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
LIAAD-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 data set 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. We show that motifs contain valuable information that can be further exploited for Urban Sound Classification especially when combined with MFCCs.

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 extration 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 on [4]. We use frames with length 1024 (window size) with 50% of overleap. For each of this 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.

C3S2E '16, July 20-22, 2016, Porto, Portugal
Copyright 2016 ACM 978-1-4503-4075-5/16/07 ...$15.00.
http://dx.doi.org/10.1145/2948992.2949015.

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 as other well-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 or each resolution in (g_{min}, $g_{min}\times 2$, ...,g_{max}). 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 g_{min} in iSAX is 2 and the maximum resolution g_{max} 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.

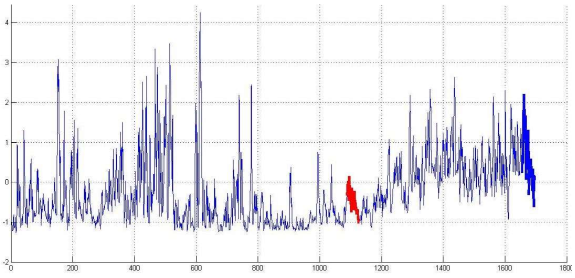


Figure 1: Motif with 2 repetitions of length 40 for one audio file

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_i is a top-10 motif of resolution 4. Values are frequencies of motifs. The last column is class (CH, 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]. 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].

Class	number
air_conditioner	974
car_horn	429
children_playing	1000
dog_bark	999
drilling	978
enginge_idling	1000
gun_shot	374
jackhammer	1000
siren	920
street_music	1027

Table 1: Class distribution

M1	M2	M3	M4	M5	M6	Class
1	0	5	2	0	2	CH
2	2	7	4	3	1	CH
5	4	5	7	5	5	DB

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 R 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.

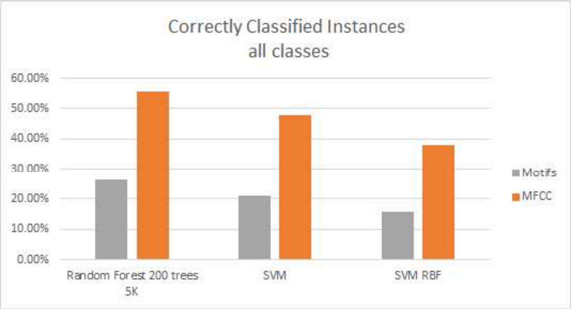


Figure 2: Classification results for all classes (Motif vs MFCC)

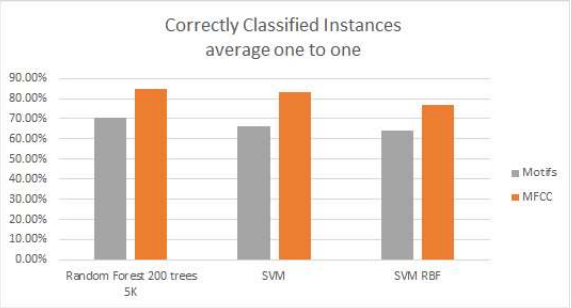


Figure 3: Classification results for each pair of classes (Motif vs MFCC)

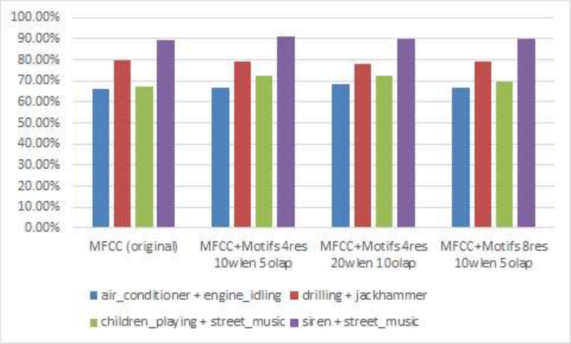


Figure 4: Motif and MFCCs classification for each pair of the four classes

In these experiments we varied the motif parameters and used random forests [9] only. In previous experiments best results have been achieved the with this classifier. This algorithm builds multiple decision trees with different attribute sets sampled from the original dataset. This results in complementary models that together have an effect similar to a probabilistic disjunctive model. The important parameters for the random forest algorithm are the number of models I

Table 3: Compariing the MFCCs approach with the combination of MFCCs and Motifs (varying number of motifs) using Random Forest.

Classes	MF	MF+Mo1	MF+Mo2	MF+Mo3
air_c+eng_id	66.38	66.78	**68.53**	66.67
ch_pl+str_m	**79.7**	79.00	78.30	**79.25**
drill+jckhm	67.12	**72.65**	72.14	69.27
sir+str_m	89.07	**91.06**	89.73	89.75

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 + MotifsR4W10O5 (Mo1), MFCC + MotifsR4W20O10 (Mo2), MFCC + MotifsR8W10O5(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 + MotifsR4W10O5 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 56.37% using combined features. In the second set of experiments it was 85% 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=0.1 (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).

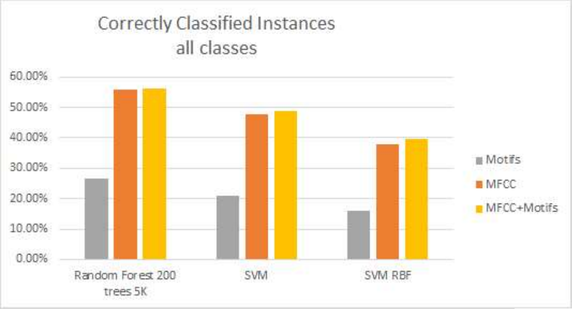


Figure 5: Classification results for all classes (Motif vs MFCC vs Motif+MFCC)

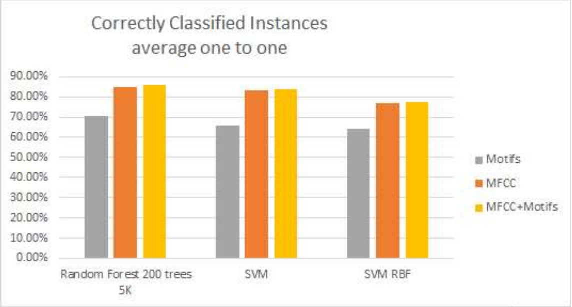


Figure 6: Classification results for each pair of classes (Motif vs MFCC vs Motif+MFCC)

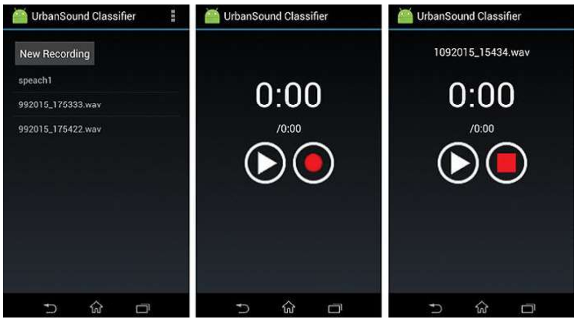


Figure 7: Steps to save a new sound using the UrbanSound Classifier

The main features are:

1. To capture, save and maintain sounds with the mobile device. In Figure 7 we can see the steps to create a new recording. In Figure 8 we can see the steps to reproduce, remove and eliminate a sound file to the UrbanSound Classifier. It is possible to add records manually by adding a sound file to a folder on the external storage, created by the application on the first run.

2. To classify the sounds giving a feedback to expand the existing dataset.

To classify an added sound that has not yet been classified, just choose the sound to classify in the main screen and

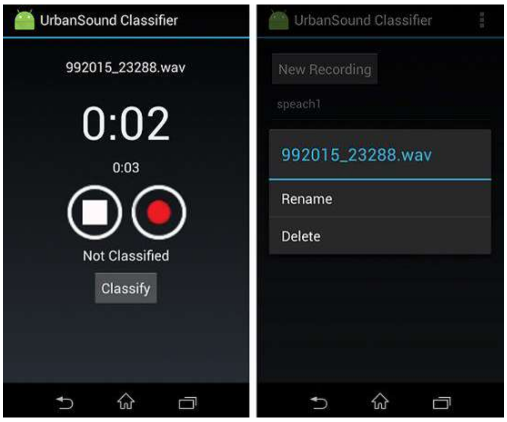
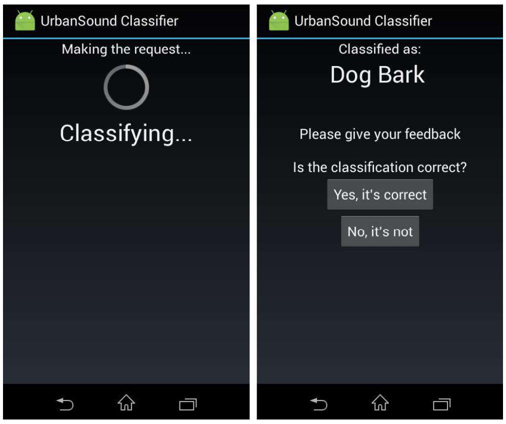


Figure 8: Reproduce, remove and remove a sound file at UrbanSound Classifier



Making the request...

Please give your feedback

Is the classification correct?

Figure 9: Classify and give feedback at Urban Sound Classifier

click the "Classify" button on the next screen (see Figure 9). The sound is then sent to the server and the returned result will appear on the screen. The user can then provide feedback indicating whether the classification is correct or not (choosing the right button). If the user selects the button "No, it's not" a new menu appears on the screen where the correct class must be selected (Figure 9). The server, receiving the feedback, adds the new sound to the same dataset associating the class indicated on the feedback.

The communication between the components client/server is assured by a web service that uses the SOAP protocol, it uses XML message format and HTTP protocol for the transmission (Figure 10). The Server component was implemented in Java. The client component was implemented with IDE Android Studio in Java. Information concerning to the recording of sounds are saved in a database implemented through DBAdapter and DBHelper classes using the SQLite technology [5]. The communication with the server is done using the ksoap2-android library [6].

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.

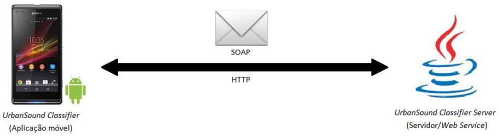


Figure 10: System Architecture

Table 4: Performance test to the Urban Sound Classifier

Nr	Real class	Obtained class	Time (ms)
1	air_conditioner	jackhammer	2440
2	air_conditioner	jackhammer	2259
3	car_horn	car_horn	2176
4	car_horn	car_horn	2117
5	children_playing	drilling	2496
6	children_playing	children_playing	2312
7	dog_bark	dog_bark	2306
8	dog_bark	drilling	3370
9	drilling	jackhammer	2457
10	drilling	children_playing	2563
11	engine_idling	engine_idling	3059
12	engine_idling	gun_shot	2342
13	gun_shot	gun_shot	2155
14	gun_shot	gun_shot	920
15	jackhammer	jackhammer	2575
16	jackhammer	drilling	2757
17	siren	siren	2454
18	siren	siren	2481
19	street_music	jackhammer	2442
20	street_music	street_music	2547

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/EEA/50014/2013.

7. REFERENCES

[1] http://www.euro.who.int/en/health-topics/environment-and-health/noise/data-and-statistics, 2011.

[2] http://publish.illinois.edu/audioanalytics/, 2015.

[3] https://serv.cusp.nyu.edu/projects/urbansounddataset, 2015.

[4] https://github.com/jameslyons/python_speech_features, 2015.

[5] https://www.sqlite.org/, 2015.

[6] http://simpligility.github.io/ksoap2-android/index.html, 2015.

[7] F. Beritelli and R. Grasso. A pattern recognition system for environmental sound classification based on mfccs and neural networks. In *Signal Processing and Communication Systems, 2008. ICSPCS 2008. 2nd International Conference on*, pages 1–4, Dec 2008.

[8] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT'92)*, pages 144–152, Pittsburgh, PA, USA, July 1992. ACM Press.

[9] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[10] N. Castro. Multiresolution motif discovery in time series website. http://www.di.uminho.pt/ castro/mrmotif.

[11] N. Castro and P. J. Azevedo. Multiresolution Motif Discovery in Time Series. In *SDM*, pages 665–676, 2010.

[12] S. Chaudhuri and B. Raj. Unsupervised hierarchical structure induction for deeper semantic analysis of audio. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 833–837, May 2013.

[13] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(4):357–366, Aug 1980.

[14] D. Ellis, X. Zeng, and J. McDermott. Classifying soundtracks with audio texture features. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5880–5883, May 2011.

[15] P. G. Ferreira, P. J. Azevedo, C. G. Silva, and R. M. M. Brito. Mining approximate motifs in time series. In *Discovery Science*, pages 89–101, 2006.

[16] J. Geiger, B. Schuller, and G. Rigoll. Large-scale audio feature extraction and svm for acoustic scene classification. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*, pages 1–4, Oct 2013.

[17] E. F. Gomes and F. Batista. Using multiresolution time series motifs to classify urban sounds. *International Journal of Software Engineering and Its Applications*, 9(8):189–196, 2015.

[18] E. F. Gomes, A. M. Jorge, and P. J. Azevedo. Classifying heart sounds using multiresolution time series motifs: an exploratory study. In *Proceedings of the International C* Conference on Computer Science and Software Engineering*, pages 23–30. ACM, 2013.

[19] E. F. Gomes, A. M. Jorge, and P. J. Azevedo. Classifying heart sounds using sax motifs, random forests and text mining techniques. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, pages 334–337. ACM, 2014.

[20] T. Heittola, A. Mesaros, T. Virtanen, and A. Eronen. Sound event detection in multisource environments using source separation. *Proc CHiME*, pages 36–40, 2011.

[21] J. Lin, E. Keogh, S. Lonardi, and P. Patel. Finding motifs in time series. In *Proceedings of the 2nd Workshop on Temporal Data Mining*, pages 53–68, 2002.

[22] S. Ntalampiras, I. Potamitis, and N. Fakotakis. Automatic recognition of urban soundscenes. In *New Directions in Intelligent Interactive Multimedia*, pages 147–153. Springer, 2008.

[23] G. Roma, W. Nogueira, P. Herrera, and R. de Boronat. Recurrence quantification analysis features for auditory scene classification. *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, 2013.

[24] J. Salamon, C. Jacoby, and J. P. Bello. A dataset and taxonomy for urban sound research. In *22st ACM International Conference on Multimedia (ACM-MM'14)*, Orlando, FL, USA, Nov. 2014.

[25] B. Uzkent, B. D. Barkana, and H. Cevikalp. Non-speech environmental sound classification using svms with a new set of features. *International Journal of Innovative Computing, Information and Control*, 8(5B):3511–3524, 2012.

[26] X. Valero and F. Alías. Hierarchical classification of environmental noise sources considering the acoustic signature of vehicle pass-bys. *Archives of Acoustics*, 37(4):423–434, 2012.

[27] I. H. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques. 2005.

[28] D. Yankov, E. J. Keogh, J. Medina, B. Y. chi Chiu, and V. B. Zordan. Detecting time series motifs under uniform scaling. In P. Berkhin, R. Caruana, and X. Wu, editors, *KDD*, pages 844–853. ACM, 2007.