Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/110617

A Decision Tree Approach for the Musical Genres Classification

Glaucia M. Bressan[1,*], *Beatriz C. F. de Azevedo*[2] *and Elisangela Ap. S. Lizzi*[1]

[1] Department of Mathematics, Federal University of Technology of Parana, Cornelio Procopio - PR, Brazil.
[2] Department of Electrical Engineering, Federal University of Technology of Parana, Cornelio Procopio - PR, Brazil.

Received: 2 Sep. 2017, Revised: 2 Oct. 2017, Accepted: 6 Oct. 2017
Published online: 1 Nov. 2017

Abstract: The interest in the music classification has increased due to its wide applicability and discoveries obtained from researches. However, efficient methods for systemic organization of digital libraries are required, since users need to classify the available music files. When an automatic classification is desired, the extraction of input attributes and an efficient system, able to process them, are needed. In this context, the use of decision trees as a tool to predict musical genres classes allows the monitoring of the ramification, since nodes and branches of the tree can be accessed in this process. Decision tree is a technique very useful in data mining to extract information of a data set, normally using a TDIDT (Top-Down Induction Decision Tree) algorithm. Therefore, the goal of this paper is to propose an automatic classification method for Latin musical genres, by applying decision tree approach. The real database used is named *Latin Music Database* [20]. Two algorithms are executed: CART (Classification and Regression Tree) [2] and C4.5 [18], which have constructive criteria distinguished. The obtained results are compared and discussed in order to evaluate the classification performance.

Keywords: Latin musical genres, automatic classification, decision tree, machine learning, data mining.

1 Introduction

Music is frequently present in our lives. In recent decades, there was an observable growth of music information, especially in online media. Automatic musical genre classification is a fundamental component of music information retrieval systems and has been enjoying a growing amount of attention with the emergence of digital music on the Internet [12]. More than leisure and entertainment, music is able to manipulate reactions and decisions; presenting multiple features. As an example, we can mention the music therapy, which is applied in patients with mental disorders, and it is discussed in literature [7,11]. For the reasons described before, commercial and financial interests has been increasing significantly, as well as the academic research interest, specially in music information classification and recuperation study areas [9,22,26].

The automatic extraction of music information has had considerable importance as a way to structure and organize large number of digital music files available on the Web [25]. In the literature and in the digital music libraries, the musical genre classification is the most common way to catalog this type of media. Although musical genres do not have strict rules and well defined boundaries, this category is more easily quantified than other musical parameters. For this reason the musical genres classification has had more attention in the scientific community [1,6,12,26].

It is expected that music from the same genre share similarities among themselves, which makes them different from other genres [1]. However, the expansion of mediatic resources has increasing the number of genres and their mergers. According to Li and Ogihara (2005), as the music industry grows, the boundaries between musical genres become blurred, making the process of classification more difficult. Although considerable research has been proposed in the literature, there is no a general solution for the automatic classification task of musical genres and little has been done on hierarchical classification [12]. For that, the search for a solution using mathematical and statistics modeling, applied to data mining, is relevant to propose solutions to classification problems.

In this context, the goal of this paper is to propose an automatic classification method for musical genres, by

* Corresponding author e-mail: glauciabressan@utfpr.edu.br

© 2017 NSP
Natural Sciences Publishing Cor.

decision tree approach. Decision trees are prediction models that allow researchers to examine the data relationships by accessing the nodes and the branches of the tree. Two algorithms for construction of decision trees are executed: CART (Classification and Regression Tree - [2]) and C4.5 [18] algorithms. Both structure decision trees from a data sample, however they have constructive criteria distinguished. For that, the outputs of the two algorithms can be compared in order to find the best solution to the classification problem. The database used is named *Latin Music Database*, from [20], which is composed by real numerical features that describes Latin genres. The CART algorithm generates binary decision trees and C4.5 algorithm can generate more than two splits. In order to compare the results obtained from the classification trees, both of them are considered as binary trees.

This paper is organized as follows: The section 2 describes some papers that apply decision trees algorithm to classify music or something related to the subject. The section 3 presents the decision tree technique. The methods and algorithms used in this work are presented in the section 4. And the database in the section 5. The section 6 presents the results. Finally, section 7 comments on the conclusion obtained from the results of classification.

2 Related Works

Decision tree is very used in data mining, since it is frequently applied in areas such as finance, marketing, engineering and medicine [19]. However, the employment of decision tree as a tool to predict musical genres classes is little used in the literature. An advantage of using decision tree is the fact that not only numerical parameters are shown, but also it provides for the researcher the possibility to follow the data classification through the graphical analysis of the tree structure. This characteristic upgrade the investigation about the behavior of the attributes involved in the classification task.

Considering authors that investigated music, sounds or something related to the subject, Castán et. al (2010) propose a system able to classify broadcast radio data considering music, voice or both. The difficulty in working with this kind of data is to create a robust model for the identification of music signal. The solution proposed combines six features using C4.5 algorithm, which build a binary decision tree to provide the minimum classification error. The experiment result shows the decision tree method improve the results of the individual features what highlights the complementarity among them, besides this obtains an improvement of more than 10% over the most discriminative feature [3].

Another interesting approach is described by Jensen and Arnspang (1999), who present a method based on binary trees created with average entropy. The main aim of this method is comprehend which timbre attributes split the sounds of the instrument into classes. In this scenario, classification trees show positive results providing more information about the importance of the timbre attributes in the identification of instruments. Furthermore, they could be used to classify new, unknown sounds, and help understand which timbre attributes are pertinent in the identification of musical sounds [8].

Yuan et al. (2002) apply decision tree to classify videos as different genres and produce a set of decision rules. According to the authors, videos could be categorized into different genre using attributes, due to videos belonging to a specific genre have similar features that differ from the others, alike music genres. As result decision trees present nearly 75% of accuracy and for this reason they might be considered promising to classification task [27].

Norowi et al. (2005) present the results of five different experiments used to classify eight traditional Malay music genres. The authors conduct the study varying some conditions as data set size, start point of the music, track length, numbers of Cross-Validation and considering two classifier, OneR and J48 (decision tree algorithm). This last experiment demonstrated a superior accuracy of J48 compared to OneR classifier. They conclude that some classifiers are suitable in performing a classification problem while other may not [13].

Considering the state of the art, the main contribution of this paper is to automate the classification task of Latin musical genres, using decision trees, an approach little explored in literature. Compared to other advanced machine learning models, as Black Box Methods (Neural Networks and Support Vector Machines), the decision trees generally perform nearly as well but are much easier to understand and deploy [10].

3 Decision Trees

Decision tree is a technique very useful in data mining to extract information of a data set. Normally, this extract information process is done by a TDIDT (Top-Down Induction Decision Tree) algorithm, described in Figure 1 [19], which induces a tree structure by splitting data into subgroups more and more uniforms based on divide and conquer method [10]. This split process stops when the subset contains just one class or when no more improvement is possible.

The split process also can be interrupted by a stopped criterion pre-determined. Mathematically, a split can be represented by a test function $t : X \longrightarrow R_t$ that maps instances into split outcomes so a separate outgoing branch is associated with each possible outcomes of a node's split [4]. Decision tree models can be used in classification tasks (Classification Tree), in order to classify objects or instances in tagged classes; or in regression tasks (Regression Tree), when the outcome is describe by a numeric or categoric value.

© 2017 NSP
Natural Sciences Publishing Cor.

A decision tree consists of a data set, partitioned into groups known as *nodes*. The top node is called *root node*, which is selected using some attribute selection measures, like described in Section 4.

```
TreeGrowing (S,A,y,SplitCriterion,StoppingCriterion)
Where:
S - Training Set
A - Input Feature Set
y - Target Feature
SplitCriterion - the method for evaluating a certain split
StoppingCriterion - the criteria to stop the growing process
Create a new tree T with a single root node.
IF StoppingCriterion(S) THEN
    Mark T as a leaf with the most
    common value of y in S as a label.
ELSE
    ∀a_i ∈ A find a that obtain the best SplitCriterion(a_i, S).
    Label t with a
    FOR each outcome v_i of a:
        Set Subtree_i= TreeGrowing (σ_{a=v_i} S, A, y).
        Connect the root node of t_T to Subtree_i with
                an edge that is labelled as v_i
    END FOR
END IF
RETURN TreePruning (S,T,y)
TreePruning (S,T,y)
Where:
S - Training Set
y - Target Feature
T - The tree to be pruned
DO
    Select a node t in T such that pruning it
        maximally improve some evaluation criteria
    IF t ≠ ∅ THEN T = pruned(T,t)
UNTIL t = ∅
RETURN T
```

Fig. 1: Top-Down Algorithm for Decision Tree Induction. Source: [19]

Under the root node are the internal nodes, originating from the division of the data set; they constitute the tree branches. At the end of each branch is the terminal node, designed leaves, which represent the most appropriated class for the rule. One rule is composed by each terminal node, plus the internal nodes that belongs to one specific branch and the root node. The rules are describe by a IF-THEN model ("IF attribute w is $y1$ AND attribute x is $y2$ THEN the class is z"). These rules are used to classify unknown data at the inference process.

A simple decision tree structure is illustrated in Figure 2.

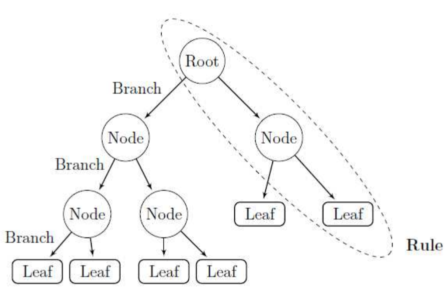


Fig. 2: Example of decision tree structure.

4 Methods and Algorithms

In this section, algorithms employed to classify the Latin Music Database are described. There are many different ways to construct a decision tree. CART (Classification and Regression Tree) and ID3 (Iterative Dichotomiser) are very common algorithm to induce trees. These algorithms present other more recent versions, obtained from an evolution or an improvement of them, with different construction criteria. C4.5 algorithm is an evolution of ID3 algorithm.

CART and C4.5 algorithms construct the tree in two phases: growing and pruning, while other ones, as ID3, just execute the growing process. The growing is a recursive process that establishes the structure of the decision tree according to some splitting criterion, such Information Gain, Gini Index, Twoing, and Gain Ratio, which definitions are described in this section. ID3 algorithm uses Information Gain which is an impurity-based criteria that uses entropy measure as the impurity measure [17,19]. This estimator is closely related to the Maximum Likelihood Estimator, which is a popular statistical method used to make inferences about parameters of the underlying probability distribution from a given data set [19].

CART algorithm is characterized by the fact that it constructs binary trees, namely each internal node has exactly two outgoing edges [19]. The splitting criteria is the Twoing criteria, that search for two classes that will make up together more then 50% of the data; Twoing splitting rule allows us to build more balanced trees but this algorithm works slower than Gini rule [23].

C4.5 algorithm is an ID3 evolution and it uses Gain Ratio as splitting criteria. The element with highest Gain Ratio is taken as the root node and data set is split based on the root element values [14]. The Information Gain is calculated for all the sub-nodes and the process is repeated until the prediction is completed. CART and C4.5 algorithms use a technique of exhaustive research to define the thresholds to be used in the nodes to divide the continuous attributes.

The second phase, pruning, is responsible to reduce the complexity of the tree. It means reducing size of the tree that are too large and deep. The problem of noise and overfitting reduces the efficiency and accuracy of data. The overfitting happens when the tree lost the ability of generalizing to instances not present during the training process. By increasing the number of nodes, the training error usually decreases while at some point the generalization error becomes worse, (Rokach and Maimon (2008)).

Pruning methods typically use statistical measures to remove the least reliable branches [15]. As a consequence it optimize the computational operation, eliminate overfitting, and improve the classification of unknown data. There are two types of pruning, pre-pruning and post-pruning. The first one, while building the decision tree keep on checking whether tree is overfitting based on

© 2017 NSP
Natural Sciences Publishing Cor.

different measures like Laplace Error and Minimum Description Length [15]. And in the second one, the tree is built completely, after this, the branches and levels are reduce using Error-Based Pruning [19], for example.

Other possibility to avoid large tree is establish stopping criteria. However, Rokach et. al (2008) mention tight stopping criteria tends to create small and underfitted tree, whereas loose criteria tends to generate large and overfitted decision trees [19]. To solve this problem Breiman et. al (1984) developed a pruning methodology based on a loose stopping criterion and allowing the decision tree to overfit the training set. Then the overfitted tree is cut back into a smaller tree by removing sub-branches that are not contributing to the generalization accuracy [2, 19].

The pruning techniques varies according to the decision tree algorithm, Cost-Complexity Pruning, Reduced Error Pruning and Minimum Error Pruning are some examples that can be used to prune trees.

4.1 CART Algorithm

CART stands for classification (categoric attributes) and regression (continuous attributes) trees. The splits are selected using the Twoing criteria and the pruning process is the Cost-Complexity. An important feature of CART is its ability to generate regression trees. The leafs predict a real number and not a class. In case of regression the algorithm looks for splits that minimize the prediction squared error [19]. The prediction in each leaf is based on the weighted mean for node.

The CART can construct classification or regression trees in two phases: growing and pruning.

4.1.1 Growing CART tree

First of all, it is important to introduce the Gini Index. Gini Index is an impurity-based criteria that measures the divergences between the probability distributions of the target attributes values. It is defined as (1), given a training set S and a target attribute y [19]. The σ denotes selection of tuples, for example, $\sigma_{y=c_j}S$ denotes the selection of instances of the attribute y that belong to c_j class, given the data set S.

$$Gini(y,S) = 1 - \sum_{c_j \in dom(y)} \left(\frac{\sigma_{y=c_j}S}{|S|}\right)^2. \tag{1}$$

Consequently the evaluation criterion for selecting the attribute a_i is defined as (2)

$$GiniGain(a_i,S) = Gini(y,S) +$$

$$- \sum_{v_{i,j} \in dom(a_i)} \left(\frac{\sigma_{a_i=v_{i,j}}S}{|S|}\right) \times Gini(y, \sigma_{a_i=v_{i,j}}S). \tag{2}$$

The Gini Index may encounter problems when the domain of the target attribute is relatively wide. In such cases, it is possible to employ binary criterion called Twoing. This criterion is defined as (3),where $dom(a_i)$ and $dom(y)$ represent the domain of the attribute a_i and of the target attribute, respectively.

$$Twoing(a_i, dom_1(a_i), dom_2(a_i), S) = 0.25\times$$

$$\times \frac{\sigma_{a_i \in dom_1(a_i)}S}{|S|} \times \frac{\sigma_{a_i \in dom_2(a_i)}S}{|S|} \times$$

$$\times \left(\sum_{c_i \in dom(y)} \left| \frac{\sigma_{a_i \in dom_1(a_i) AND y=c_i}S}{|\sigma_{a_i \in dom_1(a_i)}S|} - \frac{\sigma_{a_i \in dom_2(a_i) AND y=c_i}S}{|\sigma_{a_i \in dom_2(a_i)}S|} \right| \right)^2 \tag{3}$$

When the target attribute is binary the Gini and Twoing criteria are equivalent. For multi-class problems the Twoing criteria prefers attributes with evenly divided splits [19].

4.1.2 Pruning CART Tree

The tree obtained by CART algorithm is pruned using Cost-Complexity (CC), which proceeds in two stages. In the first stage, a sequence of trees $T_0, T_1, \ldots, T_k$ is built on the training data where T_0 is the original tree before pruning and T_k is the root tree [19].

In the second stage, one of these trees is chosen as the pruned tree, based on its generalization error estimation. Tree T_{i+1} is obtained by replacing one or more of the sub-trees in the predecessor tree T_i with suitable leaves. The pruned sub-trees are those that obtain the lowest increase in apparent error rate per pruned leaf, as (4).

$$\alpha = \frac{\varepsilon(pruned(T,t),S) - \varepsilon(T,S)}{|leaves(T)| - |leaves(pruned(T,t))|} \tag{4}$$

where:

I) $\varepsilon(T,S)$ indicates the error rate of tree T over the sample S;

II) $|leaves(T)|$ indicates the number of leaves in T;

III) $pruned(T,t)$ indicates the tree obtained by replacing the node t in T with a suitable leaf.

In the second stage, the generalization error of each pruned tree is estimated and the best one is then selected.

4.2 C4.5 Algorithm

C4.5 was developed by Quinlan (1986) [16], which is an evolution of ID3. The main contributions of C4.5 algorithm, compared to ID3, are: it can run categorical or continuous attributes; it can works with unknown values; it uses Gain Ratio as splitting criteria, which generates

© 2017 NSP
Natural Sciences Publishing Cor.

more robust trees than Information Gain; it presents a post-pruning procedure. ID3 does not apply any pruning procedure nor does it handle numeric attributes or missing data.

Although there is already the C5.0 algorithm, the C4.5 is the most frequently used in literature, for having obtained excellent results in classification problems and for having the source code available; while the C5.0 is a commercial software. The procedure of C4.5 algorithm is described below, in growing and pruning phases.

4.2.1 Growing C4.5 tree

C4.5 algorithm uses Gain Ratio as splitting criteria. When the number of instances to be split is below a certain threshold, the splitting stopes. The algorithm can handle numeric attributes. In order to introduce the Gain Ratio, we must present the Information Gain and Entropy concepts.

The Entropy of a sample of data indicates how mixed the class values are. It consists of a measuring purity. The minimum value of 0 indicates that the sample is homogeneous and 1 indicates the disorder [10]. The definition of entropy can be expressed as (5).

$$Entropy(y,S) = \sum_{c_j \in dom(y)} -\frac{|\sigma_{y=c_j}S|}{|S|} \times log_2 \frac{|\sigma_{y=c_j}S|}{|S|} \quad (5)$$

Information Gain uses the Entropy measure as purity measure, as (6). Using (6), the algorithm decides which feature must split upon and uses Entropy to calculate the change in homogeneity, resulting from a split on each possible feature.

$$InfoGain(a_i,S) = Entropy(y,S) +$$

$$-\sum_{v_{i,j} \in dom(a_i)} \frac{|\sigma_{a_i=v_{i,j}}S|}{|S|} \times Entropy(y, \sigma_{a_i=v_{i,j}}S) \quad (6)$$

Finally, the Gain Ratio can be express by (7). According to Quinlan (1993), it normalizes the Information Gain.

$$GainRatio(a_i,S) = \frac{InfoGain(a_i,S)}{Entropy(a_i,S)} \quad (7)$$

The ratio may tend to favor attributes for which the denominator is very small. First the Information Gain is calculated for all attributes. Then, the attribute that has obtained the best Gain Ratio is selected, as a consequence of considering only attributes that have performed at least as well as the average Information Gain [19]. On other words, the Gain Ratio criterion selects a test to maximize the ratio (7), subject to the constraint that the Information Gain must be large - at least as great as the average gain over all tests examined.

4.2.2 Pruning C4.5 Tree

Some methods of pruning process discard part of the tree called subtrees, and replace them with leaves. The strategy used by C4.5 named Error-Based Pruning (EBP) it is an evolution of other method called Pessimistic Pruning. C4.5 equates the predicted error rate at a leaf with this upper limit, on the argument that the tree has been constructed to minimize the observed error rate [18]. Rokach and Maimon (2008) describe the EBP as the following way: as in pessimistic pruning, the error rate is estimated using the upper bound of the statistical confidence interval for proportions, as (8).

$$\varepsilon_{UB}(T,S) = \varepsilon(T,S) + Z_\alpha \times \sqrt{\frac{\varepsilon(T,S) \times (1-\varepsilon(T,S))}{|S|}} \quad (8)$$

where $\varepsilon(T,S)$ denotes the misclassification rate of the tree T on the training set S; Z is the inverse of the standard normal cumulative distribution; and α is the desired significance level.

Let $subtree(T,t)$ denote the subtree rooted by the node t. Let $maxchild(T,t)$ denote the most frequent child node of t (namely most of the instances in S reach this particular child) and let S_t denote all instances in S that reach the node t. The procedure traverses bottom-up all nodes and compares the following values:

I) $\varepsilon_{UB}(subtree(T,t),S_t)$

II) $\varepsilon_{UB}(pruned(subtree(T,t),t),S_t)$

III) $\varepsilon_{UB}(subtree(T,maxchild(T,t)),Smaxchild(T,t))$

According to the lowest value, the procedure either leaves the tree as is; prune the node t; or replaces the node t with the subtree rooted by $maxchild(T,t)$ [19].

An alternative of pruning method is called *Reduced Error Pruning* (REP), which is a post-pruning method, proposed by Quinlan (1987), that uses a hold out set for error estimates. While traversing over the internal nodes from the bottom to the top, the procedure checks each internal node to determine whether replacing it with the most frequent class does not reduce the trees accuracy. So, the node is pruned if accuracy is not reduced and the procedure continues until any further pruning would decrease the accuracy.

4.3 Evaluation of Classification Trees

Evaluating the performance of a classification tree is a fundamental aspect of machine learning. The decision tree inducer receives a training set as input and constructs a classification tree that can classify an unseen instance. In this work, Cross-Validation, Receiver Operating Characteristic and Confusion Matrix are applied as efficient indicators for assessing the quality of the analysis results.

In n-fold *Cross-Validation*, the data is randomly split into n mutually exclusive subsets of approximately equal

© 2017 NSP
Natural Sciences Publishing Cor.

size. An inducer is trained and tested n times; each time it is tested on one of the k folds and trained using the remaining $n-1$ folds. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set $n-1$ times. The variance of the resulting estimate is reduced as n is increased. Therefore, the basic idea is that some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on "new" data.

The *Confusion Matrix* is used as an indication of the properties of a classification rule. It contains the number of elements that have been correctly or incorrectly classified for each class. The main diagonal presents the number of observations that have been correctly classified for each class. The off-diagonal elements present the number of observations that have been incorrectly classified [19].

The Confusion Matrix $M(C_i,C_j)$, for $i,j=1,...,k$, classes provides an effective measure of the classification model performance, since it shows the number of elements correctly classified and predict, for each class. Let T be a set of examples and h the hypothesis. The Confusion Matrix of h is given by:

$$M(C_i,C_j) = \sum_{\forall(x,y)\in T:y=c_i} \parallel h(x) = C_j \parallel$$

where C_i is the true class and the C_j is the predict class. Table 1 illustrates the Confusion Matrix.

Table 1: Confusion Matrix

Class	Predict C_1	Predict C_2	$\ldots$	Predict C_k
True C_1	$M(C_1,C_1)$	$M(C_1,C_2)$	$\ldots$	$M(C_1,C_k)$
True C_2	$M(C_2,C_1)$	$M(C_2,C_2)$	$\ldots$	$M(C_2,C_k)$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
True C_k	$M(C_k,C_1)$	$M(C_k,C_2)$	$\ldots$	$M(C_k,C_k)$

Receiver Operator Characteristic (ROC) curve, or ROC curve, is commonly used to present results for decision problems in machine learning. It examines the tradeoff between the detection of true positives, while avoiding the false positives. Curves are defined on a plot with the proportion of true positives on the vertical axis, and the proportion of false positives on the horizontal axis. These values are equivalent to *sensitivity* and *(1 specificity)*, respectively. The closer the curve is to the perfect classifier, the better it is at identifying positive values. This can be measured using a statistic known as the *area under the ROC curve*, which measures the total area under the ROC curve. The area ranges from 0.5 (for a classifier with no predictive value), to 1.0 (for a perfect classifier).

The points comprising ROC curves indicate the true positive rate at varying false positive thresholds. To create the curves, a classifier's predictions are sorted by the model's estimated probability of the positive class, with the largest values first. Beginning at the origin, each prediction's impact on the true positive rate and false positive rate will result in a curve tracing vertically (for a correct prediction), or horizontally (for an incorrect prediction) [10].

5 Latin Music Database

In this paper, we used a set of the available music in *Latin Music Database*, from [20], a musical recordings database from the ten different genres: Tango, Salsa, Forró, Axé, Bachata, Bolero, Merengue, Gaúcha, Sertanejo and Pagode. The framework from [24] was employed for feature extraction.

The features employed in this paper comprise short-time Fourier transform, Mel frequency cepstral coefficients (MFCC), beat and pitch related features, inter-onset interval histogram coefficients, rhythm histograms and statistical spectrum descriptors. More details can be seen in [21].

The features can be split into three groups: the beat-related features (features 1 to 6) include the relative amplitudes and the beats per minute. Timbral texture features (features 7 to 25) account for the means and variance of the spectral centroid, rolloff, flux, the time zero domain crossings, the first five MFCCs and low energy. Pitch-related features (features 26 to 30) include the maximum periods and amplitudes of the pitch peaks in the pitch histograms.

Each row of the input matrix is called "instance", it contains the numerical information about the features and it corresponds to one genre. The final feature vector is outlined in Table 2.

© 2017 NSP
Natural Sciences Publishing Cor.

Table 2: Description of the feature vector

Feature	Description
1	Relative amplitude of the first histogram peak
2	Relative amplitude of the second histogram peak
3	Ratio between the amplitudes of the second peak and the first peak
4	Period of the first peak in bpm
5	Period of the second peak in bpm
6	Overall histogram sum (beat strength)
7	Spectral centroid mean
8	Spectral rolloff mean
9	Spectral flow mean
10	Zero crossing rate mean
11	Standard deviation for spectral centroid
12	Standard deviation for spectral rolloff
13	Standard deviation for spectral flow
14	Standard deviation for zero crossing rate
15	Low energy
16	First MCFF mean
17	Second MCFF mean
18	Third MCFF mean
19	Fourth MCFF mean
20	Fifth MCFF mean
21	Standard deviation for first MFCC
22	Standard deviation for second MFCC
23	Standard deviation for third MFCC
24	Standard deviation for fourth MFCC
25	Standard deviation for fifth MFCC
26	The overall sum of the histogram (pitch strength)
27	Period of the maximum peak of the unfolded histogram
28	Range of the maximum peak of the folded histogram
29	Period of the maximum peak of the folded histogram
30	Pitch interval between the two most prominent peaks of the folded histogram

6 Results

In this paper, for the classification task, we have chosen CART and C4.5 algorithms due to both are available in free software and have different construction criteria. We use the WEKA software (Waikato Environment of Knowledge Analysis - http://www.cs.waikato.ac.nz), developed by the University of Waikato, New Zealand. In this software, CART is named SimpleCart and C4.5 is named as J48. This fact makes possible to compare the performance of different algorithms, since they are implemented using the same software. Therefore, we avoid the comparison of results of algorithms that can present different performances if they are implemented on different platforms.

The Latin Music Database is organized in a matrix, where each column represents the 30 input attributes described in Table 2. Thus, the full database is built by 31 columns, including the output - musical genres - in the last column. There are 3000 instances (rows of the matrix database), being 10 instances of each one of the ten genres. Firstly, in order to execute the decision trees algorithms, the numerical input attributes were categorized according to tercile; then, each input attribute is divided in 3 categories: low, medium and high, with 1000 values in each category. Then, database is partitioned in *training set* and *test set*. Training set is composed by 80% of full database (2400 instances) and test set is composed by 20% (600 instances). In other words, there are 300 instances of each genre. These instances are partitioned in training (80%) and test (20%) sets. Musical genres are the outputs of the decision trees.

In order to generate the decision tree model to classify the database, classification results were obtained by considering the algorithms described in Section 4 and different measures of validation. The results of musical genres classification is described in Tables 3, 4, 5 and 6.

Table 3: Numerical Data - Algorithm CART

Method	Prune	Leaves	Size	Accuracy
Training Set	CC	67	133	73.20%
Cross Validation	CC	67	133	63.33%
Test Set	CC	58	115	62.83%
Training Set	Unpruned	341	681	91.93%
Cross Validation	Unpruned	341	681	62.97%
Test Set	Unpruned	293	585	60.17%

Table 4: Numerical Data - Algorithm C4.5

Method	Prune	Leaves	Size	Accuracy
Training Set	EBP	364	727	93.57%
Cross Validation	EBP	364	727	60.97%
Test Set	EBP	301	601	62.33%
Training Set	REP	131	261	74.53%
Cross Validation	REP	131	261	59.70%
Test Set	REP	106	211	63.50%
Training Set	Unpruned	379	757	93.83%
Cross Validation	Unpruned	379	757	60.80%
Test Set	Unpruned	310	619	62.00%

Table 5: Categoric Data - Algorithm CART

Method	Prune	Leaves	Size	Accuracy
Training Set	CC	109	217	71.30%
Cross Validation	CC	109	217	60.90%
Test Set	CC	55	109	58.00%
Training Set	Unpruned	485	969	87.90%
Cross Validation	Unpruned	485	969	58.67%
Test Set	Unpruned	385	769	57.50%

© 2017 NSP
Natural Sciences Publishing Cor.

Table 6: Categoric data - Algorithm C4.5

Method	Prune	Leaves	Size	Accuracy
Training Set	EBP	475	949	88.47%
Cross Validation	EBP	475	949	58.37%
Test Set	EBP	368	735	56.83%
Training Set	REP	145	289	68.80%
Cross Validation	REP	145	289	56.93%
Test Set	REP	147	293	56.33%
Training Set	Unpruned	547	1093	89.63%
Cross Validation	Unpruned	547	1093	56.83%
Test Set	Unpruned	433	865	55.00%

The measures of validation considered are: Training Set, Cross-Validation and Test Set. Training Set uses the full database; in this case, the algorithm run the same database to learn patterns and to generate the results of the classification. For this reason, it presents the best accuracy, since the database behavior is already known. Using Cross-Validation, with $k = 10$, as described in Section 4.3, is also considered the full database, however, there is the replacements of data subsets. Finally, Test Set informs two data sets to the algorithm: training and test sets. First, the algorithm uses the training set to generate the model and to learn de data behavior. After the model formulating, the test set is used to validate the efficiency of the decision tree, verifying if the classification model classify the test set correctly.

Tables 3 and 4 refer to the results obtained from numerical data, by executing CART and C4.5 algorithms, respectively. Tables 5 and 6 present the results obtained from categoric data, using the same data sets considered to obtain the results from numerical data. The results consider pruned and unpruned trees. Results from CART algorithm consider the CC pruning criteria and unpruned trees and are presented in Tables 3 and 5. Executing C4.5 algorithm, two pruning criteria were considered: the EBP method, based on confidence factor, which is 0.25 (default value of WEKA) and the REP method, both described in Section 4.2.2. In addiction, unpruned trees results are also presented in Tables 4 and 6.

The accuracy, the number of leaves and the size of each generated decision tree are presented in Tables 3 to 6. These parameters refer to the complexity of the decision tree, as described in Section 3.

As evaluation measure, Table 7 shows the values of the total area under the ROC curve. As described in Section 4.3, the closer the curve is to the perfect classifier, the better it is at identifying positive values. Thus, the closer to 1, the better the classification is.

Each line of Tables 3 to 6 corresponds to a classification model. In order to generate the ROC curves and the respective areas, four models are chosen: the ones that use the Test Set method, since the Test Set evaluates the number of instances correctly classified and present fewer leaves and shorter trees; then, low complexity trees. It can allow that any user can read the decision tree. Therefore, the classification models selected in Table 7 are:

Model 1: numerical attributes using CART
Model 2: numerical attributes using C4.5
Model 3: categoric attributes using CART
Model 4: categoric attributes using C4.5

Table 7: Total area under the ROC curve for 4 models

Genre	Model 1	Model 2	Model 3	Model 4
Tango	0.996	1.000	0.983	0.975
Bachata	0.965	0.962	0.980	0.973
Bolero	0.882	0.847	0.881	0.799
Merengue	0.904	0.934	0.934	0.850
Salsa	0.869	0.826	0.834	0.767
Forró	0.788	0.745	0.823	0.838
Pagode	0.829	0.802	0.783	0.776
Sertanejo	0.842	0.726	0.818	0.800
Gaúcha	0.782	0.747	0.823	0.659
Axé	0.822	0.781	0.769	0.724
Weighted Avg.	0.868	0.837	0.863	0.816

From classification results, Confusion Matrices are generated for the same classification models. Tables 8 to 11 show that, as explained in Section 4.3, the diagonals present the higher numerical values, illustrating the efficiency of the classification models. Thus, results presented in Tables 8 to 11 refer to the Confusion Matrix for each one of the four models, obtained from the Test set. The other methods (Training set and Cross Validation), as expected, showed even better results, since they present better accuracy.

Table 8: Confusion Matrix for Model 1

Genre	a	b	c	d	e	f	g	h	i	j
Tango = a	60	0	0	0	0	0	0	0	0	0
Bachata = b	0	54	0	4	2	0	0	0	0	0
Bolero = c	1	0	33	0	5	3	7	9	2	0
Merengue = d	0	2	0	46	1	2	0	4	1	4
Salsa = e	0	4	0	3	41	3	1	3	3	2
Forró = f	0	3	4	4	6	21	9	9	4	0
Pagode = g	0	1	4	1	1	3	37	8	2	3
Sertanejo = h	0	0	8	0	6	1	3	29	7	6
Gaúcha = i	4	1	3	1	3	2	4	7	26	9
Axé = j	1	1	0	5	4	2	3	10	4	30

© 2017 NSP
Natural Sciences Publishing Cor.

Table 9: Confusion Matrix for Model 2

Genre	a	b	c	d	e	f	g	h	i	j
Tango = a	60	0	0	0	0	0	0	0	0	0
Bachata = b	0	54	0	2	3	1	0	0	0	0
Bolero = c	1	1	42	0	5	4	1	2	3	1
Merengue = d	0	1	0	52	0	0	2	0	2	3
Salsa = e	0	1	4	1	37	3	4	4	2	4
Forró = f	0	1	7	2	4	20	17	6	2	1
Pagode = g	0	0	5	1	0	4	37	5	4	4
Sertanejo = h	0	0	13	1	3	5	4	24	6	4
Gaúcha = i	1	1	7	2	3	5	9	3	27	2
Axé = j	0	4	1	3	6	2	3	6	7	28

Table 10: Confusion Matrix for Model 3

Genre	a	b	c	d	e	f	g	h	i	j
Tango = a	53	0	3	0	0	1	0	0	3	0
Bachata = b	0	53	0	0	5	0	0	0	0	2
Bolero = c	6	0	31	0	2	2	7	6	6	0
Merengue = d	0	4	0	42	1	2	1	2	1	7
Salsa = e	1	4	0	4	30	5	7	3	1	5
Forró = f	0	1	3	0	4	32	13	3	0	4
Pagode = g	0	2	3	2	1	9	30	10	1	2
Sertanejo = h	2	0	7	0	3	11	3	24	6	4
Gaúcha = i	3	0	4	3	5	3	1	9	29	3
Axé = j	0	3	1	4	6	4	5	10	3	24

Table 11: Confusion Matrix for Model 4

Genre	a	b	c	d	e	f	g	h	i	j
Tango = a	52	0	4	0	0	1	0	0	3	0
Bachata = b	0	53	0	3	4	0	0	0	0	0
Bolero = c	6	2	29	2	6	1	3	6	4	1
Merengue = d	0	2	0	45	1	2	1	2	1	6
Salsa = e	2	5	3	3	31	6	1	1	4	4
Forró = f	1	1	4	1	4	36	6	5	1	1
Pagode = g	1	0	4	3	1	11	25	11	3	1
Sertanejo = h	1	0	5	0	7	5	7	28	3	4
Gaúcha = i	1	0	6	3	10	5	1	10	16	8
Axé = j	0	3	1	9	7	6	6	4	1	23

In the last column of Tables 3 to 6, the accuracy is presented for each method. This performance measure is defined as the number of instances correctly classified divided by the total instances. Accuracy values are provided by Confusion Matrices, since the main diagonal indicates the number of genres correctly classified. Since the selected models in Table 7 use the Test Set, the Confusion Matrices must have 600 values. As described in Section 4.3, the higher the values are concentrated in main diagonal, the better the classifier is.

7 Conclusion

In this paper, we present a decision tree approach for the musical genres classification task. This approach is chosen due to the simple understanding and the possibility to examine the data relationships in all stages of the classification process, through the graphical analysis of the tree structure. CART and C4.5 decision tree algorithms are selected to induce binary tree. They provide different classification results, but it is possible to compare them performances.

Analyzing the results presented previously, we conclude the CART algorithm presents the best performance for database. Classification models 1 and 3 generate higher accuracy, fewer leaves and shorter trees than others. Then, CART presents low complexity trees. Besides, CART shows the higher values concentrated in main diagonal of the Confusion Matrices and the higher area under ROC curves. Therefore, the algorithm is useful to provide an automatic classification method for the considered database.

Table 4 shows, for Test Set method, that REP, which is a post-pruning method, generates better accuracy than EBP pruning method and provides low complexity tree (shorter size tree). In Table 6, REP method provides a considerable shorter size tree and the accuracy is not very decreased.

Analyzing Table 7, the higher the area under ROC curve, the better the discrimination provided by the classification model. The genres that present higher area values are: Tango, Bachata, Merengue, Bolero and Salsa. Since the weighted average of area under ROC curves are greater then 0.80 [5,10], the models present excellent performances for the classification task. Models 1 and 3 present the higher weighted averages, indicating the best performances.

Table 3 shows that using pruning method, the trees size can be reduced by almost 5 times. A similar reduction is noted in Table 5, which Test Set method is reduced by 7 times and the other in 4 times.

Although the C4.5 algorithm has not presented the best accuracy for database *Latin Music Database*, it still can be considered for the classification task, since it presents very good results, as shown in Tables 4, 6 and 7. Then, the decision trees algorithms executed in this paper are appropriated tools to classify the database.

As we can see in Tables 3, 4, 5 and 6, the pruning method is fundamental to reduce the complexity of the tree. This process makes the decision tree easier to be interpreted. By comparing the accuracies of pruned and unpruned trees, we can notice a variation not significant, however, considering the number of leaves and the size of pruned trees, we can observe a very relevant reduction in the complexity of the trees.

The methodology proposed in this paper was applied to classify Latin musical genres. It can be applied to classify other different sets of musical genres. In this case, the input features, as timbral texture, beats, frequency and

© 2017 NSP
Natural Sciences Publishing Cor.

the others, are the same. The numerical data related to these features have to be collected for the new genres in order to be classified using the proposed methodology.

As future perspectives, we intend to apply the fuzzy decision trees in order to classify musical genres automatically, considering imprecisions and fusions among genres classes.

References

[1] J. J. Aucouturier and F. Pachet. Representing musical genre: A state of the art. *Jornal of New Music Research*, **32**(1):83–93, (2003).

[2] L. Breiman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Wadworth & Books/Cole Advanced Books & Software, (1984).

[3] D. Castán, A. Ortega, and E. Lleida. Speech/music classification by using the C4.5 decision tree algorithm. *FALA 2010 - VI Jornadas en Tecnología del Habla and II Iberian SLTech Workshop*, pages 197–200, (2010).

[4] P. Cichosz. *Data mining algorithms explained using R*. John Wiley & Sons, (2015).

[5] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, **27**:861–874, (2006).

[6] F. Fernández, F. Chávez, R. Alcalá, and F. Herrera. Musical genre classification by means of fuzzy rule-based systems: A preliminary approach. *IEEE Congress on Evolutionary Computation, IEEE CEC*, **13**(2):303–319, (2011).

[7] T. Hillecke, A. Nickel, and H. V. Bolay. Scientific perspective on music therapy. *Annals of the New York Academy of Sciences*, pages 1–12, (2005).

[8] K. Jensen and J. Arnspang. Binary decision tree classification of musical sounds. *ICMC Proceedings 1999*, pages 414–417, (1999).

[9] P. N. Juslin and P. Laukka. Expression, perception, and induction of music emotion: A review and a questionnaire study of everyday listening. *Journal of New Music Research*, **33**(3):217–238, (2004).

[10] B. Lantz. *Machine learning with R*. Packt Publishing, (2013).

[11] D. T. Larose and C. D. Larose. *Discovering knowledge in data. An introduction to data mining*. John Wiley & Sons, (2014).

[12] T. Li and M. Ogihara. Music genre classification with taxonomy. *IEEE*, **5**:197–200, (2005).

[13] N. M. Norowi, S. Doraisamy, and R. Wirza. Factors affecting automatic genre classification: An investigation incorporating non-western musical forms. *International Conference on Music Information Retrieval (ISMIR)*, pages 13–20, (2005).

[14] N. Patel and S. Upadhyay. Study of various tree pruning methods with their empirical comparison in weka. *International Journal of Computer Applications*, **60**:20–25, (2012).

[15] D. D. Patil, V. M. Wadhai, and J. A. Golhale. Evaluation of decision tree pruning algorithms for complexity and classification accuracy. *International Journal of Computer Applications*, pages 23–30, (2010).

[16] J. R. Quinlan. Induction of decision trees. *Kluwer Academic Publishers*, pages 81–106, (1986).

[17] J. R. Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies*, **27**(3):221–234, (1987).

[18] J. R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, (1993).

[19] L. Rokach and O. Maimon. *Data mining with decision trees: Theory and applications*, volume **69**. World Scientific, (2008).

[20] C. N. Silla-Jr, A. L. Koerich, and C. A. A. Kaestner. The Latin music database. *Proc. International Society for Music Information Retrieval*, pages 451–456, (2008).

[21] C. N. Silla-Jr, A. L. Koerich, and C. A. A. Kaestner. A feature selection approach for automatic music genre classification. *International Journal of Semantic Computing*, **3**(2):183–208, (2009).

[22] D. Temperley. *Music and Probability*. MIT Press, (2007).

[23] R. Timofeev. *Classification and Regression Trees (CART) Theory and applications - Master Thesis*. Humboldt University - Berlim, (2004).

[24] G. Tzanetakis and P. Cook. Marsyas: A framework for audio analysis. *Organized Sound*, **4**:169–175, (1999).

[25] G. Tzanetakis and P. Cook. Music genres classification of audio signals. *IEEE transactions on speech and audio processing*, **10**(5):293–302, (2002).

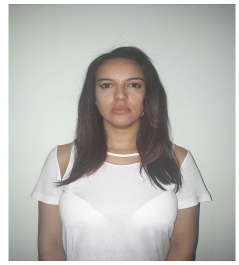
[26] C. Weihs, D. Jannach, I. Vatolkin, and G. Rudolph (Eds.). *Music data analysis, Foundations and Applications*. CRC Press, (2017).

[27] Y. Yuan, Q. Song, and J. Shen. Automatic video classification using decision tree method. *Proceeding of the First International Conference on Machine Learning and Cybernetics*, pages 1153–1157, (2002).

Glaucia M. Bressan received a teaching degree in Mathematics with emphasis in computation from Federal University of Sao Carlos (UFSCar), Brazil, in 2000, a Master's degree in Computational and Applied Mathematics from the University of Sao Paulo (USP) in 2003 and her PhD in Electrical Engineering at University of Sao Paulo (USP) in 2007. She did a post-doctoral research in Electrical Engineering at University of Sao Paulo (USP) in 2008. She works as a professor and researcher at the Federal University of Technology of Parana (UTFPR) in the Department of Mathematics, where she has been a professor since 2012. Her interests relate to Operational Research, machine learning, Bayesian Networks and Fuzzy Logic.

© 2017 NSP
Natural Sciences Publishing Cor.

Beatriz C. F. de Azevedo is Undergraduate of Control and Automation Engineering at the Federal University of Technology of Parana (UTFPR). Undergraduate research scholarship in the Department of Mathematics. Her interests relate to Mathematics Modeling and Machine Learning.



Elisangela Ap. S. Lizzi is Bachelor's degree in Statistics from the Federal University of Sao Carlos (2010), with a Master's Degree (2012) and PhD (2015) in biostatistics by University of Sao Paulo at Ribeirao Preto Medical School. Currently she works as a professor and researcher at Federal University of Technology of Parana (UTFPR) in the Department of Mathematics, in the field of statistics and math, with emphasis on applied statistics and biostatistics. Mainly in the following subjects: quantitative methods in health, public health and epidemiology, time series models, machine in learning, spatial statistics and bayesian mapping of diseases.