Using Smartphones to Classify Urban Sounds

Elsa Ferreira Gomes  
ISEP/IPP-School of Engineering, Polytechnic of Porto, Portugal  
efg@isep.ipp.pt

Fábio Batista  
ISEP/IPP-School of Engineering, Polytechnic of Porto, Portugal

Alípio M. Jorge  
IAAD-INESC TEC  
DCC-FCUP, Universidade do Porto, Portugal  
amjorge@fc.up.pt

ABSTRACT

The aim of this work is to develop an application for Android able to classifying urban sounds in a real life context. It also enables the collection and classification of new sounds. To train our classifier we use the UrbanSound8K dataset available online. We have used a hybrid approach to obtain features, by combining SAX-based multiresolution motif discovery with Mel-Frequency Cepstral Coefficients (MFCC). We also describe different configurations of motif discovery for defining attributes and compare the use of Random Forest and SVM algorithms on this kind of data.

Keywords

Urban sound classification, motif discovery, time series analysis, SAX, Random Forest, MFCC, mobile app.

1. INTRODUCTION

The automatic recognition of urban sounds can be a very useful tool for the detection of noise sources in an urban area. According to the World Health Organization (WHO) [1], noise pollution has a major negative impact on public health, which could cause, to whom is daily exposed, hearing loss, heart disease, sleep disorders, increased stress, among other problems. To effectively fight noise pollution in a city it is important to measure the volume level of sound and identify its origin [1, 2].

In this paper we present the design, implementation, and evaluation of the UrbanSound Classifier, a mobile client/server tool for urban sounds classification. With this tool, we can record a sound and classify it. We can also confirm or reject the automatic classification performed by the tool and add the labeled sound to the dataset. For the classification task we use a combination of features using SAX-based Multiresolution Motif Discovery [11] and Mel-Frequency Cepstral Coefficients (MFCC) [13] for Urban Sound Classification. For the training of the classifier, we have used UrbanSound8K, a dataset available online [3]. This dataset contains short audio snippets taken from sounds of the UrbanSound dataset also available online. In a preliminary work we have applied for the first time the motif based approach to urban sounds [17]. We compare the use of Support Vector Machines (SVM) [8] and Random Forest algorithms [9]. This study illustrates the potential of this application for urban sound classification.

2. THE CLASSIFICATION PROCESS

In this section we describe the two processes for extracting features from sounds and then learn a classifier from those features and the sound labels.

2.1 Feature extraction using MFCC

MFCCs are commonly used in environmental sound analysis and frequently used as a competitive baseline to benchmark novel techniques [16, 12, 14, 7, 22, 20, 23, 26, 25]. We implemented MFCCs in Java, from an existing version in Python, available online. We use frames with length 1024 (window size) with 50% of overlap. For each of these frames we extract 13 features, as follows:

1. The process starts by producing the Discrete Fourier Transform of each frame.
2. The Mel-spaced filterbank is calculated. We apply 40 triangular filters to the periodogram power spectral (the square of absolute value of the complex Fourier transform). Our filterbank comes in the form of 40 vectors. Each vector is mostly zeros, but is non-zero for a certain section of the spectrum. To calculate filterbank energies we multiply each filterbank with the power spectrum, then add up the coefficients. Thus, we have an indication of how much energy was in each filterbank.
3. Take the log of each of the 40 energy values obtained in step 2.
4. Take the Discrete Cosine Transform (DCT) of the 40 log filterbank energies to give 40 cepstral coefficients. Only the lower 13 of the 40 coefficients are kept.

5. Build a dataset with the resulting features (13 numbers for each frame). These features are called Mel Frequency Cepstral Coefficients. Then, we consider the average and the standard deviation of total frames as features [26].

6. Run a machine learning algorithm on the resulting dataset and estimate the predictive ability of the obtained classifier.

2.2 Multiresolution Motif Discovery

Frequent patterns (motifs) extracted from a time series database can be useful for a number of different applications [15], such as health and medicine. In particular, in EEG signal processing, discovered motifs may serve as markers for the proximity of a seizure [28]. One recent trend in time series analysis is to use SAX (Symbolic Aggregate Approximation) [21]. SAX is a symbolic approach for time series that represents the continuous series as a discretized one. It allows for dimensionality reduction and indexing. In classic data mining tasks such as clustering or classification, SAX performs as well or better-known representations such as Discrete Wavelet Transform (DWT) and Discrete Fourier Transform (DFT), while requiring less storage space. The representation allows researchers to avail of the wealth of data structures and algorithms in bioinformatics or text mining, and also provides solutions to data mining tasks, such as motif discovery [21].

We use MrMotif [11], a Multiresolution Motif Discovery in Time Series algorithm, to generate features. The aim of MrMotif algorithm is to find the top-K frequent motifs in a time series database D, given a motif length m and the value of K. This is done on each resolution (gmin, 2gmin, 2gmin×2, ..., gmax). In this work we will use MrMotif for generating motif-based features for urban sound classification. This algorithm is based on iSAX methodology to discretize the continuous signals. The iSAX methodology is a generalization of SAX that allows indexing and mining of massive datasets [21]. The main idea of the MrMotif algorithm is to start from a low iSAX resolution and then expand to higher resolutions. The minimum possible resolution gmin in iSAX is 2 and the maximum resolution gmax is assigned to 64 (it uses 2, 4, 8, 16, 32 and 64 resolutions).

In previous work, we used motifs as features in cardiac audio time series [18, 19]. The general idea is to find frequent motifs in the audio time series using a frequent pattern mining algorithm. Such discovered motifs are regarded as features. We have demonstrated that these features contain valuable information for discrimination tasks. To test this hypothesis, we first identify relevant motifs in the original dataset and build a new dataset where each relevant motif is an attribute. Then, we compare the results of using these features with the results obtained with the MFCCs features.

A motif in a time series is a frequent pattern, i.e. a repetition of a particular segment of the series. We use the Multiresolution Motif Discovery (in Time Series) algorithm [10] to detect the common (and relevant) patterns. This algorithm uses the iSAX methodology to discretize the continuous signals and looks for patterns in the resulting discrete sequences. In particular, we have used the MrMotif algorithm as implemented by its authors.

For our experimental exploration, we followed the steps below.

1. Apply to the original audio dataset the pre-processing steps (filters and normalized average Shannon energy) used in the previous approach.

2. Apply the MrMotif algorithm to the resulting time series. In this step we have to choose specific values for MrMotif’s parameters. We have tried different combinations.

3. Build a dataset of features with the most relevant motifs found in the previous step. The value of such feature is the frequency of the motif in the corresponding time series.

4. Run a machine learning algorithm on the resulting dataset and estimate the predictive ability of the obtained classifier.

The parameters of the MrMotif algorithm are the following: motif length m - the length of the sliding window that contains the section to discretize in the original time series; number of motifs generated K - these are the top-K relevant motifs for each resolution from 4 to 64; word size w - this is the number of discrete symbols of the iSAX word; and overlap o the extent to which two windows can overlap. In Figure 1 we can see an example of the processing of one audio from the dataset. The figure shows the starting location of motifs (identified by number) of length 40.

In Table 2, a sample of one of the datasets is shown. Each attribute M is a top-10 motif of resolution 4. Values are frequencies of motifs. The last column is class (CIH, Car Horn and DB, Dog Bark).

3. EXPERIMENTS

In this section we describe the experimental efforts for selecting the most appropriate algorithm for sound classification. We have considered several classification algorithms available in Weka data mining software [27]. For each experiment, we report the average accuracy using 10-fold cross validation procedure. In this paper we show results for the two classifiers that achieved best results: Random Forest and SVM.

The dataset we have used in this study is UrbanSound8K [3, 24] and it consists of 8732 labeled audio files approximately 4 seconds long (table 1). Each recording is labeled...
with the start and end times of sound events from 10 classes: air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gun_shot, jackhammer, siren and street_music. For each file, only events from a single class are labeled [24]. The UrbanSound8K dataset, available online, is a subset of short audio snippets of the UrbanSound dataset also available online. In previous work, we have done experiments with this dataset [17].

<table>
<thead>
<tr>
<th>Class</th>
<th>number</th>
</tr>
</thead>
<tbody>
<tr>
<td>air_conditioner</td>
<td>974</td>
</tr>
<tr>
<td>car_horn</td>
<td>429</td>
</tr>
<tr>
<td>children_playing</td>
<td>1000</td>
</tr>
<tr>
<td>dog_bark</td>
<td>999</td>
</tr>
<tr>
<td>drilling</td>
<td>978</td>
</tr>
<tr>
<td>engine_idling</td>
<td>1000</td>
</tr>
<tr>
<td>gun_shot</td>
<td>374</td>
</tr>
<tr>
<td>jackhammer</td>
<td>1000</td>
</tr>
<tr>
<td>siren</td>
<td>920</td>
</tr>
<tr>
<td>street_music</td>
<td>1027</td>
</tr>
</tbody>
</table>

Table 1: Class distribution

<table>
<thead>
<tr>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>CH</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>CH</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>BB</td>
</tr>
</tbody>
</table>

Table 2: Sample of a resulting dataset of resolution 4, Top-10.

Classifiers were evaluated using 10 fold cross validation with the Weka data mining suite [27]. We proceed by describing the conducted experiments.

3.1 Using motifs and MFCCs features

In this first set of experiments we have used MrMotif to obtain the features. We varied the parameter \( K \) corresponding to the number of motifs selected for attributes. The motif resolution \( K \) was fixed with value 4. For window size we have used 20, and for overlap 10. The size of the SAX word is 8 symbols. In the experiments described below these are the default values.

In the second set of experiments we have used the MFCCs approach to obtain the features for classification. In Figure 2 and Figure 3 we can see the results obtained for the two sets of experiments, using these classifiers to separate all the classes and each pair of classes respectively.

As we can see, in both experiments, MFCC features reveal better discriminant power. In the first set of experiments accuracy was 55.68% using MFCCs and 26.45% using motifs. In the second set of experiments it was 85% using MFCCs and 70.55% using motifs. We can also see that the best results were achieved with Random Forest algorithm. The proportion of the most popular class (street_music) is 11.7%.

3.2 Classes with similar timbre

In general, we have obtained better results using the features generated with MFCC. However, in the case of pairs of classes with very similar timbre, the Motif based approach obtained good results. The pairs of classes are: air_conditioner + engine_idling, drilling + jackhammer, children_playing + street_music, siren + street_music. The problem of correctly separating these classes had already been identified in the literature [24] and explored in our previous work [17]. In the following we describe new experiments on these challenging pairs of classes for the UrbanSound8K dataset.

![Figure 2: Classification results for all classes (Motif vs MFCC)](image1)

![Figure 3: Classification results for each pair of classes (Motif vs MFCC)](image2)

![Figure 4: Motif and MFCCs classification for each pair of the four classes](image3)
and the number of attributes sampled for each model (in the tables we use the letter K for this parameter as originally used by the random forest weka implementation, albeit the collision with the motif K parameter). In this experiments we used I = 200 and K = 5.

In Table 3 we present the best results for MFCC (MF), MFCC + MotifsR4W1005 (Mo1), MFCC + MotifsR4W20010 (Mo2), MFCC + MotifsR8W1005 (Mo3)). As we can see, we achieved the best accuracy results combining features obtaining for siren + street_music, 91.06%, for drilling + jackhammer, 72.65% and for air conditioner + engine idling, 68.53% for each pair of the four classes. However, for the pair children_playing + street_music, we obtained an accuracy of 79.25% using combined features, slightly lower than the accuracy achieved using only MFCCs features (79.7%). Performing a Wilcoxon signed rank test (on the paired size 10 samples generated by cross-validation) we statistically validate that MFCC + MotifsR4W1005 is superior to MFCC alone for the pairs drilling + jackhammer and siren + street music, for a confidence level of α=0.05 (p-values are 0.02 and 0.03, respectively). For the other two pairs of classes the hypothesis of equality wasn’t rejected.

### 3.3 Combining features for all classes

In the third set of experiments we assess the value of combining MFCC features with motif based features for the whole set of classes. In Figure 5 and Figure 6 we can see the results obtained in the task of separating all the classes and the average result in the tasks of discriminating each pair of classes. As we can see, in both experiments, the best results were achieved with the Random Forest algorithm. We can also see that in our experiments the results achieved by combining features were better than the results achieved using the MFCC features. In the first set of experiments the accuracy obtained was 55.68% using MFCCs and 56.87% using combined features. In the second set of experiments it was 81% using MFCCs and 85.64% using combined features.

Performing a Wilcoxon signed rank test we statistically validate that the average accuracy for all pairs of classes achieved by MFCC+Motifs is superior to MFCC, for a confidence level of α=0.01 (p-value is 0.05). In the case of all classes, although the average values are higher, the hypothesis of equality wasn’t rejected.

Given these results, we conclude that, to deploy in the application, we should use Random Forest to generate the classification model for identifying a given sound of any of the 10 classes. As features we use a combination of MFCC and motifs.

### 4. THE APPLICATION

The UrbanSound Classifier tool has 2 components: the client (mobile application) and the server (Java application).
5. EVALUATION OF THE APPLICATION

We have done a small practical experiment to evaluate the performance of the tool on real sounds. The aim was to test the classification model in real life conditions and the server's response time. We played loud 20 sounds from the training set and classified them using the mobile application. The application was installed in a smartphone Sony Xperia L and the server component was installed in a desktop with operating system Windows 10. The components were connected by wireless local network.

In table 4 we can see the results obtained. In the second column we have the true class of the sound, in the third column we have the class obtained by the application and in the fourth column the response time. The response time depends mainly on the duration and sound quality of connection. For these tests, sounds were played during between 2 to 5 seconds.

As we can see, the application correctly classified 11 out of 20 sounds. The execution times were around 1 and 3 secs.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a mobile application, the UrbanSound Classifier, that can provide support to researchers in the field of urban sounds. Using this app we can record and save new sounds, obtain the classification of the new sound and give feedback about the result obtained. It is also possible add these new sounds to the dataset and update the classification model. To characterize the sounds for classification, we combined features from an MFCC based approach with features from a motifs based approach. To learn the classifiers, we used the Random Forest algorithm.
As future work we intend to use the UrbanSound Classifier tool to expand the dataset and improve the classification process of the urban sounds. The application can also be used to build new sound datasets.

Acknowledgment

This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project “POCI-01-0145-FEDER-006961”, and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EIA/50014/2013.

7. REFERENCES


