Data & Knowledge Engineering 122 (2019) 101–115

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Circular Bayesian classifiers using wrapped Cauchy distributions

Ignacio Leguey [a,b,*], Concha Bielza [b], Pedro Larrañaga [b]

[a] Departamento de Economía Financiera y Contabilidad e Idioma Moderno, Facultad de Ciencias Jurídicas y Sociales, Universidad Rey Juan Carlos de Madrid, Spain

[b] Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain

ARTICLE INFO

Keywords:
Data mining
Classification
Circular statistics
Wrapped Cauchy distribution
Bayesian networks
Cortical layer

ABSTRACT

Capturing the dependences among circular variables within supervised classification models is a challenging task. In this paper, we propose four different supervised Bayesian classification algorithms where the predictor variables follow all circular wrapped Cauchy distributions. For this purpose, we introduce four wrapped Cauchy classifiers. The bivariate wrapped Cauchy distribution is the only bivariate circular distribution whose marginals and conditionals are also wrapped Cauchy distributions, a property that makes it possible to define these models easily. Furthermore, the wrapped Cauchy tree-augmented naive Bayes classifier requires the definition of a conditional circular mutual information measure between variables that follow wrapped Cauchy distributions. Synthetic data is used to illustrate, compare and evaluate the classification algorithms (including a comparison with the Gaussian TAN classifier, decision tree, random forest, multinomial logistic regression, support vector machine and simple neural network), leading to satisfactory predictive results. We also use a real neuromorphological dataset obtained from juvenile rat somatosensory cortex cells, where we measure the bifurcation angles of the dendritic basal arbors.

1. Introduction

Circular data is ubiquitous, present in many different areas such as biology, geology, medicine, oceanography, geophysics, meteorology, astronomy, ecology, neuroscience and geography. Some examples are directions of flight of homing pigeons [1], characterization of the phenology of species [2], formation of feldspar laths in basalt rocks [3], paleomagnetism in red slits and claystones [4,5], also in political sciences studying the gun crimes occurred in an specific period of time [6], directional word vectors in text mining [7], wildfire orientation in order to prevent fire propagation [8], wind and waves direction analysis [9,10], study and prediction of protein dihedral angles structure [11,12], and neuronal basal dendritic bifurcation angles analysis [13,14] among many others. The natural periodicity of circular data sometimes makes traditional statistics methods ineffective, since they ignore this characteristic. For instance, when dealing with circular data, 0° and 360° are considered as the same point, whereas if considered non-circular data, they are different points. Thus, circular data analysis is distinct from and more challenging than non-circular data. It should be noted that circular data has been studied extensively [3,15,16].

Probabilistic graphical models [17] are useful tools for data modeling that connect probability theory with graph theory. There are many advantages of using probabilistic graphical models, such as the fact that they are easily interpreted, they handle missing data effectively and they treat inference and learning tasks together. Bayesian networks [18] are one of the most commonly used probabilistic graphical models due to their factorization and domain representation properties. Bayesian networks have the

* Corresponding author at: Departamento de Economía Financiera y Contabilidad e Idioma Moderno, Facultad de Ciencias Jurídicas y Sociales, Universidad Rey Juan Carlos de Madrid, Spain.

E-mail addresses: ignacio.vitoriano@urjc.es (I. Leguey), mcbielza@fi.upm.es (C. Bielza), pedro.larranaga@fi.upm.es (P. Larrañaga).

https://doi.org/10.1016/j.datak.2019.05.005
Received 23 October 2017; Received in revised form 24 November 2018; Accepted 25 May 2019
Available online 28 May 2019
0169-023X/

characteristic that each variable is conditionally independent of those that are non-descendants in the graph given the value of their parents. Therefore the joint probability distribution is expressed as the product of the local distributions conditioned to their parents. For these reasons, Bayesian networks can deal efficiently with supervised classification i.e., the Bayesian network classifiers [13] and offer an explicit, graphical and interpretable representation of uncertain knowledge, which has made it possible to successfully apply them to real-world problems.

Supervised classification [19] deals with the problem of assigning a label to an instance, based on a set of variables that characterize it. Yet circular data has been commonly treated as linear data in supervised classification tasks. Only a few circular classifiers exist, and almost none of them are based on the principles of Bayesian networks, capable of capturing multivariate relationships among variables. Most of them focus on discriminant analysis and assume several circular distributions such as the von Mises distribution [20], later extended to the von Mises–Fisher distribution [21]. There are also circular discriminant analysis studies for the Watson, Selby and Arnold distributions on the sphere [22,23]. SenGupta and Roy [24] used a classification discriminant rule based on the mean chord-length to classify a new observation into one of two different circular populations that are von Mises, when training samples are available for each of them. Also a likelihood ratio test based on a bootstrapping approach for classifying into two populations was proposed for linear and circular data [25]. Kirby and Miranda [26] proposed a variation of a neural network, including a circular node, which was able to keep and send circular information. More recently, Fernandes and Cardoso [27] proposed a binary circular logistic regression as the discriminative counterpart to the naive Bayes model, which does not make assumptions on the input data distribution. López-Cruz et al. [28] is the only study in which Bayesian classifiers were used. For the von Mises and von Mises–Fisher distributions, López-Cruz et al. proposed an adaptation of the naive Bayes classifier and selective naive Bayes classifier, which are two of the simplest and best-known supervised classification models based on Bayesian network principles.

The lack of Bayesian supervised classifiers for circular data is due to the absence of circular Bayesian network models, which are very difficult to develop because of their circular multivariate distribution nature. A family of distributions is said to be closed under marginalization and conditioning when the marginals and conditionals of the multivariate distribution follow the same distribution. However, the marginals and conditionals of most circular distributions do not belong to the same family of distributions, making the modeling phase and posterior inference processes difficult.

The von Mises distribution [29], which is the analogue of the univariate Gaussian distribution, is the best-known circular model. A bivariate von Mises distribution also exists and was introduced by Mardia [30], who subsequently extended it to the multivariate case [31]. He showed that the conditional distributions are also von Mises distributions. Nevertheless, the marginal distributions are either unimodal or bimodal, and only the unimodal case could be approximated to a von Mises distribution when the concentration parameter is large. Therefore, as explained in [13] for discrete distributions, it would be much more complicated to achieve an efficient learning and inference. Therefore, we ruled out the use of von Mises distributions for our particular purpose. Another popular univariate symmetric circular distribution is the wrapped Cauchy, which was introduced by Lévy [32], and further studied by Wintner [33]. It was later obtained by mapping Cauchy distributions onto the circle [34]. Kato and Pewsey [35] developed a five-parameter bivariate wrapped Cauchy distribution for toroidal data, whose marginals and conditionals follow univariate wrapped Cauchy distributions. This family of bivariate wrapped Cauchy distributions is therefore closed under conditioning and marginalization. Leguey et al. [36] proposed a tree-structured Bayesian network model that deals with circular data which follows wrapped Cauchy distribution. However, this model only accounts for the discovery of conditional independence relationships of a set of random variables, without considering any as a class variable. This is a specificity of supervised classification problems that requires special learning algorithms.

Building on previous work regarding supervised classification using Bayesian networks for circular statistics, the novelty of this work lies in the proposal of four circular Bayesian classification models capable of dealing with supervised data following wrapped Cauchy distributions. The models to be presented are called wrapped Cauchy naive Bayes (wCNB), wrapped Cauchy selective naive Bayes (wCsNB), wrapped Cauchy semi-naive Bayes (wCsmNB) and wrapped Cauchy tree-augmented naive Bayes (wCTAN) classifiers. Even though the simplest of our proposals (i.e., wCNB) is a straightforward naive Bayes classifier extension to using wrapped Cauchy distributions, this has never been attempted before to the best of our knowledge.

The remainder of this paper is organized as follows. Section 2 reviews the bivariate wrapped Cauchy distribution of Kato and Pewsey [35]. Section 3 describes the four novel wrapped Cauchy classifiers proposed here. In Section 4, we assess the four models in synthetic domains, requiring the design of a simulation method for these wrapped Cauchy Bayesian network classifiers. Section 5 addresses a real-world neuromorphology data problem using the wrapped Cauchy classifiers. Finally, Section 6 provides concluding remarks and proposals for future work.

2. Wrapped Cauchy distribution

2.1. Definitions

A random variable Θ that follows a wrapped Cauchy distribution [32], denoted $wC(\mu, \varepsilon)$, has a density function

$$f(\theta) = \frac{1}{2\pi} \frac{1-\varepsilon^2}{1+\varepsilon^2-2\varepsilon\cos(\theta-\mu)}, \quad \theta, \mu \in (-\pi, \pi], \varepsilon \in [0, 1) \tag{1}$$

where μ is the mean angle and ε the concentration parameter. f in Eq. (1) is unimodal and symmetric about μ unless $\varepsilon = 0$, which yields the circular uniform distribution (i.e., $f(\theta) = 1/2\pi$).

A five-parameter bivariate wrapped Cauchy distribution was proposed by Kato and Pewsey [35]. A random vector (Θ_1, Θ_2) follows a bivariate wrapped Cauchy distribution, denoted $bwC(\mu_1, \mu_2, \varepsilon_1, \varepsilon_2, \rho)$, if its density function is given by

$$f(\theta_1, \theta_2) = c[c_0 - c_1 \cos(\theta_1 - \mu_1) - c_2 \cos(\theta_2 - \mu_2) - c_3 \cos(\theta_1 - \mu_1)\cos(\theta_2 - \mu_2) - c_4 \sin(\theta_1 - \mu_1)\sin(\theta_2 - \mu_2)]^{-1}, \theta_1, \theta_2 \in (-\pi, \pi], \tag{2}$$

where c, c_0, c_1, c_2, c_3 and c_4 are

$$\begin{aligned} c &= (1-\rho^2)(1-\varepsilon_1^2)(1-\varepsilon_2^2)/4\pi^2 \\ c_0 &= (1+\rho^2)(1+\varepsilon_1^2)(1+\varepsilon_2^2) - 8|\rho|\varepsilon_1\varepsilon_2 \\ c_1 &= 2(1+\rho^2)\varepsilon_1(1+\varepsilon_2^2) - 4|\rho|(1+\varepsilon_1^2)\varepsilon_2 \\ c_2 &= 2(1+\rho^2)(1+\varepsilon_1^2)\varepsilon_2 - 4|\rho|\varepsilon_1(1+\varepsilon_2^2) \\ c_3 &= -4(1+\rho^2)\varepsilon_1\varepsilon_2 + 2|\rho|(1+\varepsilon_1^2)(1+\varepsilon_2^2) \text{ and} \\ c_4 &= 2\rho(1-\varepsilon_1^2)(1-\varepsilon_2^2), \end{aligned}$$

with $\mu_1, \mu_2 \in (-\pi, \pi]$, $\varepsilon_1, \varepsilon_2 \in [0, 1)$ and $\rho \in (-1, 1)$. ε_1 and ε_2 regulate the concentration of the marginal distributions, and ρ is the parameter controlling the association between Θ_1 and Θ_2, from total independence ($\rho = 0$) to perfect correlation ($\rho = \pm 1$). When $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$, f in Eq. (2) is unimodal and pointwise symmetric about (μ_1, μ_2).

From McCullagh [34], we know that representing wrapped Cauchy models in complex form simplifies the computation in many cases. Let $Z = \exp(i\Theta)$, where Θ is distributed as in Eq. (1). Therefore the density function of Z is

$$f(z; \lambda) = \frac{1}{2\pi} \frac{|1 - |\lambda|^2|}{|z - \lambda|^2}, \quad z \in \Omega, \lambda \in \hat{\mathbb{C}} \setminus \Omega, \tag{3}$$

where $\lambda = \varepsilon \exp(i\mu)$, $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ and $\Omega = \{z \in \mathbb{C} : |z| = 1\}$, with $|z|$ the module of the complex number. We use the notation $Z \sim C^*(\lambda)$ to denote that Z is distributed as in Eq. (3).

Similarly, let $(Z_1, Z_2) = (\exp(i\Theta_1), \exp(i\Theta_2))$, where (Θ_1, Θ_2) is distributed as in Eq. (2). Therefore the density of (Z_1, Z_2) is

$$f(z_1, z_2) = \frac{(4\pi^2)^{-1}(1-\rho^2)(1-\varepsilon_1^2)(1-\varepsilon_2^2)}{|a_{11}(\overline{z}_1\eta_1)^q z_2\overline{\eta}_2 + a_{12}(\overline{z}_1\eta_1)^q + a_{21}z_2\overline{\eta}_2 + a_{22}|^2}, z_1, z_2 \in \Omega, \tag{4}$$

where q is the sign of ρ, $\eta_k = \exp(i\mu_k)$ with $k \in \{1, 2\}$, $\overline{z}_k$ is the complex conjugate of z_k, $a_{11} = \varepsilon_1\varepsilon_2 - |\rho|$, $a_{12} = |\rho|\varepsilon_2 - \varepsilon_1$, $a_{21} = |\rho|\varepsilon_1 - \varepsilon_2$, $a_{22} = 1 - |\rho|\varepsilon_1\varepsilon_2$, $\varepsilon_1, \varepsilon_2 \in [0, 1)$, $\rho \in (-1, 1)$ and $\eta_1, \eta_2 \in \Omega$.

Following the complex notation, we denote $(Z_1, Z_2) \sim bC^*(\eta_1, \eta_2, \varepsilon_1, \varepsilon_2, \rho)$ if (Z_1, Z_2) is distributed as in Eq. (4). This five-parameter bivariate wrapped Cauchy complex form representation verifies the following result:

Theorem 1 (*Kato and Pewsey [35]*). *A random vector (Z_1, Z_2) with density given by Eq. (4) has marginals $Z_1 \sim C^*(\varepsilon_1\eta_1)$ and $Z_2 \sim C^*(\varepsilon_2\eta_2)$, and conditionals $Z_1|Z_2 = z_2 \sim C^*(-\eta_1[\boldsymbol{A} \circ (z_2\overline{\eta}_2)^q])$ and $Z_2|Z_1 = z_1 \sim C^*(-\eta_2[\boldsymbol{A}^T \circ (z_1\overline{\eta}_1)^q])$, where $\boldsymbol{A}$ is the matrix with elements a_{11}, a_{12}, a_{21} and a_{22} defined in Eq. (4), $\boldsymbol{A}^T$ is the transpose of $\boldsymbol{A}$, and*

$$\boldsymbol{A} \circ z = \frac{a_{11}z + a_{12}}{a_{21}z + a_{22}}.$$

As far as we know, there is no other bivariate circular distribution for which conditional and marginal distributions belong to the same family. Therefore, wrapped Cauchy is the best choice given no better alternative, as the requirements for the classifier structures that we will develop are of at most a tree-structure (i.e., only bivariate, marginal and conditional densities are required). Furthermore, we require the definition of a conditional circular mutual information measure between variables that follow wrapped Cauchy distributions.

2.2. Parameter estimation

Working with the density given by Eq. (2), numerical methods have to be used to find the parameter estimates, since there is no closed-form expression for the maximum likelihood estimates. Kato and Pewsey [35] demonstrated that the method of moments [37] is more efficient; it is computationally very fast, easy to implement and with closed form formulas for the parameter estimates.

Let $\{(\theta_{1j}, \theta_{2j}), j = 1, \ldots, N\}$ be a random sample from a $bwC(\mu_1, \mu_2, \varepsilon_1, \varepsilon_2, \rho)$ as stated in Eq. (2). Therefore the estimators obtained from the method of moments for $\mu_1, \mu_2, \varepsilon_1, \varepsilon_2$ and ρ are [35]

$$\begin{aligned} &\hat{\mu}_r = arg(\bar{R}_r), \quad \hat{\varepsilon}_r = |\bar{R}_r|, \quad \text{with} \quad \overline{R}_r = \frac{1}{N}\sum_{j=1}^{N} \exp(i\theta_{rj}), \\ &\hat{\rho} = \frac{1}{N}\left(|\sum_{j=1}^{N} \exp(i(\Phi_{1j} - \Phi_{2j}))| - |\sum_{j=1}^{N} \exp(i(\Phi_{1j} + \Phi_{2j}))|\right), \\ &\text{with} \quad \Phi_{rj} = 2\arctan\left(\frac{1+\hat{\varepsilon}_r}{1-\hat{\varepsilon}_r}\tan\left(\frac{\theta_{rj} - \hat{\mu}_r}{2}\right)\right) \quad \text{and} \quad r = 1, 2. \end{aligned} \tag{5}$$

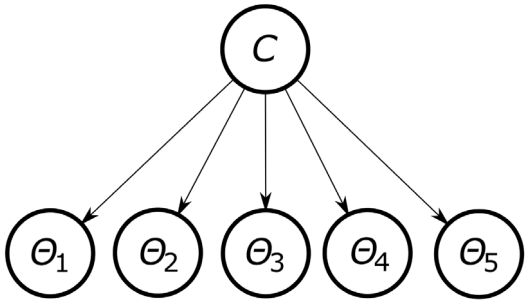

Fig. 1. wCNB structure with five circular predictor nodes, from which $p(c|\boldsymbol{\theta}) \propto p(c) f_{\Theta_1|c}(\theta_1|c) f_{\Theta_2|c}(\theta_2|c) f_{\Theta_3|c}(\theta_3|c) f_{\Theta_4|c}(\theta_4|c) f_{\Theta_5|c}(\theta_5|c)$.

3. Wrapped Cauchy classifiers

Let $\boldsymbol{\Theta} = (\Theta_1, \dots, \Theta_n)$ be a vector of circular predictor random variables or features, and let C be a discrete class variable which takes values (labels) in the set $\Lambda(C)$. Given a sample of N labeled instances $(\boldsymbol{\Theta}^1, C^1), \dots, (\boldsymbol{\Theta}^N, C^N)$, the supervised classification problem consists of developing a model capable of assigning a class label to a new object based on the values of its features.

Bayesian network classifiers [13] have been used to solve classification problems with linear data, because of their easy representation of the problem domain and the efficient computation of the algorithms associated with Bayesian networks techniques. Our novel purpose is to develop the circular domain counterpart of four well-known Bayesian network classifiers (naive Bayes, selective naive Bayes, semi-naive Bayes and tree-augmented naive Bayes) when the underlying variables follow wrapped Cauchy distributions.

3.1. Wrapped Cauchy naive Bayes

The wrapped Cauchy naive Bayes (wCNB) classifier is the simplest of the four Bayesian network classifier models that we present in this paper, where C is the parent of all circular features and these are assumed to be conditionally independent among them given C (Fig. 1)

$$p(C = c|\boldsymbol{\Theta} = \boldsymbol{\theta}) \propto p(C = c) \prod_{i=1}^{n} f_{\Theta_i|C=c}(\theta_i|c). \tag{6}$$

The wCNB determines the class value c^* for a new instance using a maximum a posteriori decision rule

$$c^* = \arg \max_{c \in \Lambda(C)} p(C = c|\boldsymbol{\Theta} = \boldsymbol{\theta}).$$

Since each predictor variable Θ_i given $C = c$ follows a wrapped Cauchy distribution with location parameter $\mu_{i,c}$ and concentration parameter $\varepsilon_{i,c}$, we can express Eq. (6) as

$$p(c|\boldsymbol{\theta}) \propto= \frac{p(C = c) \prod_{i=1}^{n} \alpha_{i,c}}{\prod_{i=1}^{n} (1 - \beta_{i,c})}, \tag{7}$$

where $\alpha_{i,c} = \frac{1}{2\pi} \frac{(1-\varepsilon_{i,c}^2)}{(1+\varepsilon_{i,c}^2)}$ and $\beta_{i,c} = \frac{2\varepsilon_{i,c} \cos(\theta_i - \mu_{i,c})}{(1+\varepsilon_{i,c}^2)}$.

3.2. Wrapped Cauchy selective naive Bayes

Sometimes there are several predictor variables that do not contribute to classification (i.e., they are redundant), and naive Bayes classifier is affected by such variables [38]. Determining which of them are unnecessary via the use of feature subset selection (FSS) techniques [39] could increase the accuracy of the classification model significantly [40]. Wrapped Cauchy selective naive Bayes (wCsNB) is a classification model with a structure similar to that of wCNB, but not all the variables are necessarily used by the classifier. FSS techniques were previously employed in a circular classification model with von Mises and von Mises–Fisher distributions in [28], where a filter-wrapper algorithm is applied to rank the variables according to the mutual information between them and the class, and therefore, using the ranking provided by the filter step, the variables are selected to induce a new classifier until the best model is achieved.

We also use a filter-wrapper algorithm at this point. The filter step is based on the computation of the mutual information (MI) between each circular variable and the class. There is no equation to compute the MI between circular variables and discrete variables. Therefore, we approach the problem using Monte Carlo methods, as in [28]; we model the conditional density functions of $\Theta_i|C = c$ as wrapped Cauchy distributions. Hence

$$\mathrm{MI}(\Theta_i, C) \approx \frac{1}{M} \sum_{j=1}^{M} \log \frac{\hat{f}_{\Theta_i|c^{*(j)}}\left(\theta_i^{*(j)}|c^{*(j)}\right) \hat{p}\left(C = c^{*(j)}\right)}{\hat{f}_{\Theta_i}\left(\theta_i^{*(j)}\right) \hat{p}\left(C = c^{*(j)}\right)}, \tag{8}$$

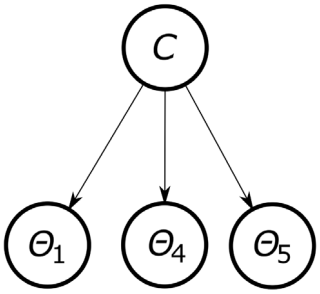

Fig. 2. wCsNB structure with three nodes selected from the original set of five predictive variables, from which $p(c|\boldsymbol{\theta}) \propto p(c) f_{\Theta_1|c}(\theta_1|c) f_{\Theta_4|c}(\theta_4|c) f_{\Theta_5|c}(\theta_5|c)$.

where M is the number of instances $\left(\theta_i^{*(j)}, c^{*(j)}\right)$ sampled from $\hat{f}_{\Theta_i|c}\left(\theta_i|c\right)\hat{p}\left(C=c\right)$, with $\hat{f}_{\Theta_i|c}\left(\theta_i|c\right)$ the fitted wrapped Cauchy density function of the conditional density function of Θ_i given $C=c$, and $\hat{p}\left(C=c\right)$ the relative frequency of instances that belong to class c in the training set.

The predictive variables are then ranked according to their MI values.

The wrapper step consists of creating a new classifier by deciding whether or not to include the ranked predictive variables from the filter step. Each iteration of the wrapper step induces a new classifier adding the next predictive variable from the list. If no accuracy improvement is achieved by including the next predictive variable from the ranked list, then the wrapper step finishes. This model is similar to the WcNB, but including only the selected wrapped Cauchy variables (set S) (Fig. 2) and therefore

$$p(c|\boldsymbol{\theta}) \propto p(c|\boldsymbol{\theta}_S) = p(C=c)\prod_{i\in S} f_{\Theta_i|C=c}(\theta_i|c). \tag{9}$$

As for the wCNB, the wCsNB determines the class value c^* for a new instance using a maximum a posteriori decision rule

$$c^* = \arg\max_{c\in\Lambda(C)} p(C=c|\boldsymbol{\Theta}_S = \boldsymbol{\theta}_S).$$

Likewise for Eq. (7), we can express Eq. (9) as

$$p(c|\boldsymbol{\theta}_S) = \frac{p(C=c)\prod_{i\in S}\alpha_{i,c}}{\prod_{i\in S}(1-\beta_{i,c})},$$

where $\alpha_{i,c} = \frac{1}{2\pi}\frac{(1-\varepsilon_{i,c}^2)}{(1+\varepsilon_{i,c}^2)}$ and $\beta_{i,c} = \frac{2\varepsilon_{i,c}\cos(\theta_{i,c}-\mu_{i,c})}{(1+\varepsilon_{i,c}^2)}$.

3.3. Wrapped Cauchy semi-naive Bayes

Usually, the assumption of conditional independence between predictive variables given the class variable is dismissed. The semi-naive Bayes classification model [41] goes one step further and considers dependencies between predictive variables.

Our proposal for this model, called wrapped Cauchy semi-naive Bayes (wCsmNB) classifier, takes into account the possible dependence between predictive wrapped Cauchy variables by introducing new features obtained as the Cartesian product of two of the original circular predictor variables. Thus we work with a bivariate wrapped Cauchy distribution. These new features remains conditionally independent given the class variable.

Given L_k with $k = 1, \ldots, T$, representing the kth feature (original or new features)

$$p(c|\boldsymbol{\theta}) \propto p(C=c)\prod_{k=1}^{T} f_{\Theta_{L_k}|C=c}(\boldsymbol{\theta}_{L_k}|c).$$

To determine those original variables that are candidates to create new features from the Cartesian product between them, we develop an adaptation of the *forward sequential selection and joining* (FSSJ) algorithm [42] described in Algorithm 1. It is important to note that once the new features are created by joining two original features, these new features cannot be used to create others. However, these new features can be separated in order to use one of the two original features to create another new feature by joining with a different original feature that had not yet been added to the model. This algorithm may result in a selection of variables that provide the best achievable solution, before all of the original variables are included in the model (Fig. 3).

Again, as for the previous models presented in this section, the wCmNB determines the class value c^* for a new instance using a maximum a posteriori decision rule

$$c^* = \arg\max_{c\in\Lambda(C)} p(C=c|\boldsymbol{\Theta} = \theta_{L_k}).$$

Algorithm 1 Adaptation of the FSSJ algorithm of [42]

1: Let T be the variable list, initialized as $T = \emptyset$.
2: Given $\Theta_1, \Theta_2, ..., \Theta_n$ circular wrapped Cauchy predictor variables from a variable list A, move the first variable from A to T.
3: Move the next variable from A to T, considering:

- Joining the variable to another variable currently in T. If the latter variable was previously joined to another variable from T, remove this from T and add it to A, and consider adding it later.
- Add the variable as conditionally independent of the other variables given C to the current classifier.

4: Repeat Step 3 until the best model is achieved

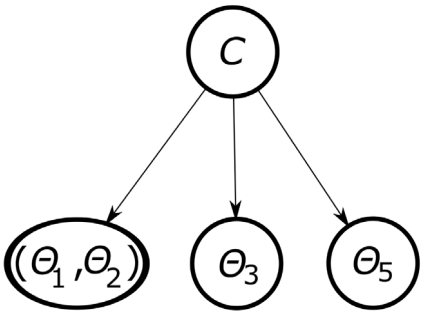

Fig. 3. wCsmNB structure with four nodes from the original set of five predictive variables, from which $p(c|\theta) \propto p(c) f_{\Theta_1,\Theta_2|c}(\theta_1, \theta_2|c) f_{\Theta_3|c}(\theta_3|c) f_{\Theta_5|c}(\theta_5|c)$.

3.4. Wrapped Cauchy tree-augmented naive Bayes

The tree-augmented naive Bayes (TAN) classifier [43] is a well-known Bayesian classifier with a tree-structure network for predictive features. Wrapped Cauchy tree-augmented naive Bayes (wCTAN) classifier is a variation of the TAN classifier with the novelty of the allowance of the use of wrapped Cauchy circular variables for predictive features. wCTAN assumes that the class variable has no parents, and the rest of the variables have at most one other variable as parent apart from C (Fig. 4).

The process for building a wCTAN is summarized in the following three steps:

- Step 1: The structure of the tree for predictive features is learned using Algorithm 2. We use the conditional circular mutual information, denoted as $\mathrm{CMI}(\Theta_i, \Theta_j|C)$, which is defined as

$$\mathrm{CMI}(\Theta_i, \Theta_j|C) = \sum_c \mathrm{CMI}(\Theta_i, \Theta_j|C = c) p(C = c),$$

with

$$\mathrm{CMI}(\Theta_i, \Theta_j|C = c) = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} f(\theta_i, \theta_j|c) \log\left(\frac{f(\theta_i, \theta_j|c)}{f(\theta_i|c) f(\theta_j|c)}\right) d\theta_i d\theta_j.$$

where the marginal density functions given the class, $f(\theta_i|c)$ and $f(\theta_j|c)$, and the joint density function given the class, $f(\theta_i, \theta_j|c)$, have been previously estimated from data. This structure learning algorithm (Algorithm 2) is based on score and search, where structure learning is posed as an optimization problem, using a maximum weighted spanning tree algorithm (where the weights are given by the CMI), a variant of the Chow Liu algorithm [44].

Algorithm 2 Adaptation of the Chow Liu algorithm of [44]

1: Given $\Theta_1, \Theta_2, ..., \Theta_n$ wrapped Cauchy variables, estimate the bivariate joint density function $f(\theta_i, \theta_j|c)$ for all pairs of variables, and the marginals $f(\theta_i|c)$, for each $c \in \Lambda(C)$, $i, j = 1, ..., n$
2: Using these, compute all conditional $\mathrm{CMI}(\Theta_i, \Theta_j|C)$ values, (i.e., the $n(n-1)/2$ edge weights) and order them
3: Assign the largest two edges to the undirected tree to be represented
4: Examine the next-largest edge, and add it to the tree unless it forms a loop, in which case discard it and examine the next largest edge
5: Repeat Step 4 until $n - 1$ edges have been selected (and the spanning undirected tree is finished)

For Step 1 in Algorithm 2, the estimate of the bivariate and marginal densities are performed for each c using the methods explained in Section 2. Like the traditional mutual information measure for linear variables, the $\mathrm{CMI}(\Theta_i, \Theta_j|C)$ denotes the entropy reduction of Θ_i (Θ_j) when the value of Θ_j (Θ_i) is known given C, and represents the weight that links Θ_i and Θ_j.

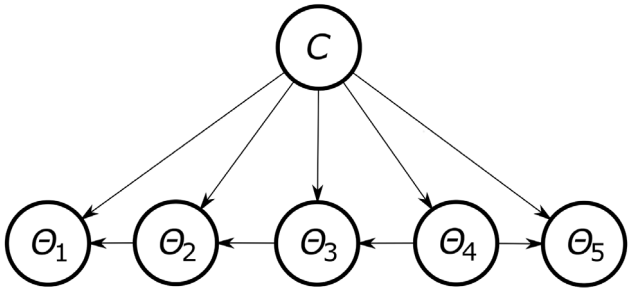

Fig. 4. wCTAN structure with five nodes, from which $p(c|\boldsymbol{\theta}) \propto p(c) f_{\Theta_1|c,\theta_2}(\theta_1|c,\theta_2) f_{\Theta_2|c,\theta_3}(\theta_2|c,\theta_3) f_{\Theta_3|c,\theta_4}(\theta_3|c,\theta_4) f_{\Theta_4|c}(\theta_4|c) f_{\Theta_5|c,\theta_4}(\theta_5|c,\theta_4)$. The associated tree-structured Bayesian network has Θ_4 as its root node.

Once we have learned the undirected structure, a root node must be selected in order to determine the root of the tree by following the structure learned by Algorithm 2. Depending on the selected root node and given the undirected tree structure with n nodes, there are n possible resulting directed trees.

- Step 2: We add a class node C to the network structure. We connect this class node to every other node with an arc from C (Fig. 4).
- Step 3: Finally, we complete the classification model with the estimation of the parameters for each node given its parent node(s).

Therefore the conditional probability of C given the predictors is

$$p(C = c|\boldsymbol{\Theta} = \boldsymbol{\theta}) \propto p(C = c) f_{\Theta_{root}|C=c}(\theta_{root}|c) \prod_{i=1, i\neq root}^{n} f_{\Theta_i|C=c,Pa_{\Theta_i}}(\theta_i|c, pa_{\theta_i}),$$

where Pa_{Θ_i} is the wrapped Cauchy parent of variable Θ_i and Θ_{root} is the root node of the tree.

Similar to the approach used in the rest of the models presented in this paper, the maximum a posteriori decision rule is used to determine the predicted class c^*

$$c^* = \arg \max_{c \in \Lambda(C)} p(C = c|\boldsymbol{\Theta} = \boldsymbol{\theta}).$$

4. Experimental results

In this section, we report experiments carried out to show the behavior of each proposed classification model in Section 3. We include the comparison among the four circular classifiers and also with some of the best-known classification algorithms for linear data, such as decision tree (DTree), random forest (Rfor), multinomial logistic regression (MLG), support vector machine (SVM), simple neural network (Nnet) and the Gaussian tree-augmented naive Bayes classifier (GTAN) for continuous data, with the structure learned with the algorithm in [45] where predictor variables given the class value are assumed to follow Gaussian distributions.

The experiments were run using R software [46]. To generate the artificial datasets to test the models, we used the "Circular" R package for the simulation of circular data, and to implement the structure of the wCTAN classifier, we have adapted the "bnclassify" R package [47]. Simulating data that follows wrapped Cauchy distributions is easy and computationally very fast. Given the parameters, the "Circular" R package simulates wrapped Cauchy data by wrapping the simulation of a Cauchy distribution whose location parameter is the same as the wrapped Cauchy location parameter and the scale parameter is the negative logarithm of the wrapped Cauchy concentration parameter. If the wrapped Cauchy concentration parameter is equal to 1, then the value of the simulation will be the location parameter, whereas if the concentration parameter is equal to 0, the simulation is performed from a Uniform distribution in $[0, 2\pi)$.

In order to test the algorithms, we enforced dependence between nodes giving values of $|\rho|$ in $[0.5, 1)$. The remaining parameters were assigned randomly to each node with $-\pi < \mu < \pi$ and $0 < \varepsilon < 1$. For each classifier, we simulated 10 datasets each with 1000, 200 and 50 instances and 3, 5, 10, 20, 30, 45, 65, and 100 wrapped Cauchy predictor variables and a discrete class variable with 3, 6, 10, 15 and 20 different labels, so we simulated 1200 different datasets for each type of classifier. A 10-fold cross-validation was used to estimate the classification accuracy. Results for Bayesian network classifiers are shown in Table 1, while results for traditional linear classification algorithms are shown in Table 2.

We also applied the non-parametric Friedman test to detect statistically significant differences among our classification models as a whole set [48]. When the null hypothesis was rejected, we proceeded with post-hoc tests. We chose the Nemenyi test [49], as suggested by [50]. The significance level α for all tests was 0.05.

Since multiple classifiers are compared, it is useful to represent the results of the post-hoc tests visually. The graph proposed by Demšar [50] is a simple diagram to easily represent these results. The top line is the axis on which we plot the average Friedman test ranks of the classifiers. The lowest (best) ranks are to the right, and we therefore consider the classifiers to the right as better. For the comparison results of all classifiers against each other, those that are not significantly different (p-value ≥ 0.05 in the Nemenyi post-hoc test) are connected.

Table 1
Mean ± standard deviation accuracy of wCNB, wCsNB, wCsmNB, wCTAN and GTAN for different number of variables, different number of labels in the class variable for simulated datasets with 50, 200 and 1000 instances.

		No. of instances														
		50					200					1000				
		No. of labels					No. of labels					No. of labels				
		3	6	10	15	20	3	6	10	15	20	3	6	10	15	20
wCNB	No. of variables															
	3	0.868 ± 0.090	0.544 ± 0.130	0.382 ± 0.171	0.306 ± 0.141	0.252 ± 0.146	0.882 ± 0.046	0.766 ± 0.071	0.597 ± 0.061	0.530 ± 0.058	0.373 ± 0.076	0.882 ± 0.031	0.775 ± 0.056	0.679 ± 0.071	0.605 ± 0.065	0.548 ± 0.046
	5	0.936 ± 0.067	0.744 ± 0.142	0.486 ± 0.160	0.330 ± 0.155	0.310 ± 0.115	0.957 ± 0.032	0.857 ± 0.050	0.806 ± 0.055	0.672 ± 0.084	0.574 ± 0.074	0.959 ± 0.028	0.886 ± 0.026	0.831 ± 0.032	0.778 ± 0.044	0.737 ± 0.041
	10	0.996 ± 0.008	0.916 ± 0.087	0.604 ± 0.165	0.490 ± 0.149	0.318 ± 0.187	0.997 ± 0.004	0.976 ± 0.020	0.933 ± 0.039	0.909 ± 0.045	0.832 ± 0.054	0.995 ± 0.009	0.983 ± 0.012	0.974 ± 0.016	0.959 ± 0.016	0.948 ± 0.035
	20	0.996 ± 0.008	0.976 ± 0.039	0.750 ± 0.177	0.622 ± 0.177	0.413 ± 0.194	0.999 ± 0.001	0.999 ± 0.002	0.996 ± 0.006	0.983 ± 0.017	0.963 ± 0.030	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.997 ± 0.005	0.994 ± 0.009
	30	0.999 ± 0.001	0.996 ± 0.008	0.832 ± 0.163	0.460 ± 0.209	0.300 ± 0.128	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.995 ± 0.011	0.992 ± 0.016	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.998 ± 0.002
	45	0.999 ± 0.001	0.999 ± 0.001	0.830 ± 0.154	0.430 ± 0.217	0.250 ± 0.111	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
	65	0.999 ± 0.001	0.999 ± 0.001	0.920 ± 0.109	0.500 ± 0.224	0.240 ± 0.132	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
	100	0.999 ± 0.001	0.999 ± 0.001	0.860 ± 0.147	0.490 ± 0.271	0.240 ± 0.072	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
wCsNB	No. of variables															
	3	0.876 ± 0.106	0.628 ± 0.142	0.574 ± 0.141	0.256 ± 0.072	0.238 ± 0.097	0.891 ± 0.044	0.771 ± 0.060	0.607 ± 0.068	0.523 ± 0.070	0.383 ± 0.067	0.895 ± 0.019	0.827 ± 0.022	0.720 ± 0.025	0.627 ± 0.037	0.579 ± 0.035
	5	0.932 ± 0.062	0.740 ± 0.157	0.432 ± 0.173	0.268 ± 0.105	0.232 ± 0.084	0.949 ± 0.031	0.841 ± 0.054	0.784 ± 0.075	0.644 ± 0.101	0.536 ± 0.098	0.967 ± 0.008	0.914 ± 0.011	0.852 ± 0.023	0.791 ± 0.021	0.738 ± 0.030
	10	0.948 ± 0.076	0.776 ± 0.142	0.478 ± 0.160	0.276 ± 0.093	0.230 ± 0.117	0.979 ± 0.020	0.910 ± 0.066	0.822 ± 0.102	0.760 ± 0.147	0.646 ± 0.148	0.976 ± 0.005	0.967 ± 0.005	0.944 ± 0.008	0.918 ± 0.007	0.913 ± 0.011
	20	0.936 ± 0.064	0.824 ± 0.126	0.500 ± 0.170	0.288 ± 0.119	0.252 ± 0.090	0.990 ± 0.014	0.944 ± 0.047	0.872 ± 0.094	0.798 ± 0.120	0.718 ± 0.134	0.994 ± 0.006	0.978 ± 0.018	0.968 ± 0.020	0.948 ± 0.018	0.938 ± 0.025
	30	0.972 ± 0.033	0.830 ± 0.101	0.560 ± 0.215	0.280 ± 0.110	0.238 ± 0.121	0.990 ± 0.016	0.960 ± 0.046	0.855 ± 0.100	0.887 ± 0.039	0.665 ± 0.174	0.996 ± 0.005	0.981 ± 0.016	0.971 ± 0.019	0.950 ± 0.015	0.953 ± 0.025
	45	0.988 ± 0.023	0.832 ± 0.121	0.462 ± 0.155	0.292 ± 0.120	0.238 ± 0.106	0.997 ± 0.005	0.980 ± 0.026	0.895 ± 0.120	0.865 ± 0.078	0.792 ± 0.068	0.997 ± 0.002	0.987 ± 0.012	0.976 ± 0.018	0.960 ± 0.012	0.954 ± 0.014
	65	0.994 ± 0.009	0.856 ± 0.120	0.474 ± 0.150	0.272 ± 0.106	0.262 ± 0.121	0.995 ± 0.013	0.987 ± 0.018	0.960 ± 0.033	0.845 ± 0.172	0.702 ± 0.139	0.998 ± 0.003	0.992 ± 0.009	0.981 ± 0.019	0.965 ± 0.019	0.957 ± 0.02
	100	0.996 ± 0.008	0.876 ± 0.113	0.584 ± 0.165	0.310 ± 0.103	0.238 ± 0.110	0.995 ± 0.111	0.980 ± 0.027	0.940 ± 0.048	0.832 ± 0.143	0.785 ± 0.136	0.999 ± 0.001	0.996 ± 0.004	0.987 ± 0.012	0.974 ± 0.015	0.963 ± 0.038
wCsmNB	No. of variables															
	3	0.880 ± 0.080	0.592 ± 0.136	0.436 ± 0.188	0.266 ± 0.112	0.250 ± 0.088	0.887 ± 0.042	0.778 ± 0.053	0.626 ± 0.082	0.546 ± 0.085	0.403 ± 0.066	0.903 ± 0.010	0.814 ± 0.028	0.715 ± 0.022	0.650 ± 0.039	0.537 ± 0.029
	5	0.952 ± 0.062	0.820 ± 0.133	0.560 ± 0.158	0.290 ± 0.106	0.271 ± 0.122	0.963 ± 0.027	0.871 ± 0.054	0.929 ± 0.041	0.716 ± 0.076	0.623 ± 0.061	0.965 ± 0.028	0.909 ± 0.019	0.844 ± 0.018	0.828 ± 0.039	0.754 ± 0.051
	10	0.996 ± 0.008	0.922 ± 0.082	0.768 ± 0.175	0.352 ± 0.156	0.286 ± 0.110	0.997 ± 0.006	0.982 ± 0.016	0.954 ± 0.032	0.940 ± 0.030	0.873 ± 0.056	0.998 ± 0.002	0.987 ± 0.012	0.982 ± 0.020	0.970 ± 0.019	0.960 ± 0.024
	20	0.999 ± 0.001	0.992 ± 0.014	0.854 ± 0.131	0.486 ± 0.155	0.354 ± 0.113	0.999 ± 0.001	0.998 ± 0.002	0.997 ± 0.003	0.987 ± 0.019	0.979 ± 0.026	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.998 ± 0.002	0.995 ± 0.008
	30	0.999 ± 0.001	0.996 ± 0.008	0.864 ± 0.129	0.556 ± 0.207	0.360 ± 0.102	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
	45	0.999 ± 0.001	0.998 ± 0.004	0.914 ± 0.134	0.632 ± 0.168	0.366 ± 0.127	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
	65	0.999 ± 0.001	0.999 ± 0.001	0.980 ± 0.167	0.650 ± 0.137	0.360 ± 0.142	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
	100	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.640 ± 0.018	0.420 ± 0.083	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
wCTAN	No. of variables															
	3	0.868 ± 0.093	0.544 ± 0.130	0.382 ± 0.171	0.230 ± 0.096	0.200 ± 0.100	0.879 ± 0.049	0.754 ± 0.054	0.590 ± 0.060	0.513 ± 0.083	0.378 ± 0.087	0.879 ± 0.025	0.793 ± 0.031	0.716 ± 0.030	0.598 ± 0.040	0.554 ± 0.039
	5	0.938 ± 0.077	0.774 ± 0.151	0.484 ± 0.142	0.240 ± 0.097	0.160 ± 0.069	0.955 ± 0.034	0.851 ± 0.063	0.774 ± 0.072	0.662 ± 0.078	0.539 ± 0.087	0.980 ± 0.010	0.910 ± 0.014	0.835 ± 0.021	0.776 ± 0.018	0.744 ± 0.024
	10	0.992 ± 0.017	0.890 ± 0.099	0.694 ± 0.164	0.282 ± 0.170	0.206 ± 0.119	0.995 ± 0.007	0.967 ± 0.023	0.909 ± 0.043	0.872 ± 0.051	0.809 ± 0.061	0.995 ± 0.006	0.981 ± 0.014	0.969 ± 0.015	0.950 ± 0.021	0.935 ± 0.022
	20	0.999 ± 0.001	0.978 ± 0.031	0.784 ± 0.166	0.396 ± 0.169	0.292 ± 0.135	0.999 ± 0.001	0.997 ± 0.002	0.994 ± 0.010	0.974 ± 0.024	0.951 ± 0.034	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.997 ± 0.004	0.995 ± 0.007
	30	0.999 ± 0.001	0.999 ± 0.001	0.896 ± 0.123	0.390 ± 0.130	0.282 ± 0.127	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.997 ± 0.005	0.995 ± 0.011	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
	45	0.999 ± 0.001	0.999 ± 0.001	0.848 ± 0.162	0.450 ± 0.199	0.252 ± 0.138	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
	65	0.999 ± 0.001	0.999 ± 0.001	0.890 ± 0.134	0.534 ± 0.178	0.246 ± 0.113	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
	100	0.999 ± 0.001	0.999 ± 0.001	0.886 ± 0.149	0.620 ± 0.156	0.244 ± 0.140	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
GTAN	No. of variables															
	3	0.484 ± 0.162	0.302 ± 0.144	0.214 ± 0.111	0.134 ± 0.082	0.058 ± 0.059	0.541 ± 0.106	0.321 ± 0.079	0.229 ± 0.070	0.180 ± 0.062	0.146 ± 0.052	0.698 ± 0.042	0.387 ± 0.051	0.228 ± 0.041	0.178 ± 0.066	0.119 ± 0.105
	5	0.588 ± 0.142	0.372 ± 0.131	0.252 ± 0.115	0.154 ± 0.098	0.034 ± 0.028	0.641 ± 0.099	0.643 ± 0.087	0.425 ± 0.079	0.243 ± 0.080	0.225 ± 0.057	0.714 ± 0.021	0.576 ± 0.038	0.467 ± 0.035	0.285 ± 0.041	0.220 ± 0.086
	10	0.656 ± 0.162	0.460 ± 0.161	0.400 ± 0.148	0.212 ± 0.101	0.008 ± 0.014	0.622 ± 0.091	0.583 ± 0.090	0.521 ± 0.072	0.382 ± 0.070	0.215 ± 0.060	0.803 ± 0.056	0.518 ± 0.048	0.421 ± 0.030	0.325 ± 0.087	0.258 ± 0.090
	20	0.720 ± 0.152	0.676 ± 0.139	0.488 ± 0.163	0.306 ± 0.162	0.036 ± 0.060	0.777 ± 0.082	0.637 ± 0.054	0.598 ± 0.093	0.437 ± 0.066	0.269 ± 0.089	0.777 ± 0.071	0.637 ± 0.055	0.598 ± 0.059	0.437 ± 0.091	0.372 ± 0.082
	30	0.774 ± 0.131	0.710 ± 0.148	0.546 ± 0.167	0.340 ± 0.143	0.102 ± 0.106	0.977 ± 0.018	0.805 ± 0.074	0.657 ± 0.059	0.730 ± 0.072	0.677 ± 0.047	0.867 ± 0.026	0.723 ± 0.043	0.679 ± 0.053	0.576 ± 0.059	0.440 ± 0.096
	45	0.802 ± 0.148	0.778 ± 0.128	0.612 ± 0.158	0.414 ± 0.127	0.144 ± 0.120	0.902 ± 0.057	0.915 ± 0.044	0.855 ± 0.058	0.750 ± 0.058	0.640 ± 0.082	0.940 ± 0.025	0.781 ± 0.021	0.819 ± 0.052	0.688 ± 0.066	0.609 ± 0.096
	65	0.920 ± 0.076	0.828 ± 0.125	0.706 ± 0.152	0.460 ± 0.174	0.182 ± 0.106	0.951 ± 0.006	0.820 ± 0.020	0.819 ± 0.045	0.867 ± 0.049	0.827 ± 0.047	0.927 ± 0.071	0.874 ± 0.053	0.819 ± 0.046	0.758 ± 0.084	0.681 ± 0.098
	100	0.816 ± 0.144	0.890 ± 0.083	0.750 ± 0.140	0.504 ± 0.139	0.208 ± 0.141	0.965 ± 0.015	0.935 ± 0.016	0.972 ± 0.018	0.847 ± 0.028	0.710 ± 0.046	0.964 ± 0.019	0.942 ± 0.024	0.834 ± 0.051	0.815 ± 0.074	0.772 ± 0.089

Table 2
Mean ± standard deviation accuracy of DTre, Rfor, MLG, SVM and Nnet for different number of variables, different number of labels in the class variable for simulated datasets with 50, 200 and 1000 instances.

		No. of instances														
		50					200					1000				
		No. of labels					No. of labels					No. of labels				
		3	6	10	15	20	3	6	10	15	20	3	6	10	15	20
DTre	3	0.710 ± 0.321	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.817 ± 0.109	0.085 ± 0.039	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.792 ± 0.093	0.116 ± 0.009	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001
	5	0.790 ± 0.128	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.817 ± 0.141	0.082 ± 0.028	0.001 ± 0.001	0.032 ± 0.068	0.001 ± 0.001	0.845 ± 0.060	0.115 ± 0.021	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001
	10	0.890 ± 0.152	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.817 ± 0.124	0.087 ± 0.039	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.919 ± 0.041	0.113 ± 0.017	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001
	20	0.780 ± 0.091	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.885 ± 0.105	0.095 ± 0.028	0.001 ± 0.001	0.007 ± 0.023	0.001 ± 0.001	0.930 ± 0.052	0.128 ± 0.010	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001
	30	0.890 ± 0.144	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.915 ± 0.072	0.102 ± 0.024	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.919 ± 0.029	0.126 ± 0.014	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001
	45	0.890 ± 0.119	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.922 ± 0.080	0.090 ± 0.024	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.955 ± 0.033	0.134 ± 0.014	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001
	65	0.790 ± 0.191	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.910 ± 0.059	0.117 ± 0.012	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.948 ± 0.023	0.135 ± 0.008	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001
	100	0.880 ± 0.147	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001	0.892 ± 0.048	0.105 ± 0.025	0.001 ± 0.001	0.275 ± 0.060	0.001 ± 0.001	0.949 ± 0.018	0.136 ± 0.011	0.001 ± 0.001	0.001 ± 0.001	0.001 ± 0.001
Rfor	3	0.833 ± 0.090	0.706 ± 0.114	0.460 ± 0.152	0.333 ± 0.133	0.306 ± 0.095	0.821 ± 0.083	0.706 ± 0.114	0.628 ± 0.066	0.458 ± 0.068	0.391 ± 0.069	0.873 ± 0.061	0.793 ± 0.059	0.672 ± 0.083	0.586 ± 0.055	0.468 ± 0.054
	5	0.900 ± 0.118	0.740 ± 0.110	0.466 ± 0.125	0.333 ± 0.104	0.313 ± 0.094	0.935 ± 0.044	0.848 ± 0.049	0.700 ± 0.086	0.618 ± 0.091	0.521 ± 0.074	0.950 ± 0.028	0.884 ± 0.048	0.806 ± 0.035	0.719 ± 0.053	0.655 ± 0.078
	10	0.946 ± 0.068	0.886 ± 0.044	0.686 ± 0.109	0.373 ± 0.071	0.286 ± 0.117	0.965 ± 0.036	0.933 ± 0.041	0.878 ± 0.032	0.781 ± 0.084	0.741 ± 0.068	0.987 ± 0.006	0.963 ± 0.017	0.927 ± 0.020	0.883 ± 0.030	0.847 ± 0.021
	20	0.973 ± 0.064	0.920 ± 0.087	0.820 ± 0.121	0.540 ± 0.138	0.326 ± 0.119	0.996 ± 0.007	0.978 ± 0.015	0.971 ± 0.017	0.913 ± 0.039	0.856 ± 0.047	0.997 ± 0.003	0.992 ± 0.007	0.987 ± 0.011	0.979 ± 0.006	0.966 ± 0.015
	30	0.999 ± 0.001	0.960 ± 0.046	0.880 ± 0.075	0.493 ± 0.114	0.433 ± 0.175	0.999 ± 0.001	0.998 ± 0.005	0.991 ± 0.008	0.953 ± 0.034	0.940 ± 0.023	0.999 ± 0.001	0.999 ± 0.001	0.998 ± 0.002	0.996 ± 0.003	0.990 ± 0.007
	45	0.999 ± 0.001	0.980 ± 0.044	0.860 ± 0.145	0.626 ± 0.144	0.413 ± 0.159	0.998 ± 0.005	0.999 ± 0.001	0.996 ± 0.010	0.988 ± 0.015	0.966 ± 0.030	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.002	0.999 ± 0.001	0.997 ± 0.002
	65	0.999 ± 0.001	0.993 ± 0.021	0.893 ± 0.109	0.613 ± 0.182	0.373 ± 0.089	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.996 ± 0.007	0.981 ± 0.018	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
	100	0.999 ± 0.001	0.999 ± 0.001	0.946 ± 0.068	0.706 ± 0.144	0.526 ± 0.119	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.998 ± 0.005	0.985 ± 0.027	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
MLG	3	0.793 ± 0.073	0.446 ± 0.129	0.240 ± 0.071	0.186 ± 0.112	0.253 ± 0.098	0.703 ± 0.112	0.526 ± 0.108	0.450 ± 0.105	0.285 ± 0.050	0.205 ± 0.073	0.745 ± 0.083	0.581 ± 0.068	0.442 ± 0.094	0.372 ± 0.053	0.268 ± 0.054
	5	0.780 ± 0.171	0.513 ± 0.166	0.340 ± 0.096	0.260 ± 0.119	0.233 ± 0.084	0.811 ± 0.120	0.641 ± 0.086	0.475 ± 0.129	0.416 ± 0.072	0.330 ± 0.086	0.840 ± 0.049	0.665 ± 0.083	0.560 ± 0.039	0.445 ± 0.047	0.383 ± 0.045
	10	0.733 ± 0.160	0.593 ± 0.119	0.326 ± 0.096	0.266 ± 0.104	0.220 ± 0.099	0.853 ± 0.088	0.716 ± 0.070	0.531 ± 0.054	0.441 ± 0.079	0.356 ± 0.070	0.896 ± 0.054	0.777 ± 0.064	0.691 ± 0.041	0.622 ± 0.041	0.544 ± 0.029
	20	0.860 ± 0.101	0.626 ± 0.178	0.546 ± 0.153	0.340 ± 0.167	0.260 ± 0.091	0.915 ± 0.048	0.753 ± 0.059	0.625 ± 0.088	0.488 ± 0.086	0.415 ± 0.057	0.961 ± 0.019	0.889 ± 0.030	0.800 ± 0.025	0.736 ± 0.028	0.648 ± 0.059
	30	0.900 ± 0.134	0.693 ± 0.126	0.5336 ± 0.251	0.503 ± 0.122	0.340 ± 0.142	0.915 ± 0.030	0.825 ± 0.057	0.630 ± 0.116	0.498 ± 0.056	0.460 ± 0.069	0.969 ± 0.019	0.902 ± 0.017	0.834 ± 0.035	0.770 ± 0.039	0.693 ± 0.040
	45	0.913 ± 0.121	0.780 ± 0.133	0.680 ± 0.203	0.560 ± 0.206	0.353 ± 0.140	0.975 ± 0.030	0.833 ± 0.061	0.718 ± 0.106	0.710 ± 0.106	0.533 ± 0.082	0.976 ± 0.022	0.927 ± 0.034	0.854 ± 0.037	0.854 ± 0.062	0.791 ± 0.083
	65	0.960 ± 0.064	0.880 ± 0.081	0.666 ± 0.206	0.480 ± 0.150	0.400 ± 0.166	0.933 ± 0.084	0.863 ± 0.149	0.883 ± 0.118	0.725 ± 0.217	0.740 ± 0.184	0.992 ± 0.01	0.946 ± 0.050	0.921 ± 0.049	0.869 ± 0.053	0.735 ± 0.084
	100	0.973 ± 0.056	0.913 ± 0.177	0.693 ± 0.208	0.700 ± 0.178	0.520 ± 0.171	0.980 ± 0.035	0.848 ± 0.168	0.895 ± 0.185	0.836 ± 0.175	0.811 ± 0.144	0.994 ± 0.016	0.965 ± 0.056	0.914 ± 0.106	0.899 ± 0.097	0.933 ± 0.090
SVM	3	0.793 ± 0.101	0.433 ± 0.161	0.240 ± 0.151	0.220 ± 0.109	0.200 ± 0.094	0.823 ± 0.078	0.658 ± 0.100	0.530 ± 0.099	0.355 ± 0.055	0.268 ± 0.071	0.858 ± 0.048	0.741 ± 0.049	0.605 ± 0.050	0.515 ± 0.049	0.418 ± 0.045
	5	0.840 ± 0.144	0.473 ± 0.167	0.300 ± 0.122	0.166 ± 0.105	0.213 ± 0.103	0.853 ± 0.084	0.725 ± 0.071	0.561 ± 0.058	0.473 ± 0.065	0.348 ± 0.096	0.921 ± 0.031	0.825 ± 0.042	0.719 ± 0.043	0.600 ± 0.051	0.523 ± 0.050
	10	0.860 ± 0.101	0.700 ± 0.122	0.346 ± 0.182	0.166 ± 0.090	0.286 ± 0.044	0.901 ± 0.073	0.826 ± 0.091	0.686 ± 0.069	0.551 ± 0.122	0.456 ± 0.089	0.960 ± 0.025	0.882 ± 0.033	0.812 ± 0.038	0.756 ± 0.037	0.678 ± 0.035
	20	0.966 ± 0.047	0.773 ± 0.141	0.413 ± 0.150	0.240 ± 0.078	0.240 ± 0.095	0.995 ± 0.011	0.921 ± 0.029	0.821 ± 0.053	0.721 ± 0.085	0.583 ± 0.060	0.989 ± 0.008	0.955 ± 0.016	0.905 ± 0.022	0.863 ± 0.032	0.815 ± 0.041
	30	0.973 ± 0.046	0.873 ± 0.037	0.453 ± 0.168	0.293 ± 0.056	0.240 ± 0.078	0.988 ± 0.013	0.965 ± 0.025	0.941 ± 0.051	0.843 ± 0.040	0.736 ± 0.057	0.997 ± 0.003	0.924 ± 0.013	0.953 ± 0.016	0.942 ± 0.018	0.920 ± 0.019
	45	0.993 ± 0.021	0.900 ± 0.084	0.520 ± 0.159	0.220 ± 0.099	0.173 ± 0.095	0.999 ± 0.001	0.999 ± 0.014	0.971 ± 0.023	0.931 ± 0.038	0.808 ± 0.078	0.999 ± 0.001	0.995 ± 0.003	0.990 ± 0.006	0.988 ± 0.005	0.981 ± 0.009
	65	0.999 ± 0.001	0.766 ± 0.175	0.626 ± 0.172	0.253 ± 0.081	0.226 ± 0.154	0.999 ± 0.001	0.991 ± 0.011	0.995 ± 0.011	0.971 ± 0.024	0.918 ± 0.041	0.999 ± 0.001	0.999 ± 0.001	0.996 ± 0.003	0.993 ± 0.004	0.995 ± 0.003
	100	0.999 ± 0.001	0.906 ± 0.166	0.606 ± 0.209	0.226 ± 0.105	0.280 ± 0.103	0.999 ± 0.001	0.999 ± 0.001	0.998 ± 0.005	0.985 ± 0.027	0.951 ± 0.039	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.999 ± 0.001
Nnet	3	0.740 ± 0.131	0.480 ± 0.195	0.300 ± 0.137	0.200 ± 0.094	0.153 ± 0.094	0.753 ± 0.097	0.611 ± 0.100	0.465 ± 0.084	0.293 ± 0.073	0.215 ± 0.057	0.856 ± 0.049	0.766 ± 0.062	0.631 ± 0.069	0.558 ± 0.057	0.471 ± 0.039
	5	0.740 ± 0.127	0.453 ± 0.120	0.246 ± 0.137	0.253 ± 0.093	0.266 ± 0.108	0.816 ± 0.110	0.660 ± 0.084	0.406 ± 0.084	0.321 ± 0.108	0.243 ± 0.088	0.895 ± 0.032	0.804 ± 0.053	0.672 ± 0.041	0.567 ± 0.060	0.513 ± 0.067
	10	0.740 ± 0.115	0.506 ± 0.163	0.380 ± 0.133	0.220 ± 0.083	0.233 ± 0.095	0.876 ± 0.073	0.701 ± 0.072	0.443 ± 0.091	0.300 ± 0.054	0.246 ± 0.065	0.917 ± 0.022	0.773 ± 0.046	0.648 ± 0.059	0.525 ± 0.037	0.444 ± 0.056
	20	0.880 ± 0.103	0.666 ± 0.150	0.273 ± 0.091	0.200 ± 0.111	0.220 ± 0.070	0.940 ± 0.038	0.745 ± 0.118	0.535 ± 0.102	0.268 ± 0.091	0.241 ± 0.106	0.956 ± 0.017	0.817 ± 0.087	0.679 ± 0.054	0.474 ± 0.101	0.427 ± 0.068
	30	0.873 ± 0.017	0.660 ± 0.019	0.353 ± 0.029	0.273 ± 0.096	0.293 ± 0.075	0.961 ± 0.035	0.846 ± 0.081	0.575 ± 0.139	0.301 ± 0.131	0.235 ± 0.081	0.979 ± 0.026	0.900 ± 0.043	0.758 ± 0.053	0.542 ± 0.059	0.437 ± 0.096
	45	0.920 ± 0.171	0.713 ± 0.196	0.400 ± 0.121	0.280 ± 0.112	0.220 ± 0.070	0.980 ± 0.036	0.860 ± 0.057	0.635 ± 0.106	0.440 ± 0.115	0.203 ± 0.127	0.988 ± 0.011	0.928 ± 0.024	0.758 ± 0.152	0.630 ± 0.089	0.443 ± 0.092
	65	0.953 ± 0.054	0.666 ± 0.160	0.320 ± 0.162	0.273 ± 0.110	0.320 ± 0.136	0.990 ± 0.021	0.888 ± 0.057	0.641 ± 0.235	0.386 ± 0.141	0.291 ± 0.119	0.963 ± 0.100	0.961 ± 0.020	0.700 ± 0.268	0.550 ± 0.158	0.408 ± 0.099
	100	0.966 ± 0.047	0.633 ± 0.169	0.460 ± 0.252	0.240 ± 0.095	0.240 ± 0.105	0.986 ± 0.024	0.931 ± 0.044	0.750 ± 0.162	0.448 ± 0.182	0.358 ± 0.134	0.994 ± 0.007	0.974 ± 0.019	0.809 ± 0.259	0.490 ± 0.218	0.521 ± 0.173

Table 3
Mean ± standard deviation accuracy of wCNB, wCsNB, wCsmNB, wCTAN, GTAN, DTree, Rfor, MLG, SVM and Nnet classifiers for different number of variables. Results are averaged from the classification performance from Tables 1 and 2 with 3, 6, 10, 15 and 20 different labels with 1000 instances. Bolded results are best performing classifiers.

		Classifiers				
		wCNB	wCsNB	wCsmNB	wCTAN	GTAN
No. of variables	3	0.735 ± 0.108	**0.755 ± 0.097**	0.754 ± 0.109	0.743 ± 0.108	0.352 ± 0.171
	5	0.866 ± 0.069	0.877 ± 0.069	**0.879 ± 0.066**	0.876 ± 0.074	0.409 ± 0.174
	10	0.976 ± 0.015	0.948 ± 0.021	**0.983 ± 0.011**	0.974 ± 0.018	0.491 ± 0.159
	20	**0.998 ± 0.001**	0.970 ± 0.017	**0.998 ± 0.001**	**0.998 ± 0.001**	0.610 ± 0.122
	30	0.998 ± 0.001	0.976 ± 0.015	**0.999 ± 0.001**	**0.999 ± 0.001**	0.674 ± 0.116
	45	**0.999 ± 0.001**	0.980 ± 0.013	**0.999 ± 0.001**	**0.999 ± 0.001**	0.790 ± 0.091
	65	**0.999 ± 0.001**	0.984 ± 0.013	**0.999 ± 0.001**	**0.999 ± 0.001**	0.824 ± 0.076
	100	**0.999 ± 0.001**	0.989 ± 0.012	**0.999 ± 0.001**	**0.999 ± 0.001**	0.873 ± 0.062
		DTree	Rfor	MLG	SVM	Nnet
No. of variables	3	0.182 ± 0.021	0.678 ± 0.062	0.482 ± 0.070	0.627 ± 0.048	0.656 ± 0.055
	5	0.193 ± 0.017	0.803 ± 0.048	0.579 ± 0.053	0.718 ± 0.043	0.690 ± 0.051
	10	0.207 ± 0.012	0.921 ± 0.019	0.706 ± 0.046	0.818 ± 0.034	0.661 ± 0.044
	20	0.212 ± 0.013	0.984 ± 0.008	0.807 ± 0.032	0.905 ± 0.024	0.671 ± 0.065
	30	0.210 ± 0.009	0.996 ± 0.003	0.834 ± 0.030	0.947 ± 0.014	0.723 ± 0.055
	45	0.218 ± 0.010	**0.999 ± 0.001**	0.880 ± 0.048	0.991 ± 0.005	0.749 ± 0.074
	65	0.217 ± 0.007	**0.999 ± 0.001**	0.893 ± 0.049	0.996 ± 0.002	0.716 ± 0.129
	100	0.218 ± 0.006	**0.999 ± 0.001**	0.941 ± 0.073	**0.999 ± 0.001**	0.758 ± 0.135

4.1. Comparison of classification models

In this section, we compare the performance of the wCNB, wCsNB, wCsmNB and wCTAN models, as well as the DTree, Rfor, MLG, SVM, Nnet and the GTAN algorithms, which ignores the circular nature of the data. We analyze the results of the simulation with 1000 instances. Additionally, we analyze the Bayesian network classifiers performance for 50 and 200 instances.

Table 3 shows the mean± standard deviation accuracy for each classifier for different number of variables. Each mean± standard deviation accuracy values was obtained from the results of 50 independent 10-fold cross-validation procedures varying the number of labels of the class variable (3, 6, 10, 15 and 20 different labels) with 1000 instances.

The statistical analysis after Friedman test rejection (p-value = 0.000000005) reveals (Fig. 5A) that, varying the number of variables, the best classifiers are wCsmNB, wCTAN, wCNB, Rfor and wCsNB with no statistically significant differences among them , whereas the DTree, GTAN, Nnet and MLG classifiers are the worst, presenting significant differences with respect to the rest of the classifiers but for the SVM (which does not present statistical differences with the Nnet and MLG)and demonstrating that treating circular data as linear-continuous is not effective. The SVM also presents statistical differences when compared with the wCsmNB, wCTAN and wCNB classifiers, which outperforms the SVM results. Nevertheless, there were no significant differences between the SVM and the remaining circular classifier (i.e., wCsNB) and the Rfor classifier.

Performing the same statistical analysis among Bayesian network classifiers with 50 and 200 instances yields similar results. The Friedman test null hypothesis that there is no significant difference is rejected for both (p-value = 0.00021 and p-value = 0.00004, respectively). The post-hoc analysis displays quite similar results to the 1000 instances one; in both cases, there are no statistically significant differences among the wCsmNB, wCTAN and wCNB classifiers, which are the best. Nevertheless, for 50 and 200 instances, there are no significant differences among GTAN and wCsNB classifiers. Furthermore, there are significant differences between the wCsNB and the wCsmNB classifier for the analysis with 50 instances, whereas for 200 instances, statistical differences were seen between the wCsNB classifier and both the wCsmNB and the wCTAN.

We also calculated the mean accuracy for each classifier for different number of labels in the class variable (see Table 4). Each mean accuracy value was obtained from the results of 60 independent 10-fold cross-validation procedures varying the number of variables to be used: 3, 5, 10, 20, 30 and 45 with 1000 instances. We do not include the results of the experiments with more than 45 variables due to the high mean accuracy values obtained in most of the classifiers from Tables 1 and 2, which would bias the results.

Since the Friedman test null hypothesis that there is no significant difference was rejected (p-value = 0.0000011), we performed the corresponding Nemenyi post-hoc analysis. Statistical test results (Fig. 5B) reveal that based on changing the number of labels, the best classifiers are the circular Bayesian network classifiers (i.e., wCsmNB, wCTAN, wCNB and wCsNB), with no statistically significant differences between them. Again, DTree GTAN, Nnet and MLG are the worst, with no significant differences among them. These classifiers shows significant differences with wCsmNB, wCTAN, wCNB, Rfor and wCsNB, whereas SVM only presents significant differences with the wCsmNB, wCTAN, wCNB, DTree and GTAN classifiers.

The analysis for Bayesian network classifiers with 50 and 200 instances again yielded quite similar results to those obtained for 1000 instances. After Friedman test rejections (p-value = 0.0325 for 50 instances, and p-value = 0.00066 for 200 instances), post-hoc tests for 200 instances reveal the same statistically significant differences as for 1000 instances, where there is no statistical differences among the wCsmNB, wCTAN and wCNB classifiers, which are the best. For 50 instances, wCsmNB, wCTAN and wCNB are also the best classifiers together with the wCsNB, with no statistically significant differences among them. Likewise, for 1000 instances, GTAN is the worst for the analysis with 50 instances as well as the 200 instances, with no significant differences with the wCsNB classifier.

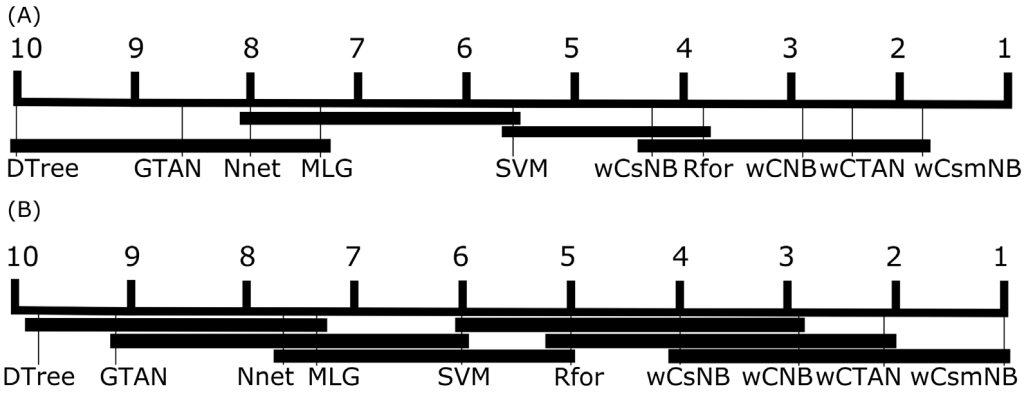

Fig. 5. Demšar diagrams presenting the statistical comparison among wCNB, wCsNB, wCsmNB, wCTAN, GTAN, DTree, Rfor, MLG, SVM and Nnet classification models for synthetic datasets with 1000 instances. Those classifiers that are not connected show differences that are statistically significant (p-value < 0.05). The lowest rank classifiers are to the right side of the graph (i.e., they can be considered the best). (A) Comparison varying the number of labels. (B) Comparison varying the number of variables.

Table 4
Mean ± standard deviation accuracy of wCNB, wCsNB, wCsmNB, wCTAN, GTAN, DTree, Rfor, MLG, SVM and Nnet classifiers for different number of labels. Results are averaged from the classification performance with 3, 5, 10, 20, 30 and 45 different variables with 1000 instances. Bolded results are best performing classifiers.

		Classifiers				
		wCNB	wCsNB	wCsmNB	wCTAN	GTAN
No. of labels	3	0.973 ± 0.042	0.970 ± 0.034	**0.979 ± 0.035**	0.976 ± 0.043	0.709 ± 0.061
	6	0.940 ± 0.083	0.939 ± 0.054	**0.952 ± 0.069**	0.946 ± 0.074	0.526 ± 0.097
	10	0.910 ± 0.118	0.901 ± 0.089	**0.921 ± 0.106**	0.918 ± 0.106	0.384 ± 0.124
	15	0.884 ± 0.146	0.862 ± 0.117	**0.906 ± 0.128**	0.884 ± 0.148	0.311 ± 0.096
	20	0.858 ± 0.167	0.839 ± 0.134	**0.872 ± 0.171**	0.860 ± 0.164	0.253 ± 0.098
		DTree	Rfor	MLG	SVM	Nnet
No. of labels	3	0.893 ± 0.051	0.968 ± 0.017	0.898 ± 0.041	0.954 ± 0.019	0.932 ± 0.026
	6	0.122 ± 0.014	0.938 ± 0.022	0.790 ± 0.049	0.887 ± 0.026	0.831 ± 0.053
	10	0.001 ± 0.001	0.898 ± 0.026	0.697 ± 0.045	0.831 ± 0.029	0.691 ± 0.071
	15	0.001 ± 0.001	0.860 ± 0.025	0.633 ± 0.045	0.777 ± 0.032	0.549 ± 0.067
	20	0.001 ± 0.001	0.821 ± 0.030	0.555 ± 0.052	0.723 ± 0.033	0.456 ± 0.070

5. Real data example

We applied our classifiers to a dataset of 3027 combinations of dendritic bifurcation angles coming from the basal arbors of 288 3D pyramidal neurons in layers *II, III, IV, Va, Vb* and *VI* (48 neurons per layer) of the 14-day-old (P14) rat hind limb somatosensory (S1HL) neocortex, recently published in [14] (Fig. 6).

We used the Bayesian network classification models presented in Section 3 and wrapped Cauchy distributions to model the bifurcation angles produced by the splitting of the dendritic segments of basal dendritic trees. The dendritic bifurcation angles are an important part of the geometry of pyramidal cell arbors. Since it is thought that these angles determine the space to be filled by the dendritic wiring, understanding and modeling them are crucial for advances in neuroscience to replicate brain functioning and structure in order to make further on how the brain processes information. This is important not only to understand it biologically (i.e., thoughts, emotions, feelings) but also technological, making essential contributions to new computing. Moreover, brain knowledge is basic for treating brain diseases such as Parkinson or Alzheimer.

Predicting which layer a neuron belongs to is an important task to help understand any neural circuit, and it represents part of the picture regarding the identification and characterization of all its components. To the best of our knowledge, there is no any supervised classification model that predicts the layer using circular predictive variables. Thus, we developed a classification model to predict which layer a given neuron belongs to, i.e., $\Lambda(C) = (II, III, IV, Va, Vb, VI)$.

Following the notation used in [14], Θ_1 will correspond to the first bifurcation angle (Order 1) generated for the first split of the dendritic segments starting from the soma. The second angle generated by the next consecutive splits will be represented as variable Θ_2 (Order 2), etc. (Fig. 7). Angles of orders higher than six which were relatively scarce were not included in the model. For each set of angles of the same order, a wrapped Cauchy distribution was fitted (Table 5). We performed a goodness-of-fit test by transformation on the circle of the variables into circular uniform variables via $2\pi F(\Theta_1), \ldots, 2\pi F(\Theta_6)$, where F is the cumulative distribution function, and applied Kuiper's test [51] for circular uniformity with a significance level of $\alpha = 0.05$.

Note that in Table 5 the circular mean tends to decrease as the order increases. A neuroscientific explanation for this behavior relates to the fact that it is the first bifurcation orders that determine the volume of space to be filled by the dendritic trees [14]. This

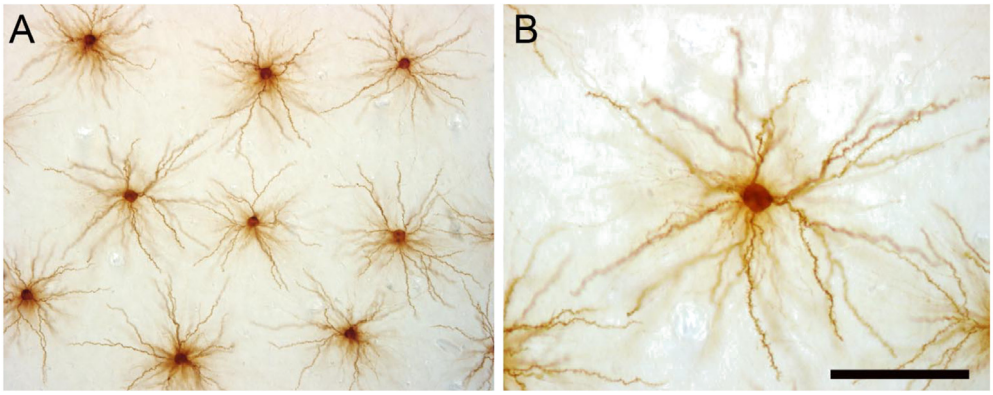

Fig. 6. (A) Low-power photomicrograph showing injected neurons in layers III from the S1HL region of P14 rats, as seen in the plane of section parallel to the cortical surface. (B) Higher magnification photomicrograph showing an example of a pyramidal cell basal dendritic arbor. Scale bar (in B) = 200 μm in A; 90 μm in B.
Source: Adapted from [14].

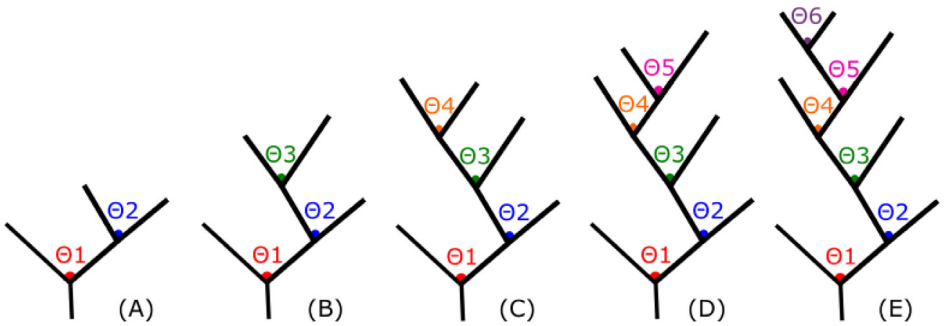

Fig. 7. Angles of different branch orders (from 1 to 6) measured between sibling segments in a dendritic arbor. The dendritic arbor has a maximum branching order of (A) 2 (B) 3 (C) 4 (D) 5 (E) 6.

Table 5
Characteristics of the six different branching orders shown in Fig. 7.

Bifurcation order	Variable	No. of angles	$\hat{\mu}$ (in radians)	$\hat{\varepsilon}$
1	Θ_1	1607	1.02	0.90
2	Θ_2	2072	0.90	0.91
3	Θ_3	1773	0.82	0.92
4	Θ_4	998	0.78	0.92
5	Θ_5	382	0.77	0.92
6	Θ_6	106	0.81	0.92

regulates the dendritic branching development rules that seem to determine the synaptic connectivity of pyramidal neurons. We also observe that the concentration values are high (around 0.91) and quite similar in every bifurcation order. This fact demonstrates that the dendritic structure (in terms of bifurcation angles) is determined by the location parameter.

Since not all dendritic arbors present angles of all orders, one classifier for the whole dataset is not suitable. Therefore, for each classification model proposed in this paper, we created a battery of five classifiers depending on the maximum bifurcation order of the arbor, when this is higher than 1 (Fig. 8). Before predicting class c^*, we have to check the maximum bifurcation order of the instance to be classified. For the wCTAN and GTAN structures (which require a root node in addition to the class node) we select as root node Θ_2 for every classifier of the battery. We performed 10 fold cross-validation procedures in order to obtain the mean classification accuracy values for each classifier and maximum bifurcation order (Table 6).

We observe in Table 6 that the wCsNB classifier leads to the best results for arbors with a maximum branching order of Θ_2 and Θ_3. Furthermore, for arbors with a maximum branching order of Θ_4, Θ_5 or Θ_6, the wCsmNB seems to perform best in terms of classification accuracy. The wCTAN and wCNB classifiers also report acceptable values in comparison with the highest ones for each maximum bifurcation arbor, although the wCsNB or wCsmNB classification models always perform better for this neuronal dataset. Comparing the accuracy results with the random label assignation (i.e., $1/6 = 0.16$), we observe that all of these results are over 0.16. In addition, for every case, the GTAN classifier exhibits the lowest accuracy values, below 0.16 except for Θ_4. This classifier was especially inaccurate for arbors that had maximum branching order of 6; the mean accuracy value was 0.047 for such cases.

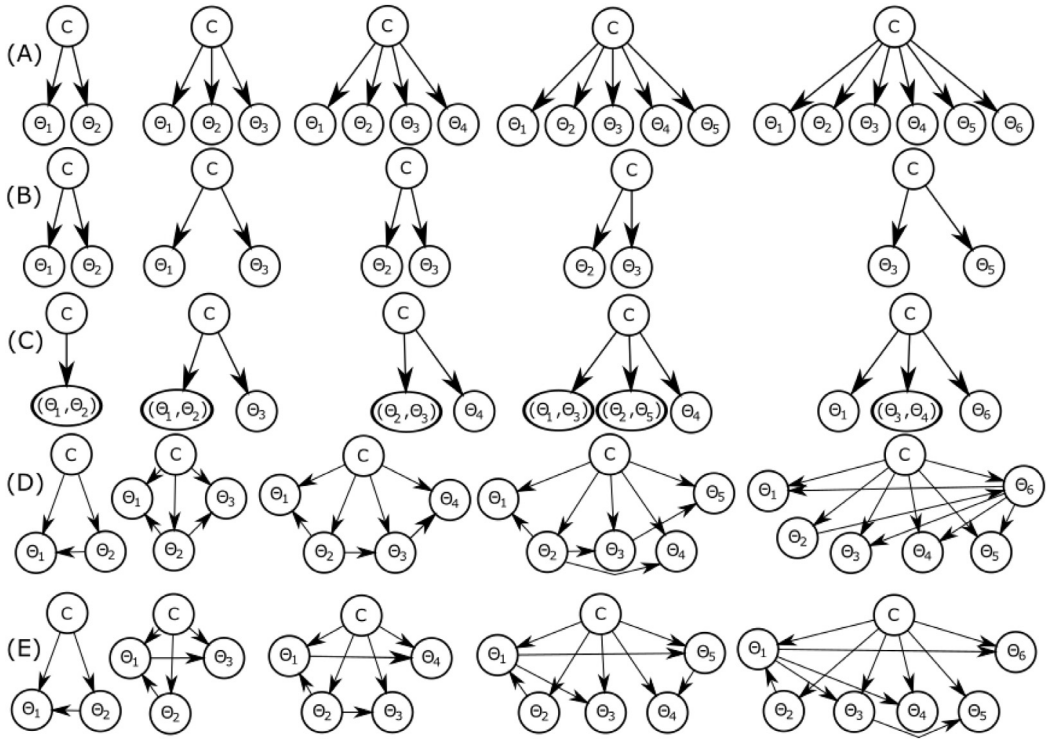

Fig. 8. Bayesian network classifier structures associated with the battery of classifiers depending on the maximum bifurcation order, for each type of classification algorithm: (A) wCNB, (B) wCsNB, (C) wCsmNB, (D) wCTAN, (E) GTAN.

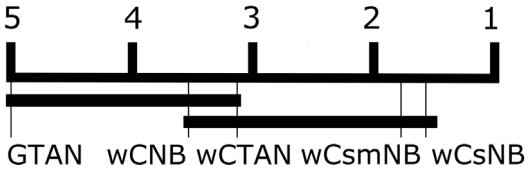

Fig. 9. Demšar diagram for the comparison of wCNB, wCsNB, wCsmNB, wCTAN and GTAN classification models using Friedman test and Nemenyi post-hoc test.

Table 6
Mean ± standard deviation of layers *II, III, IV, Va, Vb* and *VI* classification accuracy results of the battery of classifiers for each type of classifier applied over the dataset of dendritic bifurcation angles coming from the basal arbors of 288 3D pyramidal neurons of P14 rat S1HL neocortex. Bolded results are best performing classifiers.

		Classifiers				
		wCNB	wCsNB	wCsmNB	wCTAN	GTAN
Max. bifurc. angle	2	0.182 ± 0.034	**0.184 ± 0.055**	0.182 ± 0.039	0.182 ± 0.034	0.158 ± 0.038
	3	0.191 ± 0.035	**0.219 ± 0.035**	0.203 ± 0.057	0.205 ± 0.057	0.113 ± 0.030
	4	0.222 ± 0.016	0.224 ± 0.034	**0.239 ± 0.057**	0.222 ± 0.046	0.212 ± 0.081
	5	0.196 ± 0.091	0.235 ± 0.127	**0.239 ± 0.063**	0.189 ± 0.055	0.128 ± 0.131
	6	0.220 ± 0.113	0.270 ± 0.094	**0.290 ± 0.137**	0.240 ± 0.126	0.047 ± 0.069

We applied the Friedman non-parametric test to detect statistically significant differences in the results provided by our algorithms. Since the null hypothesis that there is no significant difference was rejected (p-value = 0.004), we used Nemenyi post-hoc test to determine which pairwise of algorithms was the cause of the Friedman test rejection. In Fig. 9, the statistically significant differences between our classifiers are represented as a Demšar diagram. We noted that there are no statistically significant differences between our classification algorithms except for two cases; between the wCTAN and wCsNB and between GTAN and the wCsmNB.

Therefore, we can conclude that (i) apart from the difficulty identifying the layer a case belongs to, it seems reasonable to use any of our four proposed circular classifiers for this neuronal dataset, since there are no any statistically significant differences between them and (ii) GTAN is never recommended.

6. Conclusions and future work

Introducing the first set of supervised Bayesian classification models capable of dealing with circular wrapped Cauchy predictive variables was the main objective of this paper. We have presented four models and their algorithms, designed to perform classification. We demonstrated using synthetic data that these models could perform classification accurately given circular datasets. We also provided evidence of the improvement of the circular classifiers over linear classifiers for datasets of circular nature that follow wrapped Cauchy distributions.

We performed statistical comparisons among the classifiers using synthetic data with 50, 200 and 1000 instances. Based on the results, we realized that the wCsmNB, the wCTAN and the wCNB are the best classification models for circular data that follows wrapped Cauchy distributions, with no statistically significant differences among them. The linear classifier never outperformed any of the wrapped Cauchy classifiers

For each of our new proposals, we evaluated a battery of classifiers using a real-world neuroscience dataset, in order to predict the layer that an instance belongs to. Results revealed that all of our four classification models are suitable. Performing Friedman test and its corresponding Nemenyi post-hoc test after rejection, we realized that there are no any statistically significant differences between wCNB, wCsNB, wCsmNB and wCTAN for this dataset. Wrapped Cauchy classifiers always outperformed their linear (Gaussian) counterparts.

The models shown in this paper are limited to no more than bivariate relationships. In future work, we intend to develop multivariate models in order to extend the Bayesian network classifiers for circular data to other more-sophisticated Bayesian network models (like k-dependence Bayesian network classifiers) capable of representing and taking into account multivariate relationships between circular variables — a difficult task due to the non-closed nature of the circular families that are known to date.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the TIN2016-796842-P and Cajal Blue Brain (C080020-09) projects. This project has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under Specific Grant Agreement No. 785907 (HBP SGA2). I.L. is supported by the Spanish Ministry of Education, Culture and Sport Fellowship (FPU13/01941). The authors thankfully acknowledge the Cortical Circuits Laboratory (CSIC-UPM) for providing the neuron dataset.

References

[1] K. Schmidt-Koenig, On the role of the loft, the distance and the site of release in pigeon homing, Biol. Bull. 125 (1963) 154–164.
[2] L. Morellato, L. Alberti, I. Hudson, Application of circular statistics in plant phenology: a case studies approach, in: Phenological Research: Methods for Environmental and Climate Change Analysis, Springer, 2010, pp. 339–359.
[3] N.I. Fisher, Statistical Analysis of Circular Data, Cambridge University Press, 1995.
[4] F. Fisher, Dispersion on a sphere, Proc. Roy. Soc. Lond. A Math. Phys. Eng. Sci. 217 (1130) (1953) 295–305.
[5] J. Graham, The stability and significance of magnetism in sedimentary rocks, J. Geophys. Res. (1949) 131–167.
[6] J. Gill, D. Hangartner, Circular data in political science and how to handle it, Political Anal. 18 (3) (2010) 316–336.
[7] A. Banerjee, I. Dhillon, J. Ghosh, S. Sra, Clustering on the unit hypersphere using von Mises-Fisher distributions, J. Mach. Learn. Res. 6 (2005) 1345–1382.
[8] A. Barros, J. Pereira, U. Lund, Identifying geographical patterns of wildfire orientation: A watershed-based analysis, Forest Ecol. Manag. 264 (2012) 98–107.
[9] J. Bowers, I. Morton, G. Mould, Directional statistics of the wind and waves, Appl. Ocean Res. 22 (1) (2000) 13–30.
[10] J.A. Carta, C. Bueno, P. Ramrez, Statistical modelling of directional wind speeds using mixtures of von Mises distributions: Case study, Energy Convers. Manage. 49 (5) (2008) 897–907.
[11] E. Batschelet, Circular Statistics in Biology, Academic Press, 1981.
[12] T. Harder, W. Boomsma, M. Paluszewski, J. Frellsen, K.E. Johansson, T. Hamelryck, Beyond rotamers: a generative, probabilistic model of side chains in proteins, BMC Bioinformatics 11 (1) (2010) 306.
[13] C. Bielza, P. Larrañaga, Discrete Bayesian network classifiers: A survey, ACM Comput. Surv. 47 (1) (2014) 5.
[14] I. Leguey, C. Bielza, P. Larrañaga, A. Kastanauskaite, C. Rojo, R. Benavides-Piccione, J. DeFelipe, Dendritic branching angles of pyramidal cells across layers of the juvenile rat somatosensory cortex, J. Comp. Neurol. 524 (13) (2016) 2567–2576.
[15] S.R. Jammalamadaka, A. SenGupta, Topics in Circular Statistics, World Scientific, 2001.
[16] K.V. Mardia, P.E. Jupp, Directional Statistics, John Wiley & Sons, 2009.
[17] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, The MIT Press, 2009.
[18] J. Pearl, Probabilistic reasoning in intelligent systems: Networks of plausible reasoning, Morgan Kaufmann 23 (1988) 33–34.
[19] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification. 2nd edition, John Wiley & Sons.
[20] J.E. Morris, P. Laycock, Discriminant analysis of directional data, Biometrika 61 (2) (1974) 335–341.
[21] M. Romanazzi, Discriminant analysis with high dimensional von Mises-Fisher distributions, in: 8th Annual International Conference on Statistics, 2014, pp. 1–16.
[22] S. El Khattabi, F. Streit, Identification analysis in directional statistics, Comput. Statist. Data Anal. 23 (1) (1996) 45–63.
[23] A. Figueiredo, P. Gomes, Discriminant analysis based on the Watson distribution defined on the hypersphere, Statistics 40 (5) (2006) 435–445.
[24] A. SenGupta, S. Roy, A simple classification rule for directional data, in: Advances in Ranking and Selection, Multiple Comparisons, and Reliability, Springer, 2005, pp. 81–90.
[25] A. SenGupta, F.I. Ugwuowo, A classification method for directional data with application to the human skull, Comm. Statist. Theory Methods 40 (3) (2011) 457–466.
[26] M.J. Kirby, R. Miranda, Circular nodes in neural networks, Neural Comput. 8 (2) (1996) 390–402.
[27] K. Fernandes, J.S. Cardoso, Discriminative directional classifiers, Neurocomputing 207 (2016) 141–149.
[28] P.L. López-Cruz, C. Bielza, P. Larrañaga, Directional naive Bayes classifiers, Pattern Anal. Appl. 18 (2) (2015) 225–246.
[29] R. von Mises, Über die "Ganzzahligkeit" der Atomgewichte und verwandte Fragen, Z. Phys. 19 (1918) 490–500.

[30] K.V. Mardia, Statistics of directional data, J. R. Stat. Soc. Ser. B Stat. Methodol. (1975) 349–393.
[31] K.V. Mardia, G. Hughes, C.C. Taylor, H. Singh, A multivariate von Mises distribution with applications to bioinformatics, Canad. J. Statist. 36 (1) (2008) 99–109.
[32] P. Lévy, L'addition des variables aléatoires définies sur une circonférence, Bull. Soc. Math. France 67 (1939) 1–41.
[33] A. Wintner, On the shape of the angular case of cauchy's distribution curves, The Annals of Mathematical Statistics 18 (4) (1947) 589–593.
[34] P. McCullagh, Möbius transformation and Cauchy parameter estimation, Ann. Statist. 24 (2) (1996) 787–808.
[35] S. Kato, A. Pewsey, A Möbius transformation-induced distribution on the torus, Biometrika 102 (2) (2015) 359–370.
[36] I. Leguey, C. Bielza, P. Larrañaga, Tree-structured Bayesian networks for wrapped Cauchy directional distributions, in: Advances in Artificial Intelligence, Vol. 9868, Springer, 2016, pp. 207–216.
[37] K. Bowman, L. Shenton, Methods of moments, Encyclopedia Stat. Sci. 5 (1985) 467–473.
[38] P. Langley, S. Sage, Induction of selective Bayesian classifiers, in: Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1994, pp. 399–406.
[39] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.
[40] R. Blanco, I. Inza, M. Merino, J. Quiroga, P. Larrañaga, Feature selection in Bayesian classifiers for the prognosis of survival of cirrhotic patients treated with TIPS, J. Biomed. Inform. 38 (5) (2005) 376–388.
[41] I. Kononenko, Semi-naive Bayesian classifier, in: European Working Session on Learning, Springer, 1991, pp. 206–219.
[42] M.J. Pazzani, Constructive induction of Cartesian product attributes, in: Feature Extraction, Construction and Selection, Springer, 1998, pp. 341–354.
[43] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29 (1997) 131–163.
[44] C. Chow, C. Liu, Approximating discrete probability distributions with dependence trees, IEEE Trans. Inform. Theory 14 (3) (1968) 462–467.
[45] D. Geiger, D. Heckerman, Learning Gaussian networks, in: Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1994, pp. 235–243.
[46] R Development Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing (2008).
[47] B. Mihaljevic, C. Bielza, P. and Larrañaga, bnclassify: Learning discrete Bayesian network classifiers from data, R package version 0.3.2 (2015). URL https://cran.r-project.org/src/contrib/Archive/bnclassify/.
[48] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Amer. Statist. Assoc. 32 (200) (1937) 675–701.
[49] P. Nemenyi, Distribution-free multiple comparisons, Biometrics 18 (2) (1962) 263.
[50] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[51] N.H. Kuiper, Tests concerning random points on a circle, Indag. Math. (N.S.) 63 (1960) 38–47.

Ignacio Leguey is currently a Visiting Professor with the Department of Economía Financiera y Contabilidad e Idioma Moderno, Universidad Rey Juan Carlos de Madrid, Spain. His research interests include directional statistics, supervised classification, particularly Bayesian network classifiers, probabilistic graphical models, and real applications, like meteorology and neuroscience.

Concha Bielza is currently a Full Professor with the Department of Artificial Intelligence, Universidad Politécnica de Madrid, Spain. Her research interests include probabilistic graphical models, decision analysis, metaheuristics for optimization, classification models, and real applications, like bioinformatics, neuroscience and industry. She was awarded the 2014 UPM Research Prize.

Pedro Larrañaga is currently a Full Professor with the Department of Artificial Intelligence, Universidad Politécnica de Madrid, Spain. His research interests include probabilistic graphical models, data mining, classification models, and real applications, like biomedicine and neuroscience. He is ECCAI fellow since 2012 and he has been awarded the 2013 Spanish National Prize in Computer Science.