Hindawi
Complexity
Volume 2018, Article ID 4075656, 14 pages
https://doi.org/10.1155/2018/4075656

Research Article

Predicting Facial Biotypes Using Continuous Bayesian Network Classifiers

Gonzalo A. Ruz [1,2] **and Pamela Araya-Díaz** [3]

[1] *Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Av. Diagonal Las Torres 2640, Peñalolén, Santiago, Chile*
[2] *Center of Applied Ecology and Sustainability (CAPES), Santiago, Chile*
[3] *Departamento del Niño y Adolescente, Área de Ortodoncia, Facultad de Odontología, Universidad Andrés Bello, Santiago, Chile*

Correspondence should be addressed to Gonzalo A. Ruz; gonzalo.ruz@uai.cl

Received 29 June 2018; Revised 7 November 2018; Accepted 15 November 2018; Published 2 December 2018

Guest Editor: Panayiotis Vlamos

Bayesian networks are useful machine learning techniques that are able to combine quantitative modeling, through probability theory, with qualitative modeling, through graph theory for visualization. We apply Bayesian network classifiers to the facial biotype classification problem, an important stage during orthodontic treatment planning. For this, we present adaptations of classical Bayesian networks classifiers to handle continuous attributes; also, we propose an incremental tree construction procedure for tree like Bayesian network classifiers. We evaluate the performance of the proposed adaptations and compare them with other continuous Bayesian network classifiers approaches as well as support vector machines. The results under the classification performance measures, accuracy and kappa, showed the effectiveness of the continuous Bayesian network classifiers, especially for the case when a reduced number of attributes were used. Additionally, the resulting networks allowed visualizing the probability relations amongst the attributes under this classification problem, a useful tool for decision-making for orthodontists.

1. Introduction

In orthodontics, it is essential to know the changes that occur during facial growth when planning a treatment, especially in children and adolescents, because the amount and direction of growth can significantly alter the need for different treatment mechanics [1, 2]. Normally, clinicians use radiographs or photographs to compute angular, linear, or proportional measurements of the face and skull to obtain growth patterns or facial biotypes [3]. One of the most popular methods to determine the facial biotype is through the VERT index proposed by Ricketts [4]. The VERT index is computed using five different features (or attributes) that allows analyzing the facial morphology [5]. Based on the VERT index, the biotypes can be classified into Dolichofacial (long and narrow face), Brachyfacial (short and wide face), and an intermediate type called Mesofacial [3, 5]. These three biotypes are shown in Figure 1.

It has been described that some attributes used in the VERT index can alter the index in patients in whom the sagittal relationship between the jaws is altered, leading to possible diagnostic errors [3]. That is why, the possibility of automatically determining the facial biotype using attributes that are not altered by the sagittal position of the jaws would eliminate the errors observed with the use of the VERT index. Thus, in this work, we propose a machine learning approach to automatically classify a patient's biotype using alternative attributes.

In recent years, we have seen great advances in the field of machine learning in relation to predictive modeling, in particular, supervised learning algorithms for classification and regression problems, such as random forests (RF) [6], support vector machines (SVM) [7], neural networks with random weights such as feedforward neural networks with random weights (RWSLFN) [8], random variable functional link neural networks (RVFLN) [9], and extreme learning machine (ELM) [10]. All of these models are achieving extraordinary performances in several applications, including orthodontics, such as the automatic Dent-landmark detection in 3D cone-beam computed tomography dental data [11], a method

(a) (b) (c)

Figure 1: Examples of the three facial biotypes. (a) Dolichofacial, (b) Brachyfacial, and (c) Mesofacial.

that objectively evaluates orthodontic treatment need and treatment outcome from the lay perspective [12], pattern classification for finding facial growth abnormalities [13], and an automated diagnostic imaging system for orthodontic treatment in dentistry [14], just to mention a few.

While high accuracy in the predictions and good generalization power are the main goals in several applications, the use of machine learning in medical treatment planning requires additionally that these models should be simple to interpret and therefore use them as a tool for decision-making. The algorithms mentioned before, although very powerful from a quantitative point of view, are somewhat limited from a qualitative aspect, in the sense that, for example, a trained SVM classifier, does not give you explicit classification rules or a simple visual interpretation on how the attributes interact in order to obtain the classification of a new data point. This issue has been tackled by other types of machine learning techniques, where the qualitative aspect plays a key role such as inductive learning algorithms [15–18] and decision trees [19]. These techniques are known as white box models (opposite to the black box models mentioned before) since the prediction process is open to the user. An interesting machine learning model that combines probability theory (quantitative) with graph theory for visualization (qualitative) is Bayesian networks introduced by J. Pearl [20], and in particular for this work, Bayesian network classifiers [21]. A Bayesian network (BN) is a directed acyclic graph (DAG), whose nodes represent discrete attributes and the edges probabilistic relationships among them. Additionally, each node has associated a conditional probability table, indicating the conditional probability for each discrete value of the node conditioned for each value of the parent nodes in the network (graph). The structure of the graph encodes the assertion that each attribute (node) is conditionally independent of its nondescendants, given its parents in the graph (this is known as the *Markov condition*). Therefore, given that a Bayesian network satisfies the Markov condition, the joint probability distribution of all the attributes can be computed in a factorized form. Bayesian networks have been applied in the domain of dentistry, for example, a decision-making system for the treatment of dental caries [22], the assessment of tooth color changes due to orthodontic treatment [23], the evaluation of the relative role and possible causal relationships among various factors affecting the diagnosis and final treatment outcome of impacted maxillary canines [24], to establish a ranking in efficacy and the best technique for coronally advanced flap-based root coverage procedures [25], a minimally invasive technique for lateral maxillary sinus elevation and to identify the relationship between the involved factors [26], and the development of a clinical decision support system to help general practitioners assess the need for orthodontic treatment in patients with permanent dentition [27].

Learning Bayesian networks from data has two components that must be handled: (1) the structure of the networks and (2) the parameters (conditional probability tables). It has been proven that learning Bayesian networks is NP-complete [28]. Therefore, several approximate learning approaches have been devised in order to simplify the learning process [29–32].

In this paper, we consider the problem of facial biotype classification using Bayesian network classifiers with continuous attributes. The rest of the paper is organized as follows. Section 2 presents a general overview of Bayesian network classifiers; then in Section 3 we describe the dataset used in this work, the continuous attribute adaptation for common Bayesian network classifiers, a description of an incremental tree construction procedure for tree like Bayesian networks, other continuous Bayesian network classifiers approaches, and the simulation setup to test and compare the classifiers. The results and discussion appear in Section 4; then the final conclusions are given in Section 5.

2. Background

Probabilistic classification consists in computing a posterior probability given an input data point. We will use the standard notation in Bayesian networks, where random variables (attributes) are denoted by capital letters, e.g., X, and particular values with lower-case letters, e.g., x. Let us consider a training set D consisting of N data points, each one characterized by n attributes $X_1, \ldots, X_n$ and their respective

output Y or class label (with c classes). Given a new input data point $\boldsymbol{v}$, this can be classified using the Bayes rule,

$$\begin{aligned} k^{\text{predict}} &= \underset{k}{\operatorname{argmax}}\, P\left(Y = k \mid X_1 = v_1, \ldots, X_n = v_n\right) \\ &= \underset{k}{\operatorname{argmax}}\, \frac{P\left(Y = k\right) P\left(X_1 = v_1, \ldots, X_n = v_n \mid Y = k\right)}{\sum_{y'=1}^{c} P\left(Y = y'\right) P\left(X_1 = v_1, \ldots, X_n = v_n \mid Y = y'\right)} \\ &= \underset{k}{\operatorname{argmax}}\, \beta P\left(Y = k\right) P\left(X_1 = v_1, \ldots, X_n = v_n \mid Y = k\right) \end{aligned} \tag{1}$$

with β the normalizing constant. From (1), we notice that there are two probabilities that can influence the resulting prediction. The first one is $P(Y = k)$ (with $k = 1, \ldots, c$) which is known as the *a priori* probability for the class value k and represents the class k distribution in D. The computation of this probability is simple, since it consists in counting the number of training examples in D for which $Y = k$ and then dividing this value by N. The second probability, $P(X_1 = v_1, \ldots, X_n = v_n \mid Y = k)$, is called the *likelihood* and corresponds to the joint probability distribution of the attributes conditioned to the class k. There are several methods to compute the joint probability distribution, in particular, using Bayesian networks, thus, given way to *Bayesian network classifiers*. The simplest approach is to consider "naively" that the attributes are independent amongst them given the class, which yields the *naive Bayesian (NB) network classifier* [33]. The prediction is computed by

$$\begin{aligned} k^{\text{predict}} &= \underset{k}{\operatorname{argmax}}\, P\left(Y = k \mid X_1 = v_1, \ldots, X_n = v_n\right) \\ &= \underset{k}{\operatorname{argmax}}\, \beta P\left(Y = k\right) \prod_{i=1}^{n} P\left(X_i = v_i \mid Y = k\right). \end{aligned} \tag{2}$$

An example of the Bayesian network representation (with $n = 5$) of this classifier is shown in Figure 2(a).

Given the difficulty of learning Bayesian networks from data, as discussed before, learning strategies have considered restrictions on the type of the structure of the network. That is the case with the seminal work by Chow and Liu [34], which developed a learning algorithm for approximating the joint distribution by a tree structure, i.e., a network with $n-1$ edges, where one node acts as the root (no incoming edges only outgoing edges), and all the rest of the nodes have only one parent node. Let Π_i represent the parent node of the attribute X_i (for $i = 1, \ldots, n$); also let i^* be the index of the node which acts as the root; therefore, $\Pi_{i^*} = \{\emptyset\}$. Under this scheme, the training set D is partitioned according to the different class labels. Then for each partition, a tree structure is learned to model the corresponding joint probability distribution P_k (with $k = 1, \ldots, c$). The prediction is computed by

$$\begin{aligned} k^{\text{predict}} &= \underset{k}{\operatorname{argmax}}\, P\left(Y = k \mid X_1 = v_1, \ldots, X_n = v_n\right) \\ &= \underset{k}{\operatorname{argmax}}\, \beta P\left(Y = k\right) \prod_{i=1}^{n} P_k\left(X_i = v_i \mid \Pi_i\right). \end{aligned} \tag{3}$$

This model is also known as the Chow-Liu (CL) classifier. An example of the CL classifier (for $n = 5$ and $c = 2$) is shown in Figure 2(b). Notice that given that $k \in \{1, \ldots, c\}$, i.e., there are c different class labels, then the CL classifier must learn c tree structures. An alternative to this is the model called the tree augmented naive Bayes classifier or TAN [21], which learns only one tree structure for all the classes. Under this model, $\Pi_i = \{X_j, Y\}$; i.e., for each node X_i, the parent set Π_i is composed of two nodes: X_j (with $j \neq i$) and the class variable Y, with exception of i^* (the attribute root node), where $\Pi_{i^*} = \{Y\}$. The prediction using the TAN classifier can be obtained by

$$\begin{aligned} k^{\text{predict}} &= \underset{k}{\operatorname{argmax}}\, P\left(Y = k \mid X_1 = v_1, \ldots, X_n = v_n\right) \\ &= \underset{k}{\operatorname{argmax}}\, \beta P\left(Y = k\right) \prod_{i=1}^{n} P\left(X_i = v_i \mid \Pi_i\right). \end{aligned} \tag{4}$$

An example of the TAN classifier (with $n = 5$) is shown in Figure 2(c).

Of course, there are other BN classifier approaches, such as *Markov blanket* of the class variable [35], K2-attribute selection (K2-AS) algorithm [36], semi-naive Bayes model [37], k-dependence Bayesian classifier [38], Bayesian classifier inference using Bayes factor [39], etc. A complete review of discrete Bayesian network classifiers can be found in [40].

It is interesting to notice that while TAN was presented as a solution to the strong independence assumption in the naive Bayes classifier, in the tests presented in the TAN paper [21], there are cases where the naive Bayes outperformed TAN. Can it be that given that TAN forces a tree structure amongst the attributes, there may be edges in the network which should not exist but are there in order to satisfy the tree structure? With this in mind, in this paper, we propose an incremental tree construction procedure which may lead to an incomplete tree structure, known as a *forest*.

3. Methods

3.1. Dataset Description and Preprocessing. The dataset consists of 182 lateral teleradiographies from Chilean patients. For each one, cephalometric analysis was performed to compute 31 continuous attributes (see Appendix) that characterize the craniofacial morphology. This dataset has been used previously to identify craniofacial patterns through clustering analysis [41]. For this work, each lateral teleradiograph has been manually classified and validated by orthodontists into one of the three classes (Brachyfacial, Dolichofacial, and Mesofacial). A visualization of the correlation matrix of the 31 attributes is shown in Figure 3, where we can appreciate that there are several attributes which are highly (more than 0.8 in absolute value) correlated.

Highly correlated attributes are essentially attributes which capture the same information, and therefore we can reduce the number of attributes by leaving only one attribute from a highly correlated set of attributes. For example, from Figure 3 we notice that Ril0 and Mc3 are highly correlated (a correlation of 0.95); this is not surprising since both attributes indicate the sagittal position of the maxilla with respect to the skull, using different cephalometric landmarks. Therefore,

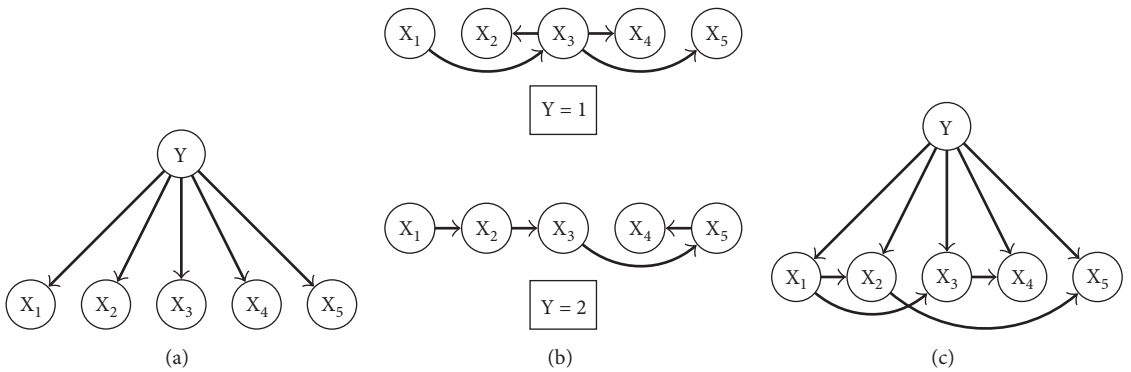

Figure 2: Examples of different Bayesian networks classifiers: (a) naive Bayes, (b) Chow-Liu tress, and (c) TAN.

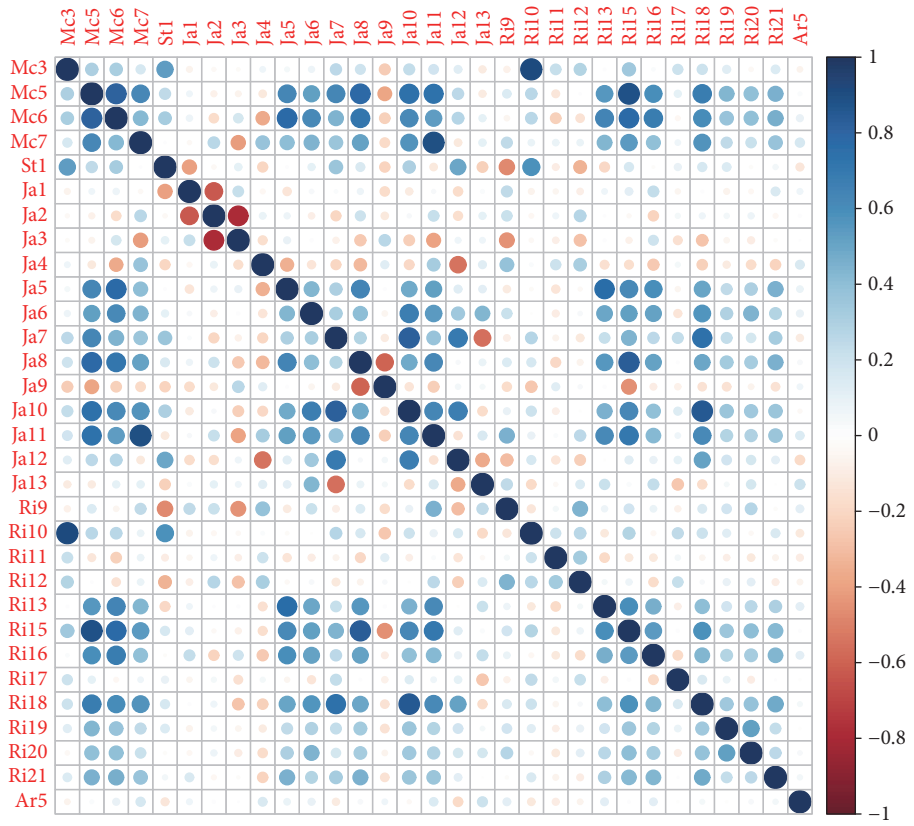

Figure 3: Correlation matrix of the 31 attributes.

we may drop Ri10 in further analyses and use only Mc3. By assuming a threshold of absolute value of 0.8 for the correlation, we excluded the following attributes: Mc5, Mc6, Ri10, Ri18, Ja8, Ja10, and Ja11. Thus, the number of attributes of the dataset is now 24. From these remaining attributes, we proceeded in visualizing their discriminatory power by performing a principal component analysis (PCA) projection of the 24-dimensional data points to a 2-dimensional space; then each point is labeled according to their class (facial biotype). The resulting visualization is shown in Figure 4.

From Figure 4, we notice that while the attributes have sufficient discriminatory power to separate the Brachyfacial class with the Dolichofacial class, the third Mesofacial class lies just between the other two, making this a difficult classification problem.

3.2. Continuous Bayesian Network Classifiers. As explained in the Introduction, we will consider Bayesian networks for this facial biotype classification problem. Given that Bayesian networks were originally formulated for discrete random variables, and our dataset has continuous variables (attributes), we need to address this issue. A typical approach is to discretize the continuous attributes and then proceed as usual. While this is a practical solution, an ideal

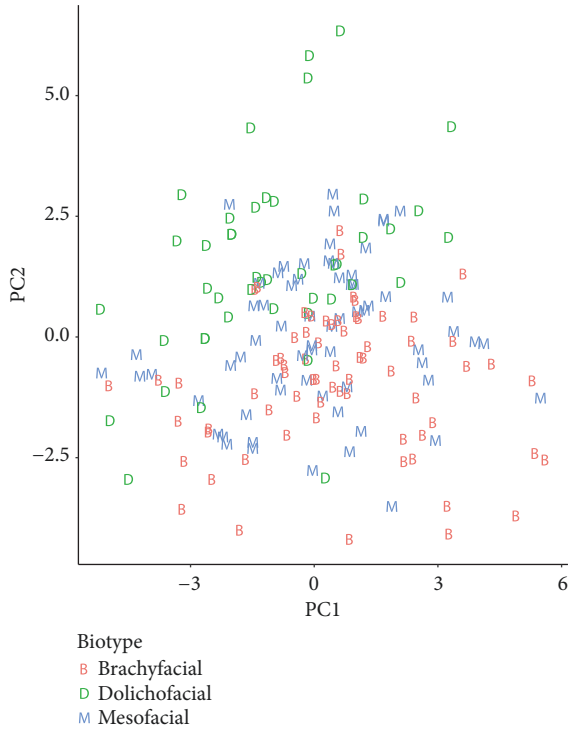

Figure 4: PCA projection of the 24-dimensional data points.

discretization is not that straightforward, and therefore, valuable information may be lost during this process. In what follows, we describe the continuous adaptation for the naive Bayes, TAN, and an incremental tree construction version of TAN, through the implementation in R, the open source software environment for statistical computing and graphics [42], that we used in our work.

3.2.1. Continuous Naive Bayes Classifier. The classification under this model is computed by (2). Here, we need to estimate the class priors $P(Y)$ and the conditional probabilities $P(X_i \mid Y)$ for $i = 1 \ldots, n$. The class priors are straightforward and can be computed by the relative frequency of each class value (Brachyfacial, Dolichofacial, and Mesofacial) in the training set. For the conditional probabilities, we partition the training set examples accordingly to their class, then for each partition we use the kernel density estimator with Gaussian kernels to compute the desired densities. The kernel density estimator function in R is called *density*. Then we use the *approx* function in R that performs linear interpolation from the estimated density to obtain the value of $P(X_i = x_i \mid Y)$ for a specific value x_i.

3.2.2. Continuous TAN Classifier. In this case, predictions are computed by (4). To evaluate in (4) we need the resulting tree structure. TAN finds this tree by applying the maximum weighted spanning tree algorithm (Kruskal's algorithm [43] or Prim's algorithm [44]) over a fully connected undirected graph of the attributes where the weights are given by the conditional mutual information measure. For the discrete case, given two attributes X_i and X_j $(i \neq j)$ with their values x_i and x_j, respectively, and the class variable Y, this measure is computed by

$$I\left(X_i; X_j \mid Y\right) = \sum_{x_i, x_j, y} P\left(x_i, x_j, y\right) \log \frac{P\left(x_i, x_j, y\right)}{P\left(x_i \mid y\right) P\left(x_j \mid y\right)}. \tag{5}$$

This is a nonnegative quantity that measures the information that X_j provides about X_i when the value of Y is known. For continuous variables, the mutual information between two attributes is given by

$$I\left(X_i; X_j\right) = \int_{X_j} \int_{X_i} p\left(x_i, x_j\right) \log \frac{p\left(x_i, x_j\right)}{p\left(x_i\right) p\left(x_j\right)} dx_i\, dx_j. \tag{6}$$

Then, the conditional mutual information for the continuous case can be computed by

$$I\left(X_i; X_j \mid Y\right) = \sum_y P\left(Y\right) I\left(X_i; X_j \mid Y\right). \tag{7}$$

So, $I(X_i; X_j \mid Y)$ can be computed for each class value k (with $k = 1, \ldots, c$) by (6) using all the training examples, where $Y = k$. We estimate (6) using the *knnmi* function available in the parmigene package in R [45]. This function estimates the mutual information between two attributes using entropy estimates from k-nearest neighbors distances [46]. Once we have computed the conditional mutual information for each pair of attributes, we construct the fully connected graph with the *graph.full* function in the igraph package in R [47]. Then the tree structure is obtained from the fully connected graph by using the *minimum.spanning.tree* function (also in igraph) that uses Prim's algorithm. Since we are interested in the maximum spanning tree, we use *minimum.spanning.tree* with the negative values of the conditional mutual information as weights. The resulting tree is undirected. To obtain the directed tree, we identify which is the pair of attributes with the highest edge weight (conditional mutual information), we consider from the winning pair one of the attributes as the root, and then we set the direction of all the remaining edges to be outward from it. To finally obtain the TAN classifier, we add an edge from Y to each attribute X_i. Now we are in conditions to compute (4) for a given data point. The priors $P(Y)$ can be computed as usual through relative frequencies. Then the terms $P(X_i \mid \Pi_i)$ in the product are computed as follows. For the root attribute i^* we have that $\Pi_{i^*} = Y$; thus, we can use the kernel density approach described for the naive Bayes classifier. For the rest of the terms in the product, we will have $\Pi_i = \{X_j, Y\}$ given by the tree structure. Therefore, we need to estimate conditional probabilities such as $P(X_i \mid X_j, Y)$. Using the product rule, we have that $P(X_i, X_j) = P(X_i \mid X_j)P(X_j)$. So, if we partition the training data set accordingly to the class, we can estimate the joint

probability $P(X_i, X_j)$ and the marginal probability $P(X_j)$ for each partition, then

$$P\left(X_i \mid X_j, Y\right) = \frac{P\left(X_i, X_j \mid Y\right)}{P\left(X_j \mid Y\right)}. \tag{8}$$

We estimate the joint probability with a two-dimensional kernel approach. In particular, we use the function *kde2d* in the MASS package in R [48]. This function performs a two-dimensional kernel density estimation with an axis-aligned bivariate normal kernel, evaluated on a square. Then, to obtain specific values from this density, we use the *interp.surface* function from the fields package in R [49]. This function uses bilinear weights to interpolate values on a rectangular grid to desired values. Finally, this joint probability estimate is normalized by $P(X_j \mid Y)$ which can be computed using the same approach used for the naive Bayes classifier.

3.2.3. Continuous Incremental Tree Construction Augmented Naive Bayes Classifier. We propose an alternative learning procedure for the TAN classifier, which we call incremental tree construction augmented naive Bayes (ITCAN). One of the limitations of the TAN model is that the resulting structure will always be a tree, even if some edges have very low weights (conditional mutual information). With ITCAN, we identify partial TAN solutions where some nodes (attributes) might end up with only the incoming edge from the class. The ITCAN learning procedure with a training set is as follows:

(1) Evaluate the accuracy of a naive Bayes classifier using k-fold cross validation. Let this value be A_{nb}.

(2) Learn the TAN tree structure as described in Section 3.2.2.

(3) Create a list with the edges in a descending order with respect to their weight ($edge_h$ for $h = 1, \ldots, n-1$).

(4) Assign $model \longleftarrow$ naive Bayes model.

(5) For each h in the list:

   (a) $model \longleftarrow model + edge_h$

   (b) Evaluate the accuracy of $model$ classifier using k-fold cross validation. Let this value be A_h.

(6) $h^* \longleftarrow \operatorname{argmax}_{x \in \{nb, 1, \ldots n-1\}} A_x$.

From the above learning procedure, if $h^* = nb$, then the resulting model is the naive Bayes classifier. If $h^* = n-1$, then the resulting model is the TAN classifier. For any other value of h^*, the resulting structure will be a forest, a midway solution between naive Bayes and TAN. For the results presented later on, we use $k = 5$ in the k-fold cross validation in the ITCAN learning procedure.

There have been other approaches to search for Bayesian network models bounded by naive Bayesian networks and the TAN classifier; one example is the Forest-Augmented Bayesian Network (FAN) algorithm [50]. While the ITCAN learns once the TAN tree structure, the FAN algorithm uses another approach. It first computes the conditional mutual information between all pairs of attributes, then it constructs the fully connected graph using the negative value of the conditional mutual information as weights between the attributes. But now instead of finding the minimum weighted spanning tree (like TAN), it searches for the minimum weighted spanning forest containing exactly k edges (with $k \geq 0$ defined by the user). So to explore the complete range of structures, the user must apply FAN n-times ($k = 0, \ldots, n-1$). Another difference is when FAN transforms the undirected forest into a directed forest, it does so by choosing a root vertex for every tree in the forest. This procedure could yield different structures when compared to ITCAN which uses the edges from the unique TAN structure.

3.2.4. Other Continuous Bayesian Network Classifiers Approaches. In [51] conditional Gaussian networks (CGN) classifiers were introduced. In particular, it is of interest for this work the Gaussian NB (gNB) and the Gaussian TAN (gTAN). In the case of gNB, the probabilities in the product term in (2) are approximated by

$$P\left(X_i \mid Y = k\right) \sim N\left(\mu_{i|k}, \sigma_{i|k}\right), \tag{9}$$

where $\mu_{i|k}$ and $\sigma_{i|k}$ are the mean and the standard deviation, respectively, of attribute X_i, computed by using only the examples that have a class value $Y = k$. For gTAN, the probabilities in the product term in (4) are approximated by

$$P\left(X_i \mid \Pi_i\right) \sim N\left(m_{i|k}, v_{i|k}\right), \tag{10}$$

where $m_{i|k}$ and $v_{i|k}$ are defined by

$$m_{i|k} = \mu_{i|k} + \beta_{ij|k}\left(x_j - \mu_{j|k}\right) \tag{11}$$

$$v_{i|k} = \sigma^2_{i|k} - \frac{\sigma^2_{ij|k}}{\sigma^2_{j|k}} \tag{12}$$

where we have considered X_j as the parent attribute of X_i. $\beta_{ij|k}$ is the regression coefficient of X_i on X_j conditioned to the class value $Y = k$, defined by

$$\beta_{ij|k} = \frac{\sigma_{ij|k}}{\sigma^2_{j|k}}, \tag{13}$$

where $\sigma_{ij|k}$ is the covariance between the variables X_i and X_j conditioned to k and $\sigma^2_{j|k}$ is the variance of X_j conditioned to k.

Also important to point out, under this approach, is that the conditional mutual information is computed by

$$I\left(X_i; X_j \mid Y\right) = -\frac{1}{2}\sum_{k=1}^{c} P\left(Y = k\right) \log\left(1 - \rho_k^2\left(X_i, X_j\right)\right), \tag{14}$$

where $\rho_k(X_i, X_j) = \sigma_{ij|k} / \sqrt{\sigma^2_{i|k}\sigma^2_{j|k}}$ is the correlation coefficient between X_i and X_j conditioned to the class value $Y = k$.

Another approach to handle continuous attributes is described in [52], where kernel density estimation is adopted

(similar to the approach presented in this paper) giving way to the so-called *flexible* classifiers. The flexible naive Bayes (fNB) classifier uses a similar approach as the one described in Section 3.2.1, where the conditional probabilities are computed with Gaussian kernels. One difference is the smoothing parameter h (used by the kernel density estimator) in fNB, which is the normal rule:

$$h = \left(\frac{4}{(m+2)\,q}\right)^{1/(m+4)}, \tag{15}$$

where m is the number of continuous variables in the density function to be estimated and q is the number of cases from which the estimator is learned. In our proposal, the smoothing parameter considered (used by the *density* function in R) is a rule-of-thumb described in [53]:

$$h = 0.9Aq^{-1/5} \tag{16}$$

with

$$A = \min\left(\text{standard deviation}, \frac{\text{interquartile range}}{1.34}\right). \tag{17}$$

The flexible tree augmented naive Bayes (fTAN) computes the conditional probabilities in the product term of (4) using (8) and employing a 2-dimensional Gaussian density with identity covariance matrix, similar to the continuous TAN proposed, but fTAN uses (15) to compute the bandwidth for the kernel, whereas our proposal uses (16) with the factor 0.9 changed to 4.24. Also, fTAN estimates the conditional mutual information in the following way:

$$\widehat{I}\left(X_i; X_j \mid Y\right) = \sum_{y=1}^{c} P(Y)\,\frac{1}{n_y}\sum_{l=y:1}^{y:n_y} \log \frac{\widehat{f}\left(x_i^l, x_j^l \mid y\right)}{\widehat{f}\left(x_i^l \mid y\right)\widehat{f}\left(x_j^l \mid y\right)} \tag{18}$$

where the super-index $y : j$ refers to the jth case in the partition induced by the value y, and n_y is the number of cases verifying that $Y = y$. $\widehat{f}$ are computed using the kernels described previously. On the other hand, in our proposal, we use another approach to estimate the conditional mutual information using entropy estimates from k-nearest neighbors distances [46].

Overall, when comparing to these previous continuous formulations (CGN and flexible), we notice that our proposal, based on kernel density estimates, resembles the flexible classifiers of [52], but with alternative implementations and using current available R functions.

3.3. Simulation Setup. We will compare the classification performance of the described continuous Bayesian network classifiers; in particular, we will compare our implementations, namely, cNB, cTAN, and cITCAN, with the conditional Gaussian networks approach: gNB, gTAN, and gITCAN, as well as the flexible approach: fNB, fTAN, and fITCAN. Also, we will consider the discrete versions: dNB, dTAN, and dITCAN. For this we will use the *discretize* function from the bnlearn package in R [54]. Finally we will also consider a black box classifier such as SVM. In particular, we use the *svm* function with default setting from the e1071 package in R [55].

Table 1: Performance measures for each classifier (with 24 attributes).

Algorithm	Accuracy %	Kappa
cNB	60.4±6.7	0.39±0.11
gNB	59.5±6.3	0.38±0.09
fNB	56.0±6.4	0.33±0.10
dNB	55.2±5.9	0.32±0.09
cTAN	56.8±6.3	0.34±0.09
gTAN	58.5±6.9	0.37±0.11
fTAN	43.9±6.4	0.15±0.10
dTAN	51.9±8.2	0.27±0.13
cITCAN	60.8±6.0	0.41±0.09
gITCAN	59.4±6.2	0.38±0.09
fITCAN	46.8±6.2	0.20±0.09
dITCAN	51.8±7.1	0.26±0.10
SVM	62.7±5.9	0.42±0.09

To compute the classification performance, we randomly sample 70% of the dataset examples to generate a training set and use the remaining 30% as a test set. We train the thirteen classifiers on the same training set and then compute the accuracy (the fraction of correct predictions) and the *kappa* statistic using the test set. The kappa statistic compares the accuracy of the trained model with the accuracy of a random model. To interpret the kappa value, we use the common characterization proposed in [56]: values $\leq$ 0 as indicating poor agreement, 0 − 0.2 as slight, 0.21 − 0.4 as fair, 0.41 − 0.6 as moderate, 0.61 − 0.8 as substantial, and 0.81 − 1 as almost perfect agreement.

We run the data splitting procedure 50 times and then report the average and the standard deviation of the accuracy and the kappa value for each run. To statistically compare the performance between all the algorithms we will consider the Friedman test and a post hoc test to evaluate the pairwise performance when all the algorithms are compared to each other; in particular, we will use the Nemenyi test. Further details of the process for comparison of multiple algorithms are given in [57].

4. Results and Discussions

The classification performance results for the thirteen classifiers are shown in Table 1. On average, the best performance was obtained by SVM, while within the Bayesian network classifiers, the cITCAN obtained the best performance. Also, considering the kappa value, only SVM and cITCAN correspond to the moderate interval of classification agreement with the true classes, whereas most of the other classifiers are in the fair interval. The worst performance was obtained by fTAN (and the second worst fITCAN); this could be due to the conditional mutual information estimation, where probably not enough samples were available to conduct

Table 2: The average ranks for all the algorithms (with 24 attributes).

Algorithm	Rank
SVM	2.94
cITCAN	4.37
cNB	4.65
gITCAN	4.99
gNB	5.09
gTAN	5.94
cTAN	6.96
fNB	6.99
dNB	7.81
dTAN	8.92
dITCAN	9.21
fITCAN	11.17
fTAN	11.96

a good estimation. It is important to point out that the standard deviation for the accuracy is high, and therefore, it is necessary to perform statistical tests to effectively compare the results.

We considered the null hypothesis to be tested that all the algorithms performed the same and that the observed differences were merely random. We conducted the Friedman test in order to analyze if there are statistically significant differences for all the algorithms. All the algorithms are ranked for each dataset (run) separately, where the best performing algorithm is the one obtaining the lowest rank. Table 2 shows the average rank for each algorithm.

The Friedman statistic is given by the following:

$$\chi_F^2 = \frac{12D}{c\,(c+1)} \left[\sum_j R_j^2 - \frac{c\,(c+1)^2}{4} \right], \tag{19}$$

where R_j^2 is the j-th average rank of the algorithms. The statistic is distributed according to χ_F^2 with $c - 1$ degrees of freedom and D is the number of datasets. For the comparison of all the algorithms with the Friedman test, the χ_F^2 statistic is 300.2 and the p value is <2.2e-16, which rejects the null hypothesis that all the algorithms have the same performance.

Then, a post hoc test is performed to evaluate the pairwise performance when all the algorithms are compared to each other. The Nemenyi test with $\alpha = 0.05$ was applied, and the results are presented in Table 3. When comparing SVM with all the other classifiers, we notice that the null hypothesis cannot be rejected when compared to cNB, gNB, cITCAN, and gITCAN, respectively, since there are no statistically significant differences between them, whereas for our second best classifier, cITCAN, we notice that the null hypothesis cannot be rejected when compared to cNB, gNB, gTAN, gITCAN, and SVM, respectively.

Figure 5 shows the best cTAN model obtained throughout the 50 runs. We notice that Ja5 and Ri15 are the two attributes with the most outgoing edges (apart from the obvious class node Biotype), conditioning the probabilities of the other attributes. In particular, Ja5 is the parent node of Ri13, Ri16, Ja13, Ja6, and Ri15. This can be explained, in part, by the following: Ja5 as well as Ri13 corresponds to the length of the anterior cranial base using different landmarks. Ja13 and Ja6 correspond to variables given by the posterior cranial length. In this case, the relationship is explained given the fact that the growth of the anterior cranial base (Ja5) and posterior cranial base (Ja13 and Ja16) depends on a common factor, which is the growth of the brain; therefore, there is a linear proportionality between both structures. There is no biologically direct relationship to explain the relation between Ja5 with Ri15 and Ri16, except that, as in any biological system, there is a proportional and compensatory relationship between the structures tending to maintain the functionality and stability of the systems.

On the other hand, Ri15 is the parent node of Ri21, Mc7, Ri20, and Mc3. In this case, a greater or smaller mandibular size is directly related to a larger or smaller size of all its components, such as the width of the symphysis (Ri21) and width of the condyle (Ri20), which explains the relationship between these variables and the size of the mandibular body (Ri15). On the other hand, there is no biologically direct relation to explain the relationship between attribute Ri15 with Mc3 and Mc7. Attribute Mc3 points out the sagittal position of the maxilla, which is independent of the size of the mandibular body (Ri15), and Mc7 is a vertical relationship (lower facial height) that is not directly influenced by the mandibular size.

The best cITCAN model obtained throughout the 50 runs is shown in Figure 6. We notice that it is a forest, where only 5 edges are considered from the total 23 of the cTAN model (without counting the outgoing edges of the class variable). Here we observe that the influence of Ja5 on Ri13 and Ri15 is still required.

We explore the possibility to improve the classification performance by identifying the most relevant attributes for classification and then proceed to repeat the simulations with a reduced number of attributes. For this, the *importance* function from the randomForest package in R [58] was used. This function computes the importance of each attribute based on the Gini importance, a measure used to quantify the node impurity during the tree inference process (in decision trees or random forests). The result is shown in Figure 7.

We observe that Ja4 is the attribute with the most discriminatory power. We proceed to select the top 4 attributes, i.e., Ja4, Ja12, Mc7, and Mc3. In particular, the first three correspond to measurements that describe vertical dimensions, which is directly related to the determination of the biotype, since the primary difference between them is the relationship between the vertical dimensions of the anterior and posterior region of the craniofacial complex. It is noteworthy that attribute Mc3 is among those of higher importance, since it indicates the sagittal position of the maxilla with respect to the skull, a characteristic that is independent and not directly related to the characteristics that allow the differentiation of biotypes.

With these four attributes, we repeat the performance evaluations and the statistical tests using the same 50 runs.

Table 3: Nemenyi test for single models (with 24 attributes) in terms of accuracy (%).

	cNB	gNB	fNB	dNB	cTAN	gTAN	fTAN	dTAN	cITCAN	gITCAN	fITCAN	SVM
gNB	1.00											
fNB	0.12	0.41										
dNB	0.00	0.03	0.99									
cTAN	0.13	0.44	1.00	0.99								
gTAN	0.91	0.99	0.98	0.44	0.99							
fTAN	0.00	0.00	0.00	0.00	0.00	0.00						
dTAN	0.00	0.00	0.39	0.97	0.36	0.01	0.01					
cITCAN	1.00	0.99	0.04	0.00	0.04	0.72	0.00	0.00				
gITCAN	1.00	1.00	0.33	0.01	0.35	0.99	0.00	0.00	0.99			
fITCAN	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.16	0.00	0.00		
dITCAN	0.00	0.00	0.17	0.85	0.16	0.00	0.02	1.00	0.00	0.00	0.36	
SVM	0.59	0.22	0.00	0.00	0.00	0.01	0.00	0.00	0.83	0.29	0.00	0.00

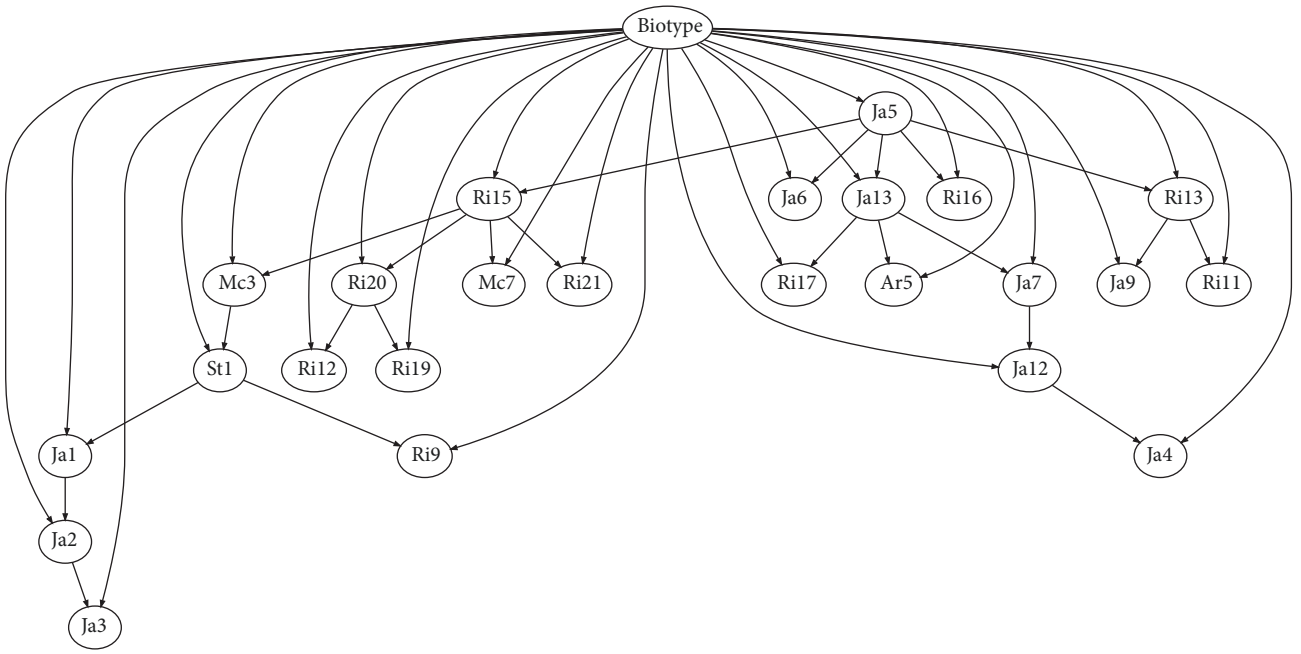

Figure 5: The cTAN classifier for the facial biotype dataset.

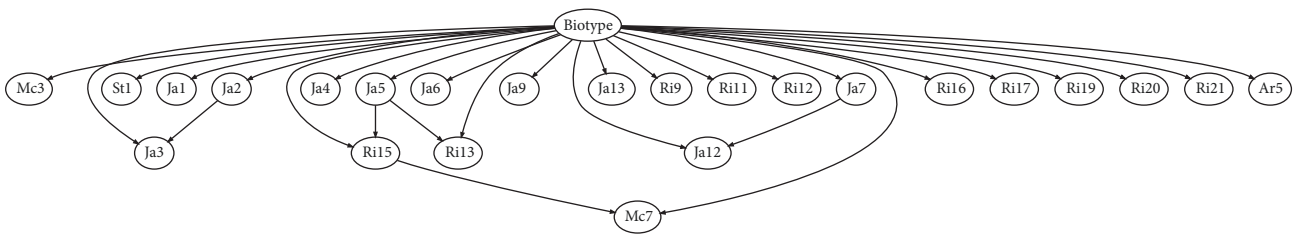

Figure 6: The cITCAN classifier for the facial biotype dataset.

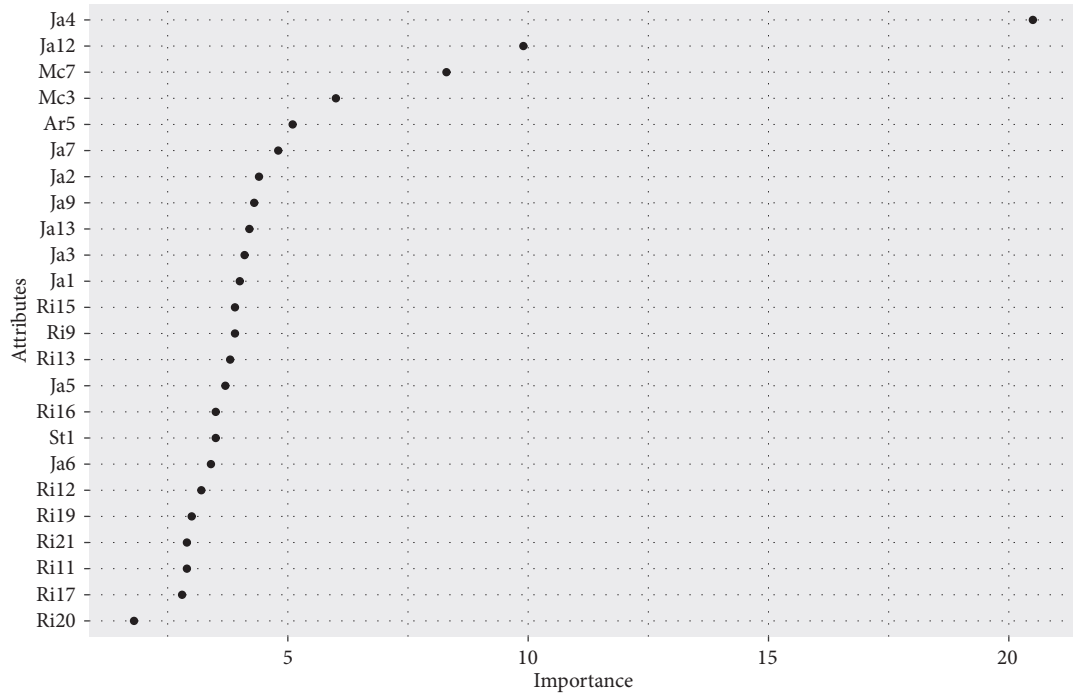

Figure 7: Attribute importance ranking based on the Gini importance measure.

Table 4: Performance measures for each classifier (with 4 attributes).

Algorithm	Accuracy %	Kappa
cNB	70.0±4.7	0.53±0.07
gNB	67.9±4.7	0.51±0.07
fNB	65.3±5.6	0.47±0.08
dNB	65.2±5.6	0.47±0.08
cTAN	68.2±5.6	0.51±0.09
gTAN	69.3±5.5	0.53±0.08
fTAN	49.6±6.9	0.22±0.10
dTAN	60.2±5.6	0.39±0.08
cITCAN	70.4±4.9	0.55±0.08
gITCAN	69.3±4.7	0.53±0.07
fITCAN	48.2±6.3	0.22±0.09
dITCAN	60.3±5.8	0.39±0.09
SVM	69.9±5.1	0.53±0.08

Table 5: The average ranks for all the algorithms (with 4 attributes).

Algorithm	Rank
cITCAN	3.96
cNB	4.14
SVM	4.22
gTAN	4.66
gITCAN	4.81
cTAN	5.47
gNB	5.61
fNB	7.01
dNB	7.08
dITCAN	9.52
dTAN	9.91
fTAN	12.15
fITCAN	12.46

The accuracy and kappa values are shown in Table 4. Overall, we see improvements in all the performance measures, in particular, the accuracy increased approximately by 10% in several classifiers. In relation to the kappa values, we notice that now cNB, gNB, fNB, dNB, cTAN, gTAN, cITCAN, gITCAN, and SVM are in the moderate interval of classification agreement with the true classes, with cITCAN obtaining the highest value. The worst accuracy and kappa value was obtained by the fITCAN classifier.

Following the same statistical tests as before, Table 5 shows the average rank for each algorithm. For the comparison of all the algorithms with the Friedman test, the χ_F^2 statistic is 372.66 and the p value is <2.2e-16, which rejects the null hypothesis that all the algorithms have the same performance.

Similar as before, a post hoc test was performed to evaluate the pairwise performance when all the algorithms are compared to each other. The Nemenyi test with $\alpha = 0.05$ was applied, and the results are presented in Table 6.

When comparing cITCAN with all the other classifiers, we notice that the null hypothesis cannot be rejected when compared to cNB, gNB, cTAN, gTAN, gITCAN, and SVM, respectively, since there are no statistically significant differences between them, whereas for our second ranked best classifier, cNB, we notice that the null hypothesis cannot be

Table 6: Nemenyi test for single models (with 4 attributes) in terms of accuracy (%).

	cNB	gNB	fNB	dNB	cTAN	gTAN	fTAN	dTAN	cITCAN	gITCAN	fITCAN	SVM
gNB	0.80											
fNB	0.01	0.85										
dNB	0.01	0.80	1.00									
cTAN	0.89	1.00	0.75	0.68								
gTAN	1.00	0.99	0.11	0.09	0.99							
fTAN	0.00	0.00	0.00	0.00	0.00	0.00						
dTAN	0.00	0.00	0.01	0.01	0.00	0.00	0.17					
cITCAN	1.00	0.65	0.01	0.00	0.77	0.99	0.00	0.00				
gITCAN	0.99	0.99	0.19	0.15	0.99	1.00	0.00	0.00	0.99			
fITCAN	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.06	0.00	0.00		
dITCAN	0.00	0.00	0.07	0.09	0.00	0.00	0.04	1.00	0.00	0.00	0.01	
SVM	1.00	0.86	0.02	0.01	0.92	1.00	0.00	0.00	1.00	0.99	0.00	0.00

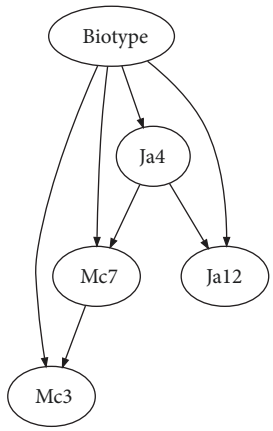

Figure 8: The cTAN classifier for the facial biotype dataset using only four attributes.

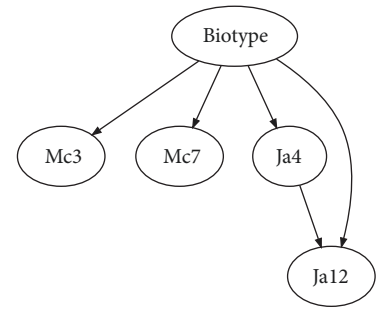

Figure 9: The cITCAN classifier for the facial biotype dataset using only four attributes.

rejected when compared to gNB, cTAN, gTAN, cITCAN, gITCAN, and SVM, respectively.

The resulting network structures for cTAN and cITCAN (for the simulations with only four attributes) are shown in Figures 8 and 9, respectively.

Overall, dropping irrelevant attributes contributed to the improvements of the classification performances of all the models.

5. Conclusion

We have presented adaptations for popular Bayesian network classifiers (naive Bayes and TAN) to handle continuous attributes. Additionally, we have proposed an incremental tree construction procedure for TAN (ITCAN) that may yield forest structures that model more effectively the posterior class distribution, thus, yielding competitive classification performances. We have applied these models to the facial biotype classification problem. Through classification performance measures and comparisons with other continuous Bayesian network classifiers approaches, we showed that these models can obtain competitive results when compared to a black-box model such as SVM. Also, the resulting network structures help to shed light on the probability relations amongst the attributes, which contributes to the understanding of their role in the classification process.

As an application in the context of medical informatics, trained Bayesian network classifiers for facial biotype classification can be used as an initial automatic screening process by orthodontists. Then, based on the posterior probability of the assigned class for each patient, define a threshold from which classifications with posterior probabilities below this threshold would require a manual validation by the orthodontist.

Appendix

Table 7 presents a list of the attributes and their description, used in this work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Table 7: A description of the attributes used in this work.

Attribute	Description
Mc3	Linear distance from point A to nasion perpendicular
Mc5	Mandibular length (Condylion to Gnathion)
Mc6	Maxillary length (Condylion to Point A)
Mc7	Lower anterior facial height (Anterior nasal spine to menton)
St1	SNA angle (Sella-Nasion-A)
Ja1	Saddle angle (Nasion-Sella-Articulare)
Ja2	Articular angle (Sella-Articulare-Gonion)
Ja3	Upper Gonial angle (Articulare-Gonion-Nasion)
Ja4	Lower Gonial angle (Nasion-Gonion-Menton)
Ja5	Anterior cranial base length (Sella to Nasion)
Ja6	Posterior cranial base length (Sella to Articulare)
Ja7	Ramus height (Articulate to Gonion)
Ja8	Mandibular corpus length (Gonion to Gnathion)
Ja9	Cranial base and Mandibular length ratio (Sella-Nasion/Gonion-Gnathion)
Ja10	Posterior facial height (Sella to Gonion)
Ja11	Anterior facial height (Nasion to Menton)
Ja12	Jarabak's ratio (Posterior facial height/Anterior facial height)
Ja13	Posterior cranial base and ramus height ratio (Sella-Articulare/Articulare-Gonion)
Ri9	Maxillary height angle (Nasion-Center of Face-A)
Ri10	Maxillary depth angle (Porion-Orbitale and Nasion-A)
Ri11	Palatal plane angle (Porion-Orbitale/anterior nasal spine-posterior nasal spine)
Ri12	Cranial deflection (Basion-Nasion/Porion-Orbitale)
Ri13	Anterior Cranial length (Center of Cranium to Nasion)
Ri15	Mandibular corpus axis (point Xi to point protuberance menti or Pm)
Ri16	Articular cavity position: Porion to Ptv (intersection of the distal outline of pterygomaxillary fissure perpendicular to the porion-orbitale plane)
Ri17	Mandibular ramus position (Porion-Orbitale/Center of Face-point Xi)
Ri18	Posterior height (Gonion to Center of Face)
Ri19	Condylar height
Ri20	Condylar neck length
Ri21	Symphysis length
Ar5	Nasolabial angle (Columella-Subnasale-upper lip)

Acknowledgments

The authors would like to thank Conicyt-Chile under grant Fondecyt 1180706 and Basal (CONICYT)-CMM, for financially supporting this research.

References

[1] R. S. Nanda, "The contributions of craniofacial growth to clinical orthodontics," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 117, no. 5, pp. 553–555, 2000.

[2] R. Mangla, V. Dua, M. Khanna, N. Singh, and P. Padmanabhan, "Evaluation of mandibular morphology in different facial types," *Contemporary Clinical Dentistry*, vol. 2, no. 3, p. 200, 2011.

[3] E. De Novaes Benedicto, S. A. Kairalla, G. M. S. Oliveira, L. R. M. Junior, H. D. Rosário, and L. R. Paranhos, "Determination of vertical characteristics with different cephalometric measurements," *European Journal of Dentistry*, vol. 10, no. 1, pp. 116–120, 2016.

[4] R. M. Ricketts, "Planning treatment on the basis of the facial pattern and an estimate of its growth," *The Angle Orthodontist*, vol. 27, pp. 14–37, 1957.

[5] S. G. F. Gomes, W. Custodio, F. Faot, A. A. Del Bel Cury, and R. C. M. R. Garcia, "Masticatory features, EMG activity and muscle effort of subjects with different facial patterns," *Journal of Oral Rehabilitation*, vol. 37, no. 11, pp. 813–819, 2010.

[6] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[8] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin, "Feedforward neural networks with random weights," in *Proceedings of the 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*, pp. 1–4, The Hague, Netherlands, 1992.

[9] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.

[10] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[11] E. Cheng, J. Chen, J. Yang et al., "Automatic Dent-landmark detection in 3-D CBCT dental volumes," in *Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2011*, pp. 6204–6207, USA, September 2011.

[12] X. Wang, B. Cai, Y. Cao et al., "Objective method for evaluating orthodontic treatment from the lay perspective: An eye-tracking study," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 150, no. 4, pp. 601–610, 2016.

[13] A. Lakkshmanan, A. A. Shri, and E. Aruna, "Pattern classification for finding facial growth abnormalities," in *Proceedings of the 2013 4th IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2013*, India, December 2013.

[14] S. Murata, C. Lee, C. Tanikawa, and S. Date, "Towards a fully automated diagnostic system for orthodontic treatment in dentistry," in *Proceedings of the 13th IEEE International Conference on eScience, eScience 2017*, pp. 1–8, New Zealand, October 2017.

[15] J. R. Quinland, "Learning efficient classification procedures and their applications to chess end games," in *Machine Learning an Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds., pp. 463–482, Morgan Kaufmann, Los Altos, Calif, USA, 1983.

[16] R. S. Michalski, "A theory and methodology of inductive learning," in *Readings in Machine Learning*, J. W. Shavlik and T.

G. Dietterich, Eds., pp. 70–95, Morgan Kaufmann, San Mateo, Calif, USA, 1990.

[17] D. T. Pham and M. S. Aksoy, "RULES: A simple rule extraction system," *Expert Systems with Applications*, vol. 8, no. 1, pp. 59–65, 1995.

[18] D. T. Pham and S. S. Dimov, "An efficient algorithm for automatic knowledge acquisition," *Pattern Recognition*, vol. 30, no. 7, pp. 1137–1143, 1997.

[19] J. R. Quinlan, *C4.5: programs for machine learning*, Morgan Kaufmann, 1993.

[20] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, Boston, Mass, USA, 1988.

[21] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[22] V. K. Mago, B. Prasad, A. Bhatia, and A. Mago, "A decision making system for the treatment of dental caries," *Studies in Fuzziness and Soft Computing*, vol. 230, pp. 231–242, 2008.

[23] A.-S. Mesaros, S. Sava, D. Mitrea et al., "In vitro assessment of tooth color changes due to orthodontic treatment using knowledge discovery methods," *Journal of Adhesion Science and Technology*, vol. 29, no. 20, pp. 2256–2279, 2015.

[24] M. Nieri, A. Crescini, R. Rotundo, T. Baccetti, and P. Cortellini, "Factors affecting the clinical approach to impacted maxillary canines: A Bayesian network analysis," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 137, no. 6, pp. 755–762, 2010.

[25] J. Buti, M. Baccini, M. Nieri, M. La Marca, and G. P. Pini-Prato, "Bayesian network meta-analysis of root coverage procedures: Ranking efficacy and identification of best treatment," *Journal of Clinical Periodontology*, vol. 40, no. 4, pp. 372–386, 2013.

[26] M. Merli, M. Moscatelli, G. Mariotti, U. Pagliaro, F. Bernardelli, and M. Nieri, "A minimally invasive technique for lateral maxillary sinus floor elevation: A Bayesian network study," *Clinical Oral Implants Research*, vol. 27, no. 3, pp. 273–281, 2016.

[27] B. Thanathornwong, "Bayesian-based decision support system for assessing the needs for orthodontic treatment," *Health Informatics Journal*, vol. 24, no. 1, pp. 22–28, 2018.

[28] D. M. Chickering, *Learning Bayesian Networks is NP-Complete*, Springer, New York, Ny, USA, 1996.

[29] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.

[30] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: the combination of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, pp. 197–243, 1995.

[31] D. Heckerman, "A tutorial on learning with bayesian networks," Tech. Rep. MSR-TR-95-06, Microsoft Research, 1995.

[32] R. E. Neapolitan, *Learning Bayesian networks*, Pearson Prentice Hall, Upper Saddle River, NJ, USA, 2004.

[33] R. O. Duda and P. E. Hart, *Pattern Classification And Scene Analysis*, John Wiley & Sons, New York, Ny, USA, 1973.

[34] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[35] D. Margaritis and S. Thrun, "Bayesian network induction via local neighborhoods," in *Advances in Neural formation Processing Systems 12*, S. A. Solla, T. K. Leen, and K. Müller, Eds., pp. 505–511, MIT Press, 2000.

[36] G. M. Provan and M. Singh, "Learning Bayesian Networks Using Feature Selection," in *Learning from Data*, vol. 112 of *Lecture Notes in Statistics*, pp. 291–300, Springer, New York, NY, USA, 1996.

[37] M. J. Pazzani, *Constructive Induction of Cartesian Product Attributes*, Springer US, Boston, Mass, USA, 1998.

[38] M. Sahami, "Learning limited dependence bayesian classifiers," in *Proceedings of the In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 335–338, 1996.

[39] G. A. Ruz and D. T. Pham, "Building Bayesian network classifiers through a Bayesian complexity monitoring system," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 223, no. 3, pp. 743–755, 2009.

[40] C. Bielza and P. Larrañaga, "Discrete bayesian network classifiers: A survey," *ACM Computing Surveys*, vol. 47, no. 1, 2014.

[41] P. Araya-Díaz, G. A. Ruz, and H. M. Palomino, "Discovering craniofacial patterns using multivariate cephalometric data for treatment decision making in orthodontics," *International Journal of Morphology*, vol. 31, no. 3, pp. 1109–1115, 2013.

[42] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2016.

[43] J. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, pp. 48–50, 1956.

[44] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Labs Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.

[45] G. Sales and C. Romualdi, "Parmigene—a parallel R package for mutual information estimation and gene network reconstruction," *Bioinformatics*, vol. 27, no. 13, pp. 1876-1877, 2011.

[46] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, Article ID 066138, 2004.

[47] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. 1695, 2006.

[48] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, Springer, New York, NY, USA, 2002.

[49] D. Nychka, R. Furrer, J. Paige, and S. Sain, "Fields: Tools for spatial data," R package version 9.0, 2015.

[50] P. J. Lucas, "Restricted BAYesian network structure learning," in *Advances in BAYesian networks*, vol. 146 of *Stud. Fuzziness Soft Comput.*, pp. 217–234, Springer, Berlin, 2004.

[51] A. Pérez, P. Larrañaga, and I. Inza, "Supervised classification with conditional Gaussian networks: increasing the structure complexity from naive Bayes," *International Journal of Approximate Reasoning*, vol. 43, no. 1, pp. 1–25, 2006.

[52] A. Pérez, P. Larrañaga, and I. Inza, "Bayesian classifiers based on kernel density estimation: flexible classifiers," *International Journal of Approximate Reasoning*, vol. 50, no. 2, pp. 341–362, 2009.

[53] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, London, UK, 1986.

[54] M. Scutari, "Learning Bayesian networks with the bnlearn R Package," *Journal of Statistical Software* , vol. 35, no. 3, pp. 1–22, 2010.

[55] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch, "e1071: Misc Functions of the Department of Statistics,

Probability Theory Group (Formerly: E1071), TU Wien," R package version 1.6-8, 2017.

[56] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[57] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[58] A. Liaw and M. Wiener, "Classification and regression by randomForest," *The R Journal*, vol. 2, no. 3, pp. 18–22, 2002.