 score. Given that the micro-averaged precision and the accuracy for C_y can be computed as follows:

$$P_{micro_y} = \frac{\sum_{c_j} tp_{c_j}}{\sum_{c_j} tp_{c_j} + \sum_{c_j} fp_{c_j}} = \frac{\sum_{c_j} tp_{c_j}}{N} = Acc_y,$$

then we can infer that the mean accuracy of multiclass class variables with the same cardinality is equivalent to the micro-averaged F_1 score:

$$F_{1_{micro}} = \frac{\sum_{y=1}^{d} \frac{1}{|\Omega_{C_y}|} \sum_{c_j} tp_{c_j}}{\sum_{y=1}^{d} \frac{1}{|\Omega_{C_y}|} \sum_{c_j} tp_{c_j} + \sum_{y=1}^{d} \frac{1}{|\Omega_{C_y}|} \sum_{c_j} fp_{c_j}} = \frac{\sum_{y=1}^{d} \sum_{c_j} tp_{c_j}}{\sum_{y=1}^{d} \left[\sum_{c_j} tp_{c_j} + \sum_{c_j} fp_{c_j} \right]}$$
$$= \frac{\sum_{y=1}^{d} \sum_{c_j} tp_{c_j}}{dN} = \frac{1}{d} \sum_{y=1}^{d} \frac{\sum_{c_j} tp_{c_j}}{N} = \overline{Acc}.$$

Note that the normalization factors $1/|\Omega_{C_y}|$ cancel each other out since the cardinality is assumed to be the same for each class variable. Otherwise, this equality would not hold.

7.3 | Results

7.3.1 | Synthetic data set

Average results obtained from fivefold cross-validations over the five data sets generated from each structure are shown in Tables 2–4. Each table includes the performance of the models when they are learned with different scores. The mean and standard deviation for each evaluation metric are reported and the best results are written in bold.

Interesting results have been obtained in all experiments except with the CLL score (Table 3). This score did not lead to the expected models, at least in the performed experiments, since most learned structures have an empty bridge subgraph even when no penalization on their complexity is applied. The reason is the difference between the likelihood and denominator terms, as the latter tends to be larger (and therefore the score smaller) with the inclusion of dependencies

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 Estimated evaluation metrics (mean ± std. deviation) over the synthetic data sets when learning the models with the BIC score

	Global accuracy		Mean accuracy		Global Brier score		Macro F_1 score	
Data sets	**CTBNCs**	**Multi-CTBNC**	**CTBNCs**	**Multi-CTBNC**	**CTBNCs**	**Multi-CTBNC**	**CTBNCs**	**Multi-CTBNC**
Figure 1A	0.2305 ± 0.0295	**0.4076 ± 0.0206**	0.7244 ± 0.0210	**0.7891 ± 0.0161**	1.3997 ± 0.0390	**1.0782 ± 0.0382**	0.6840 ± 0.0261	**0.7545 ± 0.0243**
Figure 3A	0.1636 ± 0.0190	**0.2398 ± 0.0391**	0.7007 ± 0.0179	**0.7296 ± 0.0243**	1.4429 ± 0.0752	**0.9096 ± 0.0219**	0.6222 ± 0.0404	**0.6435 ± 0.0523**
Figure 3B	0.2022 ± 0.0619	**0.2919 ± 0.0345**	0.7247 ± 0.0219	**0.7497 ± 0.0116**	1.3493 ± 0.0423	**0.9772 ± 0.0308**	0.6743 ± 0.0183	**0.6994 ± 0.0154**
Figure 3C	0.2384 ± 0.0401	**0.3563 ± 0.0480**	0.7350 ± 0.0233	**0.7735 ± 0.0209**	1.3470 ± 0.0659	**1.1419 ± 0.0745**	0.6890 ± 0.0335	**0.7321 ± 0.0294**
Figure 3D	0.2612 ± 0.0390	**0.3346 ± 0.0204**	0.7395 ± 0.0157	**0.7735 ± 0.0080**	1.3425 ± 0.0715	**1.1825 ± 0.0345**	0.6900 ± 0.0299	**0.7265 ± 0.0287**
Figure 3E	0.2246 ± 0.0421	**0.3200 ± 0.0543**	0.7379 ± 0.0283	**0.7816 ± 0.0292**	1.3743 ± 0.0925	**0.9458 ± 0.0690**	0.6851 ± 0.0121	**0.7113 ± 0.0143**
Figure 3F	0.1226 ± 0.0148	**0.2375 ± 0.0536**	0.6664 ± 0.0075	**0.7377 ± 0.0305**	1.2833 ± 0.1876	**0.9120 ± 0.0239**	0.6050 ± 0.0277	**0.6698 ± 0.0320**
Figure 3G	0.2297 ± 0.0312	**0.3196 ± 0.0188**	0.7257 ± 0.0212	**0.7666± 0.0081**	1.4134 ± 0.0625	**1.2166 ± 0.0413**	0.6720 ± 0.0152	**0.7136 ± 0.0276**
Figure 3H	0.2135 ± 0.0388	**0.3738 ± 0.0369**	0.7253 ± 0.0286	**0.7849 ± 0.0210**	1.4142 ± 0.0726	**1.1374 ± 0.0699**	0.6363 ± 0.0211	**0.7009 ± 0.0151**
Figure 3I	0.2405 ± 0.0981	**0.3365 ± 0.0679**	0.7257 ± 0.0457	**0.7630 ± 0.0336**	1.3841 ± 0.1719	**1.1870 ± 0.1222**	0.6628 ± 0.0133	**0.7054 ± 0.0214**

Abbreviations: BIC, Bayesian information criterion; CTBNC, continuous time Bayesian network classifier.

10981111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 3 Estimated evaluation metrics (mean ± std. deviation) over the synthetic data sets when learning the models with the CLL score with BIC penalization

	Global accuracy		Mean accuracy		Global Brier score		Macro F_1 score	
Data sets	**CTBNCs**	**Multi-CTBNC**	**CTBNCs**	**Multi-CTBNC**	**CTBNCs**	**Multi-CTBNC**	**CTBNCs**	**Multi-CTBNC**
Figure 1A	0.0442 ± 0.0348	**0.0794 ± 0.0276**	**0.5214 ± 0.0494**	0.4890 ± 0.0636	0.9854 ± 0.0065	**0.9726 ± 0.0079**	**0.2256 ± 0.0135**	0.2144 ± 0.0188
Figure 3A	0.1100 ± 0.0723	**0.1210 ± 0.0679**	**0.5777 ± 0.0891**	0.5514 ± 0.1130	0.9661 ± 0.0367	**0.9499 ± 0.0342**	**0.2398 ± 0.0216**	0.2303 ± 0.0305
Figure 3B	0.0613 ± 0.0376	**0.1016 ± 0.0525**	**0.5180 ± 0.0167**	0.5112 ± 0.0184	0.9861 ± 0.0027	**0.9695 ± 0.0177**	**0.2251 ± 0.0053**	0.2229 ± 0.0053
Figure 3C	0.0828 ± 0.0713	**0.0954 ± 0.0594**	**0.5385 ± 0.0648**	0.5193 ± 0.0827	0.9819 ± 0.0125	**0.9708 ± 0.0173**	**0.2304 ± 0.0179**	0.2236 ± 0.0244
Figure 3D	0.0601 ± 0.0521	**0.1035 ± 0.0327**	**0.5185 ± 0.0566**	0.5034 ± 0.0628	0.9851 ± 0.0090	**0.9654 ± 0.0079**	**0.2248 ± 0.0156**	0.2197 ± 0.0173
Figure 3E	0.0631 ± 0.0566	**0.0865 ± 0.0392**	**0.5397 ± 0.0687**	0.5283 ± 0.0795	0.9810 ± 0.0120	**0.9695 ± 0.0150**	**0.2292 ± 0.0182**	0.2250 ± 0.0227
Figure 3F	0.0283 ± 0.0145	**0.0600 ± 0.0061**	**0.4974 ± 0.0391**	0.4776 ± 0.0374	0.9893 ± 0.0028	**0.9813 ± 0.0017**	**0.2202 ± 0.0117**	0.2133 ± 0.0116
Figure 3G	0.0349 ± 0.0451	**0.0714 ± 0.0150**	**0.4948 ± 0.0536**	0.4808 ± 0.0661	0.9875 ± 0.0051	**0.9746 ± 0.0062**	**0.2184 ± 0.0149**	0.2138 ± 0.0191
Figure 3H	0.0646 ± 0.0131	0.0646 ± 0.0131	0.5941 ± 0.0190	0.5941 ± 0.0190	0.9745 ± 0.0030	0.9745 ± 0.0030	0.2454 ± 0.0053	0.2454 ± 0.0053
Figure 3I	0.0807 ± 0.1394	**0.1317 ± 0.1215**	**0.5460 ± 0.0959**	0.5343 ± 0.0959	0.9747 ± 0.0255	**0.9513 ± 0.0498**	**0.2318 ± 0.0253**	0.2278 ± 0.0255

Abbreviations: BIC, Bayesian information criterion; CLL, conditional log-likelihood; CTBNC, continuous time Bayesian network classifier.

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 4 Estimated evaluation metrics (mean ± std. deviation) over the synthetic data sets when learning the models with the BDe score

	Global accuracy		Mean accuracy		Global Brier score		Macro F_1 score	
Data sets	**CTBNCs**	**Multi-CTBNC**	**CTBNCs**	**Multi-CTBNC**	**CTBNCs**	**Multi-CTBNC**	**CTBNCs**	**Multi-CTBNC**
Figure 1A	0.2302 ± 0.0413	**0.4159 ± 0.0213**	0.7288 ± 0.0254	**0.7938 ± 0.0157**	1.3726 ± 0.0550	**1.0592 ± 0.0397**	0.6903 ± 0.0322	**0.7602 ± 0.0232**
Figure 3A	0.1684 ± 0.0178	**0.2411 ± 0.0384**	0.7049 ± 0.0166	**0.7317 ± 0.0236**	1.4069 ± 0.0712	**0.9065 ± 0.0211**	0.6256 ± 0.0349	**0.6467 ± 0.0535**
Figure 3B	0.1992 ± 0.0595	**0.2889 ± 0.0218**	0.7265 ± 0.0208	**0.7489 ± 0.0066**	1.2942 ± 0.0899	**0.9753 ± 0.0150**	0.6740 ± 0.0223	**0.6996 ± 0.0117**
Figure 3C	0.2407 ± 0.0420	**0.3568 ± 0.0535**	0.7374 ± 0.0244	**0.7744 ± 0.0233**	1.3309 ± 0.0604	**1.1403 ± 0.0827**	0.6930 ± 0.0331	**0.7352 ± 0.0317**
Figure 3D	0.2742 ± 0.0414	**0.3320 ± 0.0221**	0.7501 ± 0.0157	**0.7733 ± 0.0096**	1.3114 ± 0.0728	**1.1823 ± 0.0396**	0.7014 ± 0.0306	**0.7262 ± 0.0297**
Figure 3E	0.2299 ± 0.0436	**0.3232 ± 0.0578**	0.7409 ± 0.0288	**0.7832 ± 0.0307**	1.3542 ± 0.1018	**0.9392 ± 0.0731**	0.6890 ± 0.0137	**0.7138 ± 0.0103**
Figure 3F	0.1420 ± 0.0335	**0.2397 ± 0.0539**	0.6809 ± 0.0287	**0.7399 ± 0.0300**	1.2423 ± 0.2133	**0.9092 ± 0.0252**	0.6047 ± 0.0289	**0.6726 ± 0.0340**
Figure 3G	0.2348 ± 0.0266	**0.3305 ± 0.0285**	0.7328 ± 0.0189	**0.7735 ± 0.0127**	1.3907 ± 0.0585	**1.1864 ± 0.0570**	0.6814 ± 0.0149	**0.7225 ± 0.0274**
Figure 3H	0.2207 ± 0.0376	**0.3755 ± 0.0347**	0.7316 ± 0.0292	**0.7860 ± 0.0199**	1.3878 ± 0.0662	**1.1318 ± 0.0658**	0.6463 ± 0.0213	**0.7033 ± 0.0149**
Figure 3I	0.2405 ± 0.0980	**0.3395 ± 0.0690**	0.7300 ± 0.0432	**0.7669 ± 0.0347**	1.3716 ± 0.1694	**1.1773 ± 0.1265**	0.6702 ± 0.0139	**0.7105 ± 0.0211**

Abbreviations: BDe, Bayesian Dirichlet equivalent; CTBNC, continuous time Bayesian network classifier.

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

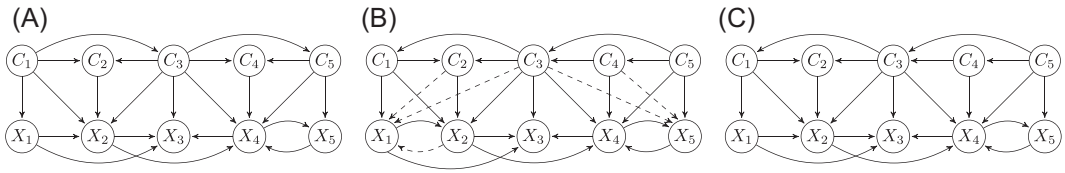

FIGURE 4 Original and learned structures with different scores. (A) Original structure, (B) BIC score, and (C) BDe score. Extra dependencies are represented by dashed arcs. BDe, Bayesian Dirichlet equivalent; BIC, Bayesian information criterion

in this subgraph. The CLL score then favors very sparse or even empty structures. Consequently, the following analysis will focus on the results achieved with the BIC and BDe scores.

As we can see in Tables 2 and 4, the Multi-CTBNC outperforms the independent CTBNCs in all synthetic experiments using both the BIC and BDe scores. As averages are susceptible to outliers, the Wilcoxon signed-rank test was applied over all the results of the 50 data sets to verify that there are statistical differences in the performance of the Multi-CTBNC and the CTBNCs. This results in p-values smaller than 0.001 for both scores in all evaluation metrics. Therefore, the differences are highly significant and the null hypothesis that both methods perform equally well is rejected in favor of our new model. The Multi-CTBNC has an important advantage in those contexts where class variables have a very weak or nonexistent relationship with the features that we were able to collect. This can be seen in some class variables from Figures 3A,B,E,F. The reason for this is that the Multi-CTBNC can model the dependencies of those class variables with others, while the CTBNCs may only rely on their prior probabilities.

The influence of the scores on the model performance was also studied and significant differences were found. If we compare the results of the Multi-CTBNC in Tables 2 and 4, a slight improvement is generally appreciated with the BDe score. This was verified for all evaluation metrics with the Wilcoxon signed-rank test and a significance level of 0.005. We observed in our experiments that this score tends to be more robust to data overfitting and to better reconstruct the original structures. As an example, Figure 4 shows how the BIC creates a slightly denser structure from a data set of Figure 1A, while the BDe reports a more accurate result.

Results obtained with the data sets from Figure 3H are of special interest since there are no dependencies among the class variables and a more even result between the independent classifiers and the Multi-CTBNC was expected. However, the Multi-CTBNC significantly improves all metrics while, in most cases, correctly does not define any association between class variables. The fact of knowing the simultaneous dependencies of the features on different class variables allows the Multi-CTBNC to learn more accurate models. For this same reason, even an empty-digraph Multi-CTBNC may obtain better results than independent CTBNCs. Table 5 shows the performance of an empty-digraph Multi-CTBNC on the synthetic data sets, highlighting in bold the results that improve those of the CTBNCs (Table 4). Significant improvements were found in all evaluation metrics, except the macro-averaged F_1 score, with the Wilcoxon signed-rank test and a significance level of 0.001. This justifies the use of Multi-CTBNCs even when it is clear that there are no dependencies between the class variables.

Finally, Table 6 shows the learning times for the Multi-CTBNC and the CTBNCs when they are built with the BIC and BDe scores. The learning of a Multi-CTBNC is considerably faster than multiple CTBNCs when using the proposed data sets. Additionally, the BIC is significantly more time consuming since it tends to build denser structures (see Figure 4). These differences are statistically significant with the Wilcoxon signed-rank test and a significance level of 0.001. Note that the learning of both a CTBNC and a Multi-CTBNC, as well as the different CTBNCs, is performed in parallel.

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

7.3.2 | Energy data set

Due to the high cardinality of the energy data set feature variables, their nodes are limited to have at most one feature as a parent. Therefore, this section compares the differences in performance between independent max1 CTBNCs and a DAG-max1 Multi-CTBNC.

Fifteen data sets with different sequence order were extracted from the original energy data set and a fivefold cross-validation was performed on each of them. The average results of the cross-validations are shown in Table 7, where the DAG-max1 Multi-CTBNC outperforms the independent classifiers in all evaluation metrics. In this occasion, the most interesting models were obtained with the LL penalized with BIC, unlike with the synthetic data sets. This is because the BDe defines empty bridge subgraphs, so the classification is performed without

TABLE 5 Estimated evaluation metrics (mean ± std. deviation) over the synthetic data sets when learning an empty-digraph Multi-CTBNC with the BDe score

Data sets	Global accuracy	Mean accuracy	Global Brier score	Macro F_1 score
Figure 1A	**0.3986 ± 0.0189**	**0.7864 ± 0.0144**	**1.0912 ± 0.0346**	**0.7517 ± 0.0219**
Figure 3A	0.1456 ± 0.0272	0.7027 ± 0.0252	**0.9486 ± 0.0143**	0.5426 ± 0.0350
Figure 3B	0.1812 ± 0.0156	**0.7316 ± 0.0056**	**1.0001 ± 0.0220**	0.6422 ± 0.0127
Figure 3C	**0.3316 ± 0.0540**	**0.7662 ± 0.0240**	**1.1817 ± 0.0831**	**0.7265 ± 0.0322**
Figure 3D	**0.2935 ± 0.0124**	**0.7597 ± 0.0074**	**1.2501 ± 0.0246**	**0.7124 ± 0.0269**
Figure 3E	**0.2566 ± 0.0897**	**0.7639 ± 0.0479**	**0.9574 ± 0.0585**	0.6611 ± 0.0312
Figure 3F	**0.1859 ± 0.0514**	**0.7298 ± 0.0329**	**0.9351 ± 0.0229**	0.5991 ± 0.0185
Figure 3G	**0.3011 ± 0.0265**	**0.7656 ± 0.0120**	**1.2380 ± 0.0513**	**0.7144 ± 0.0249**
Figure 3H	**0.3755 ± 0.0346**	**0.7860 ± 0.0198**	**1.1319 ± 0.0657**	**0.7032 ± 0.0147**
Figure 3I	**0.3135 ± 0.0622**	**0.7579 ± 0.0341**	**1.2234 ± 0.1133**	**0.7010 ± 0.0229**

Abbreviations: BDe, Bayesian Dirichlet equivalent; CTBNC, continuous time Bayesian network classifier.

TABLE 6 Model learning times in seconds (mean std. deviation) on all synthetic data sets

CTBNCs (BIC)	CTBNCs (BDe)	Multi-CTBNC (BIC)	Multi-CTBNC (BDe)
29.29 ± 2.76	24.25 ± 1.99	15.89 ± 1.82	**14.17 ± 2.38**

Abbreviations: BDe, Bayesian Dirichlet equivalent; BIC, Bayesian information criterion; CTBNC, continuous time Bayesian network classifier.

TABLE 7 Estimated evaluation metrics (mean std. deviation) over the energy data set when learning with the BIC score

Model	Global accuracy	Mean accuracy	Global Brier score	Macro F_1 score
max1 CTBNCs	0.6897 ± 0.0033	0.8600 ± 0.0015	0.5377 ± 0.0056	0.8005 ± 0.0034
DAG-max1 Multi-CTBNC	**0.7412 ± 0.0022**	**0.8622 ± 0.0017**	**0.4672 ± 0.0039**	**0.8151 ± 0.0037**

Abbreviations: CTBNC, continuous time Bayesian network classifier; BIC, Bayesian information criterion.

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

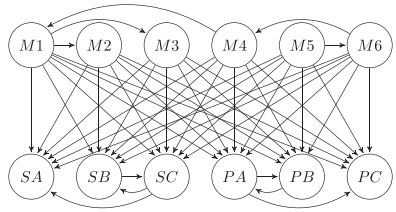

FIGURE 5 Structure of a DAG-max1 Multi-CTBNC learned from the energy data set. CTBNC, continuous time Bayesian network classifier; DAG, directed acyclic graph

considering the feature variables. Wilcoxon signed rank tests show significant differences for all evaluation metrics with a significance level of 0.001.

Figure 5 shows an example of a Multi-CTBNC structure learned on the energy data set. Unsurprisingly, all class variables have the same six features as children since they all represent similar motors. The remaining energy features are not shown as the model does not consider them for classification. The differences of the class variables lie in their influence on the total consumption of the machine and their relationships with each other. Regarding the latter, the results are very accurate since most dependencies match the description of Section 7.1.2. In this problem, the differential factor that makes the Multi-CTBNC perform better than independent CTBNCs is its ability to model these dependencies. Expected relationships among feature variables are also detected since they involve measurements over the three different phases of the electrical system. The reason for this is that the current and voltage are balanced, with the same magnitudes and 120 degrees displacement, between each phase. Nonetheless, this balance could be affected in terms of magnitude and/or phase angular displacement by the type of the electrical consumers (e.g., single phase elements) or anomalous behaviors (e.g., failure of electronic components, such as, phase diode, fuse or isolation). The advantage of using a graphical model is that these machine malfunctions could be easily detected by analyzing the learned structure.

8 | CONCLUSIONS AND FUTURE RESEARCH

This article introduces the Multi-CTBNC, a new probabilistic graphical model that is able to capture the dependencies of multivariate temporal sequences and perform multidimensional classification over them. The learning of its parameters and structure from data is discussed, and its usefulness is justified by solving a novel real-world engineering problem.

Score-based learning algorithms for Multi-CTBNCs were studied, as it is the common approach for learning CTBNs. However, a constraint-based algorithm for CTBNs has been recently proposed by Bregoli et al.[54] where it was experimentally shown to perform better in certain scenarios. Therefore, its extension to learn classification models and, more specifically, the use of its fundamentals to develop a constraint-based algorithm for Multi-CTBNCs are possible lines of research.

The adaptation of Multi-CTBNCs to handle other types of predictive problems where data and sources have different characteristics is currently being studied. First, an important limitation of the presented model is that the feature variables are assumed to be discrete, while the problem of discretizing time series still requires more attention.[39] Nonetheless, we would like to study a possible adaptation of Multi-CTBNCs (and therefore CTBNs) for continuous

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

variables, thus avoiding this discretization. Second, this study assumes no missing data or hidden variables, that is, complete data assumption, which in practice is not always the case. Third, the classifier can easily use other distributions to model the waiting times of variables in a certain state. This is the case of hypoexponential distributions, previously used with CTBNs,[55] which are more appropriate than the exponential for certain applications.

The study of structure constraints for CTBNs has not received a lot of attention and, to the best of our knowledge, only the learning of naive Bayes and structures where nodes have a maximum number of parents were considered. It could be interesting to analyze the possible benefits in terms of computational complexity and performance of learning CTBNs with other constraints, such as being trees or DAGs.

Finally, although a really limited number of class variables is commonly assumed, multiple applications stand out for having a large number of them. For example, the industrial problem introduced in this article could require classifying many more motors. Therefore, we plan to study the inclusion of feature selection approaches to the Multi-CTBNC to make classification under these conditions less prohibitive. This would allow to include restrictions over the complexity of the bridge subgraph. With a similar objective, it would be interesting to design class-bridge decomposable[47] Multi-CTBNCs to reduce the computational cost of estimating the most probable class configuration of a sequence.

ACKNOWLEDGMENTS

This study has been partially supported by the Spanish Ministry of Science, Innovation and Universities through the PID2019-109247GB-I00 and RTC2019-006871-7 projects, and by the project BAYES-CLIMA-NEURO, from the BBVA Foundation (2019). Aingura IIot has contributed to this project by providing expert knowledge in electrical systems and real-world data for our experiments. C. Villa-Blanco is supported by a predoctoral contract for the formation of doctors from the Universidad Politécnica de Madrid.

ENDNOTES

*Sequences may have different timestamps, superscript l is omitted from t for simplicity.

†The state domain of each variable can be different, but the subscript i is omitted from k for simplicity.

ORCID

Carlos Villa-Blanco https://orcid.org/0000-0001-9333-8569
Pedro Larrañaga https://orcid.org/0000-0003-0652-9872
Concha Bielza https://orcid.org/0000-0001-7109-2668

REFERENCES

1. Dietrich C, Palm G, Riede K, Schwenker F. Classification of bioacoustic time series based on the combination of global and local decisions. *Pattern Recognit.* 2004;37(12):2293-2305.
2. Fulcher BD, Jones NS. Highly comparative feature-based time-series classification. *IEEE Trans Knowl Data Eng.* 2014;26(12):3026-3037.
3. Jeong Y-S, Jeong MK, Omitaomu OA. Weighted dynamic time warping for time series classification. *Pattern Recognit.* 2011;44(9):2231-2240.
4. Moskovitch R, Shahar Y. Classification-driven temporal discretization of multivariate time series. *Data Min Knowl Discov.* 2015;29(4):871-913.
5. Xu D, Shi Y, Tsang IW, Ong Y-S, Gong C, Shen X. Survey on multi-output learning. *IEEE Trans Neural Netw Learn Syst.* 2019;31(7):2409-2429.

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

6. Read J, Pfahringer B, Holmes G, Frank E. Classifier chains for multi-label classification. *Mach Learn.* 2011; 85(3):333-359.
7. Jia B-B, Zhang M-L. Multi-dimensional classification via kNN feature augmentation. *Pattern Recognit*. 2020;106:Art. no. 107423.
8. Clare A, King RD. Knowledge discovery in multi-label phenotype data. Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery; 2001:42-53.
9. Elisseeff A, Weston J. A kernel method for multi-labelled classification. Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic; 2001:681-687.
10. Zhang M-L, Zhou Z-H. ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recognit*. 2007; 40(7):2038-2048.
11. van der Gaag LC, de Waal PR. Multi-dimensional Bayesian network classifiers. Proceedings of the 3rd European Workshop on Probabilistic Graphical Models; 2006:107-114.
12. Zhang M-L, Zhou Z-H. A review on multi-label learning algorithms. *IEEE Trans Knowl Data Eng.* 2013; 26(8):1819-1837.
13. Nanopoulos A, Alcock R, Manolopoulos Y. Feature-based classification of time-series data. *Int J Comput Res.* 2001;10(3):49-61.
14. Berndt DJ, Clifford J. Using dynamic time warping to find patterns in time series. Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining; 1994:359-370.
15. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput.* 1997;9(8):1735-1780.
16. Lipton ZC, Kale DC, Elkan C, Wetzel R. Learning to diagnose with LSTM recurrent neural networks. Proceedings of the 4th International Conference on Learning Representations; 2016.
17. Batista GEAPA, Wang X, Keogh EJ. A complexity-invariant distance measure for time series. Proceedings of the 11th SIAM International Conference on Data Mining; 2011:699-710.
18. Dean T, Kanazawa K. A model for reasoning about persistence and causation. *Comput Intell.* 1989;5(2): 142-150.
19. Murphy KP. *Dynamic Bayesian networks: Representation, inference and learning*. Ph.D. thesis, UC Berkeley; 2002.
20. Nodelman U, Shelton CR, Koller D. Continuous time Bayesian networks. Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence; 2002:378-387.
21. Stella F, Amer Y. Continuous time Bayesian network classifiers. *J Biomed Inf.* 2012;45(6):1108-1119.
22. Beinlich IA, Suermondt HJ, Chavez RM, Cooper GF. The ALARM monitoring system: a case study with two probabilistic inference techniques for belief networks. Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine; 1989:247-256.
23. Correa M, Bielza C, Ramírez MJ, Alique JR. A Bayesian network model for surface roughness prediction in the machining process. *Int J Syst Sci.* 2008;39(12):1181-1192.
24. DeFelipe J, López-Cruz PL, Benavides-Piccione R, et al. New insights into the classification and nomenclature of cortical GABAergic interneurons. *Nat Rev Neurosci.* 2013;14(3):202-216.
25. Luo J, Savakis AE, Singhal A. A Bayesian network-based framework for semantic image understanding. *Pattern Recognit*. 2005;38(6):919-934.
26. Bielza C, Larrañaga P. Discrete Bayesian network classifiers: a survey. *ACM Comput Surv.* 2014;47(1):1-43.
27. Heckerman D. A tutorial on learning with Bayesian networks. In: Holmes DE, Jain LC, eds. *Innovations in Bayesian Networks: Theory and Applications*. Berlin/Heidelberg: Springer; 2008:33-82.
28. Burge J, Lane T, Link H, Qiu S, Clark VP. Discrete dynamic Bayesian network analysis of fMRI data. *Hum. Brain Mapp.* 2009;30(1):122-137.
29. Perrin B-E, Ralaivola L, Mazurie A, Bottani S, Mallet J, d'Alché-Buc F. Gene networks inference using dynamic Bayesian networks. *Bioinformatics*. 2003;19(suppl_2):ii138-ii148.
30. Zweig G, Russell S. Speech recognition with dynamic Bayesian networks. Proceedings of the 15th National Conference on Artificial Intelligence and 10th Innovative Applications of Artificial Intelligence Conference;1998:173-180.
31. Koller D, Friedman N. *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: The MIT Press; 2009.
32. Nodelman U, Shelton CR, Koller D. Learning continuous time Bayesian networks. Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence; 2003:451-458.

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

33. Murphy KP. *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: The MIT Press; 2012.
34. Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: The combination of knowledge and statistical data. *Mach Learn.* 1995;20(3):197-243.
35. Yang L, Shami A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*. 2020;415:295-316.
36. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. *J Mach Learn Res.* 2012;13(10):281-305.
37. Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. Proceedings of the 25th International Conference on Neural Information Processing Systems—Volume 2; 2012:2951-2959.
38. Codecasa D, Stella F. Learning continuous time Bayesian network classifiers. *Int J Approx Reason.* 2014; 55(8):1728-1746.
39. Villa S, Stella F. Learning continuous time Bayesian networks in non-stationary domains. *J Artif Intell Res.* 2016;57:1-37.
40. Bouckaert RR. *Bayesian Belief Networks: From Construction to Inference*. Ph.D. thesis, Utrecht University; 1995.
41. Larrañaga P, Poza M, Yurramendi Y, Murga RH, Kuijpers CMH. Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters. *IEEE Trans Pattern Anal Mach Intell.* 1996;18(9):912-926.
42. Schwarz G. Estimating the dimension of a model. *Ann Stat.* 1978;6(2):461-464.
43. Akaike H. A new look at the statistical model identification. *IEEE Trans Autom Control*. 1974;19(6): 716-723.
44. Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. *Mach Learn.* 1997;29(2):131-163.
45. Cooper GF, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. *Mach Learn.* 1992;9(4):309-347.
46. Andrews B, Ramsey J, Cooper GF. Scoring Bayesian networks of mixed variables. *Int J Data Sci Anal.* 2018; 6(1):3-18.
47. Bielza C, Li G, Larrañaga P. Multi-dimensional classification with Bayesian networks. *Int J Approx Reason*. 2011;52(6):705-727.
48. de Waal PR, van der Gaag LC. Inference and learning in multi-dimensional Bayesian network classifiers. Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty; 2007:501-511.
49. Chickering DM, Geiger D, Heckerman D. *Learning Bayesian networks is NP-hard*. Technical Report MSR-TR-94-17, Microsoft Research; 1994.
50. Chow CK, Liu CN. Approximating discrete probability distributions with dependence trees. *IEEE Trans Inf Theory*. 1968;14(3):462-467.
51. Henrion M. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: Lemmer JF, Kanal LN, eds. *Uncertainty in Artificial Intelligence*, Vol 2. Amsterdam: North-Holland; 1988:149-163.
52. Fernandes JA, Lozano JA, Inza I, Irigoien X, Pérez A, Rodríguez JD. Supervised pre-processing approaches in multiple class variables classification for fish recruitment forecasting. *Environ Model Softw.* 2013;40: 245-254.
53. Gil-Begue S, Bielza C, Larrañaga P. Multi-dimensional Bayesian network classifiers: a survey. *Artif Intell Rev*. 2021;54(1):519-559.
54. Bregoli A, Scutari M, Stella F. Constraint-based learning for continuous-time Bayesian networks. Proceedings of the 10th International Conference on Probabilistic Graphical Models; 2020:41-52.
55. Liu M, Stella F, Hommersom A, Lucas PJ. Making continuous time Bayesian networks more flexible. Proceedings of the 9th International Conference on Probabilistic Graphical Models; 2018:237-248.

How to cite this article: Villa-Blanco C, Larrañaga P, Bielza C. Multidimensional continuous time Bayesian network classifiers. *Int J Intell Syst*. 2021;36:7839-7866. https://doi.org/10.1002/int.22611

1098111x, 2021, 12, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/int.22611 by Universidad Politecnica De, Wiley Online Library on [17/10/2022]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License