Advances in Mathematical
Sciences and Applications
Vol. 29, No. 2 (2020), pp. 295–310

GAKKOTOSHO
TOKYO JAPAN

INPUT FEATURES SELECTION TO IMPROVE THE PERFORMANCE OF A FUZZY CLASSIFIER

Glaucia M. Bressan *, Bruno C. Coscarelli,
Matheus H. Pimenta-Zanon, Beatriz C. F. de Azevedo

Universidade Tecnológica Federal do Paraná - Brazil
Departamento Acadêmico de Matemática.
1640, Alberto Carazzai, 86300-000, Cornélio Procópio
(E-mail: glauciabressan@utfpr.edu.br)

Abstract. This paper presents a proposal for the selection of input features for an automatic classification system. This selection must identify the most representative features, called *golden set*, of the problem to be studied and contribute to the improvement of the classifier's performance, increasing its success rate. Then, an automatic classification system is constructed using fuzzy Logic, since it is possible to consider uncertainties and the fuzziness among sets to be classified. In order to apply the presented proposal, a database from [8] is considered, which is constituted by 16 input features to classify 10 Latin Musical genres. The proposal of this paper was able to select the 4 of 16 most representative input features and to improve the classifier's performance, increasing the success rate from 90.7% (result obtained by [3], using the 16 input features) to 97% (using the *golden set*).

*corresponding author: glauciabressan@utfpr.edu.br
Communicated by Editors; Received June 10, 2020
AMS Subject Classification: 03B52, 03A05, 68T10.
Keywords: fuzzy classifier; neurofuzzy system; paraconsistent logic; Latin musical genres.

1 Introduction

The classification task is traditionally seen as the problem of assigning a label to data. According to [9], the classification can be defined as the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. The problem of data classification has numerous applications in a wide variety of mining applications [1]. An important question about the classification processes is that no clear definition and borders can be easily established. This action requires a large amount of human or computational effort and dedication. Due to the increasing number of data and their fusion, the decision process can bring doubts and hesitations in the classification task. Therefore, data mining and machine learning techniques were developed to automatically discover knowledge and recognize patterns from these data [9].

The act of classifying is important to categorize data or objects, assign tasks and assist in decision making. In order to do a classification, it is essential that there is a selection of the input features, which describe the phenomenon under study. For this reason, feature selection has been the focus of interest for quite some time and much work has been done [4]. According to [9], *feature extraction* approaches project features into a new feature space with lower dimensionality and the new constructed features are usually combinations of original features. On the other hand, the *feature selection* approaches aim to select a small subset of features that minimize redundancy and maximize relevance to the target such as the class labels in classification. Both feature extraction and feature selection are capable of improving learning performance, lowering computational complexity, building better generalizable models, and decreasing required storage. In this sense, in order to improve the classification performance, in addition to choosing between the classification methods, we have to pay attention to which subset of features to employ in the classifier [10].

With the presence of a large number of features, a learning model tends to overfit, resulting in the degeneration of their performance. For the classification task, feature selection aims to select subset of highly discriminant features. It selects features that are capable of discriminating samples that belong to different classes. Feature selection is a widely employed technique for reducing dimensionality among practitioners. It aims to choose a small subset of the relevant features from the original ones according to certain relevance evaluation criterion, which usually leads to better learning performance (higher learning accuracy), lower computational cost and better model interpretability [9].

This paper presents a proposal for the selection of input features for an automatic classification system. When many input features must be considered in the classification task, there is a great concern with the computational cost, since the number of combinations between the inputs is very large. Thus, the impossibility of using the whole list of features at once has imposed the selection of smaller sets of features to serve as the bases for the systems. In this context, the goal of this paper is to propose a selection method of input features for an automatic classification system. The proposed selection method must identify the most representative features set, called *golden set*, of the problem to be studied and then provide an improvement of the classifier's performance, increasing its success rate. After the selection process, an automatic classification system is constructed

using fuzzy Logic [6], since it is possible to consider uncertainties and the fuzziness among sets to be classified. For that, the numerical parameters of the membership functions are trained by a *neurofuzzy* system [5], since the training based on numerical data avoids subjectivity in the process of discretizing the input features. In order to apply the presented proposal, a database from [8] is considered, which is constituted by 16 input features to classify 10 Latin Musical genres. Finally, results obtained from [3], using the 16 input features, are compared with the results obtained from the proposal presented in this paper, using the *golden set*. The main expected contribution is to obtain a larger classification success rate. So, the question of which is (or which are) the most suitable sets of features to base a classification system is in order.

Having an answer to this question is important for two reasons: Firstly, it may provide support to the construction of even better classification systems. Second, it may support further researches on musical theory. Being so, this is a question that shall be explored. In fact, it is the question that will be tackled in this paper.

This paper is organized as follows: Section 2 describes the database used in this work to apply the presented proposals. Section 3 presents the proposed algorithmic procedure to select input features for the classification task and to obtain the *golden set*. Then, the fuzzy classification system, using the *golden set*, is presented in Section 4. In Section 5, an improvement of the classification performance is presented through the paraconsistent system. Finally, the conclusion is described in Section 6.

2 Description of Database

In order to apply the methodology proposed in this paper, which consists of selecting input features to improve the performance of a fuzzy classifier, and in order to compare results from this paper with those in the literature, a set of the available music in *Latin Music Database*, from [8], is used. This database is composed by 3.000 musical recordings database from 10 different Latin Music genres: tango, salsa, forró, axé, bachata, bolero, merengue, gaúcha, sertanejo and pagode. The MARSYAS framework was employed for feature extraction [11]. 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 about these features can be seen in [7]. According to [3], a set of 16 features, shown in Table 1, are identified and these features are able to describe music genres.

In [3], a fuzzy system for musical genres classification is constructed and tested from the 3000 instances. Ninety percent of those (2700 songs) compose the training group, which is used for constructing the system, and the remaining ten percent (300 songs) compose the testing group, which is used for testing the performance of the resulting method. In order to compare the obtained results, the same percentage of training and testing sets are considered here. Besides, in [3], for each one of the 10 music genres, three fuzzy classification systems were constructed, since the 16 input features were divided in subgroups (beat related, timbre texture and pitch related), due to the computational cost. Each of these systems takes a song as input and gives as output the probability for

Table 1: 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	First MFCC mean
8	Second MFCC mean
9	Third MFCC mean
10	Fourth MFCC mean
11	Fifth MFCC mean
12	The overall sum of the histogram (pitch strength)
13	Period of the maximum peak of the unfolded histogram
14	Range of the maximum peak of the folded histogram
15	Period of the maximum peak of the folded histogram
16	Pitch interval between the two most prominent peaks of the folded histogram

this song to belong to the reference genre. In the current paper, we also construct one fuzzy classification system for each one of the 10 music genres.

Although the beat related, timbre texture and pitch related groups seem the most suitable choices and the resulting systems have a good level of accuracy, the premise that those are indeed the most suitable choices is just something that has been taken for granted.

3 Proposal for Features Selection

In this section, the proposal for selecting features and then finding the *golden set*, in order to improve the classifier performance, is described. For each song from the database described in the previous section there are seventeen pieces of information: The measure of each of the sixteen input features and indication of the genre to which the song belongs.

The task of identifying the relevant sets of features is accomplished in six steps.

Step 1: A database where each element corresponds to one of the $2^{16} - 1$ possible nonvoid binary sets of features is created. This will be called *database of feature sets*.

Step 2: It is considered that a set of features classifies a given song as belonging to a given genre when, for every feature that belongs to that set, the parameter of that song belongs to the average interval. It is considered that a song is classified by a given set of features when it is classified by that set as belonging to at least one genre. The classification of a song by a set of features is considered to be correct when it is unique

and matches the classification registered in the last input of the vector that corresponds to that song.

Algorithm 1: Procedure to determine whether a given set of parameters classify a given song as a belonging to a given genre

input : A song, a set of parameters and a genre
output: Classification of musical genre

```
1  Define variable "answer" with initial value yes
2  forall Sets created in Step1 do
3      while i <number of features do
4          if Input has value 1 then
5              if If the value of parameter i of the song is out of the average interval
                for parameter i in the genre then
6                  Attribute value no to the variable "answer"
7              end
8          end
9          i = i + 1
10     end
11 end
```

The second step is to register how many songs each set of features classifies correctly and incorrectly and calculate the success rate as the rate of the number of correctly classified songs and the number of classified songs. In this way, the database of feature sets is completed.

Algorithm 2: Step 2: Procedure to compute the success rate of a set of features

input : A set of parameters
output: The number of right and wrong classifications and the success rate

Define the variables $x, y, right, wrong, success$ with initial value 0
forall *Songs* **do**
  forall *Genres* **do**
    if *The set of features classified the song as belonging to the genre* **then**
      $x = x + 1$
      if *The classification is correct* **then**
        $y = 1$
      end
    end
  end
  if $x = y = 1$ **then**
    $right = right + 1$
  end
  if $x > 1$ *or* $(x = 1$ *and* $y = 0)$ **then**
    $wrong = wrong + 1$
  end
end
if $right + wrong > 0$ **then**
  $success = \dfrac{right}{right + wrong}$
end
Store the number of correctly classified songs, the number of incorrectly classified songs and the success rate.

Step 3: The sets of features that have success rate 100% are stored in a separate database, which will be called *plain database*.
In order to enhance the confidence in the relevance of those sets of features in the intended sense for the term, it is necessary to test its performance and compare the results obtained in the training database with those obtained in the testing group. This is what will be done in the next steps.

Step 4: A song is considered to be classified by the plain database as belonging to a given genre if it is classified by at least one set from that database as belonging to that genre.

The fourth step is to create a procedure that verifies whether a song is classified by the plain database and, if so, what its classification is. This procedure is a classification

system that will be called *plain system.*

Algorithm 3: Step 4: Procedure to classify a song according to the *plain system*

```
   input  : A song
   output: The classification of the song
 1 Define the variable classification with initial value "not classified"
 2 forall Sets of parameters do
 3     if The value of the variable classification is different from "ambiguous" then
 4         forall Genres do
 5             if The set of features classifies the song as belonging to the genre then
 6                 if The value of the variable classification is "not classified" then
 7                     Attribute to the variable classification the name of the genre
 8                     else
 9                         if The value of the variable classification is different from
                             the name of the genre then
10                             Attribute to the variable classification the value
                                "ambiguous"
11                         end
12                     end
13                 end
14             end
15         end
16     end
17 end
```

Step 5: A song that is classified by the plain system is considered to be correctly classified if every set that classifies it gives the correct classification. The success rate of a classification system with respect to a database of songs is the rate of the number of correctly classified songs and the number of classified songs. The effectiveness index of a classification system with respect to a database of songs is the rate of the number of classified songs and the total number songs.

The fifth step is to create a procedure that finds the success and effectiveness rates of

the plain system with respect to a given song database.

Algorithm 4: Step 5: Procedure to compute the success and effectiveness rates of the plain system with respect to a database

input : A database of songs
output: The success and effectiveness rate of the plain system with respect to the database

Define the variables $classification$ and $verification$ with initial value "unknown" and the variables $right$, $wrong$, $total$, $success$ and $effectiveness$ with inicial value 0
forall *Songs* **do**
    $total = total + 1$
    Attribute the classification of the song through Algorithm 3 to the variable $classification$
    Attribute the correct classification of the song contained in the database to the variable $verification$
    if *The set of features classified the song as belonging to the genre* **then**
        $x = x + 1$
        if *The the variables classification and verification have the same value* **then**
            $right = right + 1$
            else
                if *The value of the variable classification is different from "not classified" and "ambiguous"* **then**
                    $wrong = wrong + 1$
                end
            end
        end
    end
end
if $right + wrong > 0$ **then**
    $success = \frac{right}{right + wrong}$
end
$efficiency = \frac{right + wrong}{total}$

Step 6: The final step is to test the effectiveness index of the plain system with respect to the training group (the success rate is obviously 100%) and its success and effectiveness rates with respect to the testing group.

3.1 Results and Discussion

The results of this data treatment are:

- A total of 829 sets of features had success rate 100% and compose the plain database.
- A total of 169 out of 2700 songs from the training group have been classified, which means that the effectiveness index of the plain system with respect to the training group is 6.25%.
- A total of 18 out of 300 songs from the testing group have been classified, which means that the effectiveness index of the plain system with respect to the testing group is 6%.
- The success rate of the plain system with respect to the testing group is 100%.

The results suggest that there is in fact a group of relevant sets of features and that this group is the plain database. The fact that the success rate of the plain system with respect to the testing group is 100% gives force to this suggestion. Naturally, 6.25% is not a high effectiveness index, but it is significant anyway. The fact that the effectiveness index with respect to the testing group is very close to the index with respect to the training group is also encouraging.

The conclusion that follows from the considerations above is that it is worthy to take a closer look at the plain database. Having 829 reliable sets of features is definitely a gross result, which calls for refinement. The set of features $\{7, 13, 14, 15\}$, that is, {First MFCC mean, Period of the maximum peak of the unfolded histogram, Range of the maximum peak of the folded histogram, Period of the maximum peak of the folded histogram} stands out from all the other sets of features that presented 100% of successes for the following reasons:

- It classifies more than 13% of the total of classified songs;
- Among the 829 sets from the plain database, 208 are extensions of it;
- All the sets that classified more than 3% of the total of classified songs are extensions of it;
- All the sets that classified more than 2% of the total of classified songs have at least three of its four features.

The conclusion of this section is that $\{7, 13, 14, 15\}$, that is {First MFCC mean, Period of the maximum peak of the unfolded histogram, Range of the maximum peak of the folded histogram, Period of the maximum peak of the folded histogram}, is the most likely candidate to be the most apt set of features to inform about genre classification. At this point, it deserves to be called *golden set*.

This result is somehow a surprise. In fact, the three groups of parameters selected in [3], namely beat related, timbre texture and pitch related, should be expected to be the most apt sets. However, the most apt set according to the conclusion of this section is a set that mixes two groups, taking one timbre texture and three pitch related parameters.

Table 2: Membership functions parameters for the *Tango* genre.

Feature	Parameters		
	low	medium	high
1	[0.2083 , 0.01915]	[0.2083 , 0.5096]	[0.2083 , 1]
2	[0.1968 , -0.001789]	[0.1968 , 0.4634]	[0.1968 , 0.9268]
3	[0.03944 , 0.01777]	[0.03944 , 0.1107]	[0.03944 , 0.2035]
4	[0.2123 , 0]	[0.2123 , 0.5]	[0.2123 , 1]

4 Fuzzy Classification System Based on the Golden Set

This section presents the Fuzzy Classification System, using the Golden Set, identified in the previous section. In order to construct the classification system, the numerical data set is constituted by the same 3000 instances (each one is a song) and 5 columns: the *golden set* composed by the 4 selected input features (First MFCC mean - Feature 1, Period of the maximum peak of the unfolded histogram - Feature 2, Range of the maximum peak of the folded histogram - Feature 3, Period of the maximum peak of the folded histogram - Feature 4) and, since it is a supervised learning system, the fifth column is the name of the music genre (the output).

As done in [3] and for comparison purposes, 90% of numerical data (2700 instances) compose the training set, which is used in the neurofuzzy system [5] in order to train the parameters of the membership function [6], and the remaining 10% (300 instances) compose the testing group, used for testing the performance of the classifier, i.e, the success rate.

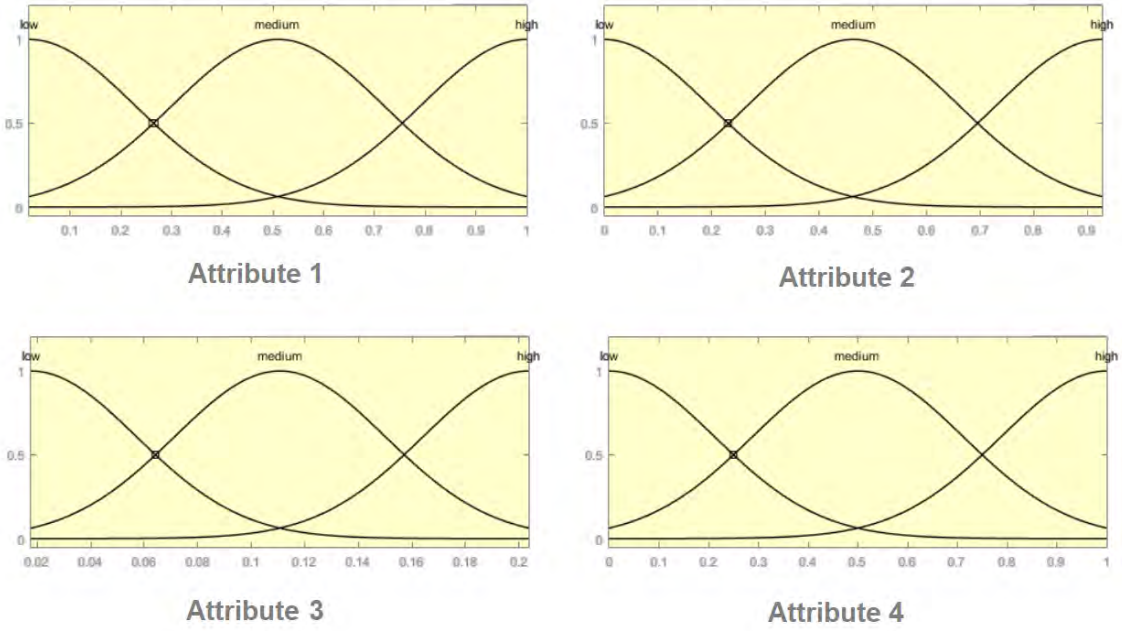
For each one of the ten Latin musical genres, a neurofuzzy system is created, which provides a method for the fuzzy modeling procedure to *learn* information from data. This learning method works as a neural network [2]. The computation of the parameters of the membership functions is facilitated by a gradient vector, which provides a measure of how well the fuzzy inference system is modeling the input/output data for a given set of parameters. Once the gradient vector is obtained, any of several optimization routines could be applied in order to adjust the parameters so as to reduce some error measure. In this paper is used the hybrid method: a combination of least square estimation and backpropagation for membership function parameter estimation [2,5].

Since we have a classification system for each one of the 10 Latin musical genres of the database, we may consider as an example, the genre *Tango*. Therefore, from neurofuzzy training, the 4 input features are discretized in 3 categories: low, medium and high, according to the ranges described in Table 2 for Tango genre.

About membership functions, the shapes employed in this paper is the Gaussian, since it presents a gradual pass between the borders of classes. The output membership functions are modeled as constant functions, since they are related to the Latin musical genres considered. Therefore, the fuzzy inference system is Sugeno type [6]. Figure 1 illustrates the behavior membership functions, which are related to the input features of

the *Tango* genre.

Figure 1: Membership functions related to the input features of the *Tango* genre.



Every fuzzy system is based on a linguistic rules set called 'if-then' rules base [6]. For 'if-then' rules construction, the classes of the inputs (low, medium and high) are logically combined applying the 'and' operator, which indicates that the inputs occurs simultaneously. Since we have one fuzzy system for each one of the 10 genres, the 4 input features of each system generates $3^4 = 81$ rules. For example, one rule of the *tango* system is described as:

IF feature 1 is MEDIUM, AND feature 2 is HIGH, AND feature 3 is LOW, AND feature 4 is HIGH, THEN genre is TANGO

After the rules base is generated and the membership functions parameters are fitted by the neurofuzzy system, these data are inserted into the fuzzy classification system in order to automatically classify the Latin musical genres considered.

As 10% of data are considered the testing set (300 instances) and the database is composed by 10 genres, so each one of the 10 fuzzy systems has 30 instances of test. The 30 test instances related to the *Tango* genre, for example, are input to the Fuzzy Classification System structured for the *tango* genre, as done in [3], for comparison purpose. Similarly, other test instances, related to the other musical genres, are input to the corresponding Fuzzy Classification System and the final membership percentage is obtained. This process checks whether the output for the correct genre is high enough. The 30 test instances related to the each genre are input to the Fuzzy Classification System structured for the 9 others genres in order to check whether the outputs of the others are low enough. Table 3 describes the success rate (or the accuracy) of the classification of each one of the 10 musical genres.

As a result, an arithmetic mean is obtained: 97% of success rate (accuracy). Moreover, this membership percentage demonstrates that the instance can be a fusion of two or more

Table 3: Fuzzy classification accuracy for testing instances.

Genre	Correctly Classified	Incorrectly Classified	Total	Accuracy
Tango	28	2	30	93.3%
Salsa	30	0	30	100%
Forró	30	0	30	100%
Axé	30	0	30	100%
Bachata	30	0	30	100%
Bolero	29	1	30	96.7%
Merengue	29	1	30	96.7%
Gaúcha	28	2	30	93.3%
Sertanejo	28	2	30	93.3%
Pagode	29	1	30	96.7%

musical genres. In order to compare the fuzzy classification with the results from [3], the arithmetic mean obtained, using the 16 input features, was 90.7% of success.

5 Improving the Classification System

As described in Section 3.1, since only 829 sets of features out of 65,535 possibilities are considered in the plain system, it is highly recommendable that less successful sets be taken into account. In order to do so, a second classification system will be created.

In the daily practice of science, industry or any activity that involves decision making, uncertainty is an ubiquitous reality. As far as logic is concerned, there are two branches of study that can be regarded as the very tools for coping with the problem, namely fuzzy and paraconsistent logics. The former regards uncertain assertions as blurred information, that shall be treated statistically; The latter regards uncertain assertions as inconsistent information, that can be admitted to be true and false at the same time without causing deductive triviality. The classification system developed in this section will illustrate this point. For reasons that will soon became clear, it will be called *paraconsistent system*.

The first two steps of its construction would be the same as those for the construction of the plain system. As the work is already done, the construction of the paraconsistent system will be described in steps 3' and 4'.

Step 3': The sets of features that have success rate greater than or equal to 50% and lower than 100% are stored in a separate database.

Step 4': The classification of a given song is made in the following manner: A vector is created with an input for each genre. The number of feature sets that classify that song as belonging to a given genre is stored in the input that corresponds to that genre. The song is classified as belonging to a given genre if the number in the corresponding input is greater than the sum of the numbers in the remaining inputs. This is how the

paraconsistent system works.

Algorithm 5: Step 4': Procedure to compute

input : A song
output: The songs classification of the song

Define a vector with 16 inputs with initial value 0
forall *Sets of parameters* **do**
  forall *Genres* **do**
    if *The set of parameters classifies the song as belonging to the genre* **then**
      Add 1 to the index i of input corresponding to the genre
    end
  end
end
$x = \max\{\text{value among the inputs}\}$
$y = \sum_{i=1, i \neq x_i}^{16} input_i$
if $x > y$ **then**
  Attribute the name of the genre that corresponds to the input of greatest value to the variable $classification$
end

Before going on, it is in order to observe that the name 'paraconsistent system' is justified by the fact that the sets of features are allowed to classify a given song as belonging to more that one genre.

Step 5': Step 5' consists in creating a routine that finds the success and effectiveness rates of the paraconsistent system with respect to a given song database.

The algorithm used is the same as the one of Step 5 using the algorithm of Step 4' instead of the algorithm of Step 4 to determine the classification of a song.

Step 6': The final step is to test the success and effectiveness rates of the paraconsistent system with respect to the training and testing groups.

5.1 Results and Discussion

The results of this data treatment are:

- A total of 1664 sets of features had success rate between 50% and 100% and compose the paraconsistent database.

- A total of 143 out of 2700 songs from the training group have been classified, which means that the effectiveness index of the plain system with respect to the training group is $5,29\%$.

- A total of 23 out of 300 songs from the testing group have been classified, which means that the effectiveness index of the plain system with respect to the testing group is $7,67\%$.
- The success rate of the paraconsistent system both with respect to the training and testing groups is 100%.

The fact that the paraconsistent system presented success rate 100% with respect to both the training group, which was the base for its creation, and the testing group is again encouraging. So is the fact that the efficiency indices are very close.

The question of whether the plain and paraconsistent systems classify approximately the same set of songs rises naturally. The answer is no. The intersection of the set of songs from the training group classified by the plain and paraconsistent systems has only 13 songs. This suggests that a classification system concatenating the two systems shall be created. This system will be called *concatenated system*. This system consists in checking the classification of a song by the plain system and, in case the classification is not done, checking it by the paraconsistent system.

Algorithm 6: Procedure for the concatenation of the plain and the $i\%$ systems

input : A song
output: The song's classification

```
1 Define the variable classification with initial value "notclassified"
2 forall Systems in the sequence (plain, paraconsistent) do
3 |   if Variable classification has value "not classified" then
4 |   |   Attribute to variable classification the classification of the song
  |   |    according to the system
5 |   end
6 end
```

The results for the concatenated system are:

- A total of 400 out of 2700 songs from the training group have been classified, which means that the effectiveness index of the plain system with respect to the training group is 14.8%.
- A total of 37 out of 300 songs from the testing group have been classified, which means that the effectiveness index of the plain system with respect to the testing group is 12.3%.
- The success rate of the concatenated system both with respect to the training and testing groups is 100%.

Again, the indices for the training and testing groups are similar and this attests to the reliability of the method. A success of 100% is definitely a great result and an effectiveness around 13% is significant.

An extra attempt of gaining efficiency was made. For this, the group of sets of features with success rate greater than 50% was split in the groups of sets with success rate from

50% to 60%, from 60% to 70%, from 70% to 80%, from 80% to 90% and from 90% to 100%. Then, a system like the paraconsistent system was constructed for each of this groups. Finally, the systems were concatenated. The results of this variation of the paraconsistent system presented results that are very close to those of the original system. For this reason, it will not be presented here.

6 Conclusion

The core result of this article is the determination of a golden set of features that turns out to be the most effective one for music classification, at least when the interest is on the ten Latin genres that were taken under consideration. Although this is a result that is interesting for its own, it is order to point its practical applications and the fields of research to which it may contribute.

Music Classification for Latin Genres An efficient classification system for Latin genres was provided. That system is ready to be used by the industry.

Music Classification in General The classification system that was provided is likely to work finely to other genres. Besides that, the same tool developed to identify the best set of features to Latin genres can be used to identify the best set of features for other genres.

Musical Theory The question of why the golden set obtained in this article works so well for music classification is a one that shall be investigated from a musical theory point of view. Raising this question is definitely a contribution of this article to this area.

Philosophy The problem of universals is a philosophical question that amounts to ancient times, starting with Plato and Aristotle, passing through medieval times and surviving even to the most sophisticated attempts of solution provided by contemporary investigation. Actually, it would be more appropriate to say that the problem of universals is a family of problems that ranges several areas of philosophy. The point that matters here is that being a tango, for example, should be a universal concept. Nevertheless, it is an apparently complex concept that was shown to be able to be apprehended by a computer system within surprisingly few features.

References

[1] C. C. Aggarwal. *Data classification: algorithms and applications*. CRC press, 2014.

[2] H. D. Beale, H. B. Demuth, and M. T. Hagan. *Neural network design*. Pws, Boston, 1996.

[3] G. M. Bressan, J. M. A. Bernardi, and C. N. Silla Jr. A fuzzy-based approach for the latin music genres intelligent classification. *Advances in Mathematical Sciences and Applications*, 27(1):65–80, 2018.

[4] M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[5] J-SR Jang. Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3):665–685, 1993.

[6] W. E Pedrycz and F. Gomide. *An Introduction to Fuzzy Sets*. MIT Press, 1998.

[7] 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.

[8] C.N. Silla Jr, V Koerich, and C.A.A Kaestner. The latin music database. *International Society for Music Information Retrieval*, 2008.

[9] J. Tang, S. Alelyani, and H. Liu. Feature selection for classification: A review. *Data classification: Algorithms and applications*, pages 37–64, 2014.

[10] M. Tsuchiya and H. Fujiyoshi. Evaluating feature importance for object classification in visual surveillance. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 978–981. IEEE, 2006.

[11] G Tzanetakis and Perry Cook. Marsyas: A framework for audio analysis. *Organized Sound*, 4:169–175, 1999.