Selection of Training Instances for Music Genre Classification

Miguel Lopes, Fabien Gouyon
INESC Porto, Portugal
{mglopes, fgouyon}@inescporto.pt

Alessandro L. Koerich
PPGIA-PPUCPR, Brasil
alekoe@ppgia.pucpr.br

Luiz E. S. Oliveira
UFPR, Brasil
lesoliveira@inf.ufpr.br

Abstract

In this paper we present a method for the selection of training instances based on the classification accuracy of a SVM classifier. The instances consist of feature vectors representing short-term, low-level characteristics of music audio signals. The objective is to build, from only a portion of the training data, a music genre classifier with at least similar performance as when the whole data is used. The particularity of our approach lies in a pre-classification of instances prior to the main classifier training: i.e. we select from the training data those instances that show better discrimination with respect to class memberships. On a very challenging dataset of 900 music pieces divided among 10 music genres, the instance selection method slightly improves the music genre classification in 2.4 percentage points. On the other hand, the resulting classification model is significantly reduced, permitting much faster classification over test data.

1. Introduction

In this paper, we tackle the problem of classifying music into different genres based on the analysis of the audio signal. Music genres are categorical labels created by humans in order to determine the style of music. The automatic classification of music genres is nowadays an important task, because music genre is a descriptor that is used to organize large collections of digital music [1, 7, 8, 11]. From a pattern recognition perspective, the task of automatic music genre classification from audio poses an interesting research problem: music signals are complex, time-varying signals requiring the processing of potentially very numerous high-dimensional instances.

Most of the approaches for music genre classification are variations over the so-called bag-of-frame (BOF) approach [1]: they represent music genres by collections of feature vectors computed on short audio frames, with no temporal modeling, and follow the assumption that all frames are equally informative and hence that music genres can satisfactorily be represented by long-term statistical distribution of such features.

In contrast to this approach, in a simulated perceptual similarity experiment, Aucouturier et al. [2] show that not considering the least representative frames of a music piece (representing less than 5% of the whole statistical distribution —i.e. Gaussian Mixture Modeling (GMM) in their case) have a significant negative impact on precision (a decrease of almost 10 percentage points in their experiments). Further, they show that including a larger number (i.e. around 25%) of more representative frames of a music piece is in fact detrimental to precision. A recent approach to music genre classification [6] proposes to select the most representative frames of each music piece (i.e. the most statistically representative in a GMM representation of each individual piece), and to create a global dictionary from these (akin to vector quantization). Each music piece is then represented into a sequence of symbols drawn from this dictionary, used in test/train classification experiments based on a language modeling approach. This approach results in an improvement in the classification accuracy of music pieces, compared to more traditional approaches.

In this paper we propose to improve upon the BOF approach with the aim of finding the critical instances that form boundaries to differentiate data points of different classes. The proposed novelty is to assume that, in each specific genre, a number of such instances are in fact not representative of that genre, i.e. do not provide good discrimination with respect to other genres, and should therefore be discarded prior to classifier training. We propose a pre-processing step in which training data is used to learn which instances in each class offer the best discrimination with respect to other classes and, complementarily, which instances should be discarded. In order to train a final classifier, each class is then represented only by its most discriminative instances.
This paper is organized as follows: Section 2 presents the proposed method for instance selection. Experimental results on the impact of instance selection in music genre classification are presented in Section 3. Section 4 presents a discussion on the results and the conclusions are stated in the last section.

2. Instance Selection Method

The goal of the instance selection method is to find the instances of each class which provide the best discrimination with respect to other classes. The approach we have selected for this consists in computing features on 46ms signal frames, with no overlap (i.e. 1,024 samples at 22,050 Hz), on 30-second segments around the middle of each audio file [9]. We then compute feature averages and variances (resulting hence in 34-dimensional feature vectors) over texture windows of 5 consecutive frames (i.e. 232 ms), with 4/5 overlap between texture windows (Figure 1). These 34-dimensional feature vectors over texture windows are the instances used in the classification experiments. Each music piece is then represented by 646 instances.

We use seventeen audio signal features to represent the music pieces: zero crossing rate, spectral centroid, rolloff frequency, spectral flux and the 13 Mel frequency cepstral coefficients (MFCCs, including MFCC0). Feature extraction can be reproduced with the free Marsyas framework.\(^1\)

Our approach requires the training and testing of two successive classifiers as shown in Figure 2. Indeed, the training of our final N-class classifier is carried out only with part of the original data: we remove all those instances of the original data that are misclassified in the process of training and testing a first N-class classifier. In order to avoid overfitting we propose the following procedure, implementing a variant of 3-fold cross-validation:

1. Train a N-class classifier \(C_{1}^{1} \) with data fold \(F1\).
2. Perform classification of each instance in a second fold \(F2\) with \(C_{1}^{1}\). For each instance in that fold, compare ground-truth labels with classifier outputs and remove all erroneously classified instances, yielding the smaller fold \(F2'\).
3. Train a N-class classifier \(C_{2}^{2} \) with \(F2'\).
4. Perform classification of the remaining (i.e. third) fold \(F3\) with \(C_{2}^{2}\).
5. Perform steps (1) to (4) three times in total with the following substitution of folds: 1-2-3, 2-3-1, 3-1-2.

A support vector machine (SVM) was used as classifier \([4]\), with an RBF kernel with \(\gamma = 1/34\) (where 34 is the number of features, and a relatively low cost \(C = 1\), allowing for more training errors, but offering better generalization. Experiments for determining optimal parameter values are left for future work. The classification is performed at the level of instances, i.e. a decision is taken for each instance (or feature vector), recalling that each music piece is represented by 646 instances. Next, the final classification is based on class memberships of its 646 constituting instances via a simple majority voting scheme: a music piece is classified as the most voted class among its instances.

3. Experimental Results

The experiments are carried out on a subset of the Latin Music Database (LMD) \([10]\). The LMD is made

\(\text{Figure 1. Feature computation: (a) audio signal; (b) spectral centroid; (c) texture windows.}\)

\(\text{Figure 2. Training instance selection procedure (instance selection over only 1 fold represented).}\)
up of 3,227 full-length music pieces uniformly distributed along 10 classes (i.e. music genres: “Axé,” “Bachata,” “Bolero,” “Forró,” “Gaucho,” “Merengue,” “Pagode,” “Salsa,” “Sertaneja” and “Tango”). In our experiments we use 900 music pieces from the LMD, which are split into 3 folds of equal size (30 music pieces per class). The splitting is done using an artist filter [5], which places the music pieces of an specific artist exclusively in one, and only one, fold of the dataset. The use of the artist filter does not allow us to employ the whole dataset since the distribution of music pieces per artist is far from uniform. Furthermore, in our particular implementation of the artist filter we added the constraint of the same number of artists per fold.

We compare the accuracy obtained for all 10 music genres with the procedure above with that obtained with a standard 2-fold cross-validation procedure, without instance selection: 1 fold used for training a N-class classifier, 1 fold for testing, 3 permutations of the test fold (i.e. 2-3, 3-1, 1-2). The accuracy is computed as number of correctly classified instances divided by total number of relevant instances.

<table>
<thead>
<tr>
<th>Class</th>
<th>No selection</th>
<th>With selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axé</td>
<td>34.6% ± 8.6</td>
<td>35.7% ± 5.3</td>
</tr>
<tr>
<td>Bachata</td>
<td>54.5% ± 8.3</td>
<td>61.3% ± 11</td>
</tr>
<tr>
<td>Bolero</td>
<td>40.2% ± 3.3</td>
<td>46.0% ± 8.2</td>
</tr>
<tr>
<td>Forró</td>
<td>14.3% ± 5.4</td>
<td>14.0% ± 5.2</td>
</tr>
<tr>
<td>Gaucho</td>
<td>31.0% ± 10.3</td>
<td>33.4% ± 8.0</td>
</tr>
<tr>
<td>Merengue</td>
<td>50.1% ± 7.3</td>
<td>57.2% ± 16.4</td>
</tr>
<tr>
<td>Pagode</td>
<td>32.4% ± 5.6</td>
<td>34.6% ± 7.2</td>
</tr>
<tr>
<td>Salsa</td>
<td>24.5% ± 3.1</td>
<td>22.8% ± 6.7</td>
</tr>
<tr>
<td>Sertaneja</td>
<td>21.9% ± 5.7</td>
<td>26.9% ± 19.4</td>
</tr>
<tr>
<td>Tango</td>
<td>83.6% ± 4.8</td>
<td>85.7% ± 7.6</td>
</tr>
<tr>
<td>Average</td>
<td>38.7% ± 6.2</td>
<td>41.7% ± 2.6</td>
</tr>
</tbody>
</table>

Table 1. Percentage of correctly classified instances.

<table>
<thead>
<tr>
<th></th>
<th>No selection</th>
<th>With selection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>57.2% ± 12.5</td>
<td>59.6% ± 13.5</td>
</tr>
</tbody>
</table>

Table 2. Percentage of correctly classified music pieces.

Steps (1) and (2) performed on all 3 folds of the dataset result in an average of 61.7% of the instances being rejected from training. From an initial number of 193,800 training instances per fold, there is in step (3) an average of 75,036 training instances per fold when instance selection is performed.

Table 1 shows the percentage of correctly classified instances without vs. with instance selection, and the breakdown with respect to music genres. Table 2 shows the average percentage of correctly classified music pieces after majority vote over all instances of each piece. Note that these figures should be compared to a baseline classification of 10% (precision when always guessing the same random class).

For comparison, random resampling of training instances from 193,800 to 75,000 (keeping uniform distributions between classes) yields the following results: 38.4% ± 6.8 precision for instances, and 55.9% ± 12 precision for music pieces.

It is also interesting to look at the complexity of the classifiers in each case (measured here as the number of support vectors). When no instances are selected before training, the resulting classifier keeps over 150,000 instances as support vectors. With the process of training instance selection, the classifier complexity drops down to 17% of the original (i.e. around 25,000 support vectors), resulting in a much faster classification of test instances. Note also that the classifier complexity is only 35% of that of a classifier obtained with random resampling.

In summary, our results indicate that the proposed process of training instance selection yields a slight increase in precision, together with a less complex, and faster final classifier. Results also show that our proposed method for instance selection is better than random selection.

4. Discussion and future work

The first classification experiments using the LMD are reported in [9]. Features are extracted from three 30-second segments from the beginning, middle and end of each musical file. For each segment, feature vectors are computed with Marsyas. Classifiers are trained for each segment, and ensemble methods are used for classification. Ensemble methods outperform single classifier method (classification accuracy is shown to improve by 1 to 7 percentage points). It is also shown that middle segments provide the best classification accuracy. Different classifiers are used, published accuracies vary from 46% to 65%, the best results being achieved with an SVM classifier. It should be noted that these results are not directly comparable to those of this paper, as the evaluation methodology is different. Among other things, in [9] a 10-fold cross-validation is used, and, importantly, no artist filter is applied.
The LMD dataset is also being used in the yearly MIREX algorithm competition. Thirty three algorithms were evaluated in the latest contest, conducted in 2009. Evaluation was done via 3-fold cross-validation, accounting for an artist filter in the creation of the folds. Published results on this dataset range from 38.8% to 74.6% (the best classification accuracy to-date being obtained with a SVM classifier [3]). Even though our experiments are not strictly comparable to those in MIREX, it is still interesting to see that the results obtained with our method stands in the first third of algorithms.

We believe that the results obtained in this paper are encouraging. A number of avenues for future work are possible. First, we plan to experiment different procedures for selecting the best training instances. Further, we will also conduct experiments regarding the design of a “rejection” class, making use of those instances currently left aside from training. Indeed, there is probably useful information in those instances that show confusion among a large number of classes. We will explore different ways of constructing the “rejection” class, making use of a detailed analysis of initial misclassification. We will also extend experiments to different music databases and different signal features.

5. Conclusions

This paper reported on experiments in automatic music genre classification from audio signals. Instances to classify consist of feature vectors representing short-term, low-level, characteristics of music audio signals. The particularity of our approach lies in a process of instance selection prior to classifier training: i.e. we select from the training data those instances that show better discrimination with respect to class memberships. On a very challenging dataset of 900 music pieces divided among 10 music genres, our method assigns the correct genre to a given test music piece in 59.6% ± 13.5 of the cases. This represents a slight improvement with respect to the precision obtained without selection of training instances, while the size of the resulting classification model is significantly reduced, permitting much faster classification over test data.

6. Acknowledgments

Convénio FCT/CAPES-2009; QREN-Adl grant for Palco3.0 project; Luís Gustavo Martins (CITAR-UCP).

References