UNSUPERVISED FEATURE LEARNING FOR URBAN SOUND CLASSIFICATION

Justin Salamon[1,2] and Juan Pablo Bello[2]

[1]Center for Urban Science and Progress, New York University, USA
[2]Music and Audio Research Laboratory, New York University, USA
{justin.salamon, jpbello}@nyu.edu

ABSTRACT

Recent studies have demonstrated the potential of unsupervised feature learning for sound classification. In this paper we further explore the application of the spherical k-means algorithm for feature learning from audio signals, here in the domain of urban sound classification. Spherical k-means is a relatively simple technique that has recently been shown to be competitive with other more complex and time consuming approaches. We study how different parts of the processing pipeline influence performance, taking into account the specificities of the urban sonic environment. We evaluate our approach on the largest public dataset of urban sound sources available for research, and compare it to a baseline system based on MFCCs. We show that feature learning can outperform the baseline approach by configuring it to capture the temporal dynamics of urban sources. The results are complemented with error analysis and some proposals for future research.

Index Terms— Unsupervised learning, sound classification, machine learning, urban, spherical k-means

1. INTRODUCTION

The automatic classification of sonic events in an urban setting has a variety of applications including context aware computing [1], surveillance [2], or the adaptation of content-based multimedia retrieval techniques such as highlight extraction [3] and video summarization [4] to urban applications (e.g. identifying important city-wide events) . Importantly, it also has the potential of improving the quality of life of city dwellers by providing a data-driven understanding of urban sound and noise patterns, partly enabled by the move towards "smart cities" equipped with multimedia sensor networks [5].

While there is a large body of research on sound classification in related areas such as speech, music and bioacoustics, work on the analysis of urban acoustic environments is relatively scarce. When existent, it mostly focuses on the classification of auditory scene type (e.g. street, park) [1, 6, 7, 8], as opposed to the identification of specific sound sources in those scenes such as a car horn, an engine idling or a bird tweet. The latter is a challenging task given the presence of multiple, often simultaneous, sources with very different mechanisms of sound production. Furthermore, those sources can, and likely will, be masked by noise, with many sources of interest, such as air conditioners or engine sounds, fairly noise-like themselves. In addition, urban auditory scenes can represent an almost infinite variety of configurations, lacking the high-level structure observed in other domains such as speech and music.

This work was supported by a seed grant from New York University's Center for Urban Science and Progress (CUSP).

Most previous work on environmental sound source classification relies on traditional, hand-crafted features [2, 9, 10] such as the tried and tested Mel-Frequency Cepstral Coefficients (MFCCs) [11] which have been shown to be sensitive to the type of background noise found in an urban environment [12]. Recent studies in audio classification have shown that accuracy can be boosted by using features that are learned from the audio signal in an unsupervised manner, with examples in the areas of bioacoustics [13] and music information retrieval [14, 15, 16]. Unsupervised feature learning has also been studied in the context of environmental sound classification [8, 17], though the focus has been on auditory scene classification, and not necessarily in an urban setting.

In this paper we explore the application of the *spherical k-means* algorithm [18] as an unsupervised feature learning technique for the classification of urban sonic events. In particular, we investigate learning features that capture the temporal dynamics of different sound sources. Whilst temporal dynamics were shown not to be an important factor in other domains such as birdsong classification [13], we show they play a key role in classifying urban sound sources whose instantaneous noise-like characteristics can be hard to distinguish in the absence of temporal context. We base our study on a dataset of field recordings [19] which is currently the largest public dataset of labelled urban sound events available for research.

The structure of the remainder of the paper is as follows: in Section 2 we describe the learning based classification approach studied in this paper. In section 3 we outline our experimental design, including dataset, evaluation measures and the variants of the proposed system we evaluate. In Section 4 we present and discuss our results, and finally we provide a summary of the paper and some directions for future work in Section 5.

2. METHOD

Our proposed feature learning and classification approach is comprised of three main processing blocks: preprocessing, feature learning and classification, where in this paper we focus on the first two. The key idea is to learn a codebook (or dictionary) of representative codewords (or bases) from the training data in an unsupervised manner. Samples are then encoded against this codebook and the resulting code vector is used as a feature vector for training / testing the classifier. In the following subsections we describe each block in further detail.

2.1. Preprocessing

As noted in [13], the raw audio signal is not suitable as direct input to a classifier due to its extremely high dimensionality and the fact that it would be unlikely for perceptually similar sounds to be neighbours in vector space. Thus, a popular approach for feature learning from

audio is to convert the signal into a time-frequency representation, a common choice being the mel-spectrogram. We extract log-scaled mel-spectrograms with 40 components (bands) covering the audible frequency range (0-22050 Hz), using a window size of 23 ms (1024 samples at 44.1 kHz) and a hop size of the same duration. We also experimented with a larger numbers of bands (128), but this did not improve performance and hence we stuck to the lower (and faster to process) resolution of 40 bands. To extract the mel-spectrograms we use the Essentia audio analysis library [20] via its Python bindings. Whilst we could use the resulting log-mel-spectrograms directly as input for the feature learning, it has been shown that the learned features can be significantly improved by decorrelating the input dimensions using e.g. ZCA or PCA whitening [18]. Following [14], we apply PCA whitening keeping enough components to explain 99% of the variance. The resulting representation is then passed to the feature learning block.

It is important to note that we can apply the feature learning either to individual frames of the log-mel-spectrograms, or alternatively to several consecutive frames resulting in 2D patches. Grouping several consecutive frames (by concatenating them into a single larger vector prior to PCA whitening), also known as *shingling*, allows us to learn features that take into account temporal dynamics. This option is particularly interesting for urban noise-like sounds such as idling engines or jackhammers, where the temporal dynamics could potentially improve our ability to distinguish sounds whose instantaneous features (i.e. a single frame) can be very similar.

2.2. Feature Learning

2.2.1. Spherical k-means

For learning features from the whitened log-mel-spectrograms we use the *spherical k-means* algorithm [18]. The main difference between this algorithm and variants of the widely-used k-means clustering algorithm [21] is that the centroids are constrained to have unit L2 norm (they must lie on the unit sphere), the benefits of which are discussed in [18, 22]. When used for feature learning, our goal is to learn an over complete codebook, so k is typically much larger than the dimensionality of the input data. The algorithm has been shown to be competitive with more advanced (and much slower) techniques such as sparse coding, and has been used successfully to learn features from audio for both music [14] and birdsong [13]. Here we study its utility for learning features from urban sound recordings.

Let us represent our data as a matrix $X \in \mathcal{R}^{n \times m}$, where every column vector $x^{(i)} \in \mathcal{R}^n$ is the feature vector for a single sample (in our case a whitened log-mel-spectrogram frame or patch), n is the number of dimensions and $i = 1 \dots m$ where m is the total number of samples. We use $s^{(i)}$ to denote the code vector for sample i which stores an assignment value for each of our k clusters. For convenience, let S be the matrix whose columns are $s^{(i)}$. Finally, let $\mathcal{D} \in \mathcal{R}^{n \times k}$ represent our codebook of k vectors (means). Then, the spherical k-means algorithm can be implemented by looping over the following three equations until convergence:

$$s_j^{(i)} := \begin{cases} \mathcal{D}^{(j)\top} x^{(i)} & \text{if } j == \underset{l}{\operatorname{argmax}} |\mathcal{D}^{(l)\top} x^{(i)}|_{\forall j,i} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

$$\mathcal{D} := XS^\top + \mathcal{D} \quad (2)$$

$$\mathcal{D}^{(j)} := \mathcal{D}^{(j)} / ||\mathcal{D}^{(j)}||_2 \forall j \quad (3)$$

where $\top$ indicates matrix transposition. In Equation (1) we assign samples to centroids, in (2) we update the centroids, and finally in (3) we normalize the centroids to have unit L2 norm. Before running the algorithm we randomly initialize the centroids in the codebook $\mathcal{D}$ from a Normal distribution and normalize them as in (3). For further details about the algorithm the reader is referred to [18].

2.2.2. Encoding

The learned codebook is used to encode the samples presented to the classifier (both for training and testing). A possible encoding scheme is vector quantization, i.e. assign each sample to its closest (or n closest for some choice of n) centroids in the codebook, resulting in a binary feature vector whose only non-zero elements are the n selected neighbours. While this approach has been shown to work for music [16], in our experiments we found that a linear encoding scheme where each sample is represented by its multiplication with the codebook matrix provides better results, in accordance with [14].

2.2.3. Pooling

After encoding, every audio recording is represented as a series of encoded frames (or patches) over time. For classification, we have to summarize over the time axis so that the dimensionality of all samples is the same (and not too large). Different studies report success using different summary (or pooling) statistics such as maximum [14], mean and standard deviation [13] or a combination of a larger number of statistics such as minimum, maximum, mean and variance [15]. In our experiments we use the mean and standard deviation, which we found to be the best performing combination of two pooling functions. The final representation is of size k times the number of pooling functions, and since k is already large (e.g. [13] and [14] use $k = 500$), it is in our interest to keep the number of pooling functions small. The resulting feature vectors are standardized (across samples) prior to classification.

2.2.4. Class-conditional codebook learning

Rather than learning the codebook $\mathcal{D}$ from all the training data combined, we can also separate the training data by class and learn a separate smaller codebook for each class. For instance, instead of one codebook with $k = 2000$ we learn 10 codebooks with $k = 200$. Finally we take the union of the codebooks as the final codebook used for encoding features. This approach has two potential advantages: first, it could help in learning class-specific features as opposed to features common to all classes that might be less predictive or even represent noise. Second, it allows us to visualize the features by the class from which they were learned, which is helpful for performing a visual sanity check on the learned features.

2.3. Classification

Since our focus is on the feature learning stage, we use a single classification algorithm for all experiments – a random forest classifier [23] (500 trees). This classifier was used successfully in combination with learned features in [13], and was also one of the top performing classifiers for a baseline system [19] evaluated on the same dataset used in this study (cf. Section 3.1). For our experiments we use the implementation provided in the *scikit-learn* Python library [24].

3. EXPERIMENTAL DESIGN

3.1. Dataset and Metrics

For evaluation we use the UrbanSound8K dataset [19]. The dataset is comprised of 8732 slices (excerpts) of up to 4 s in duration extracted

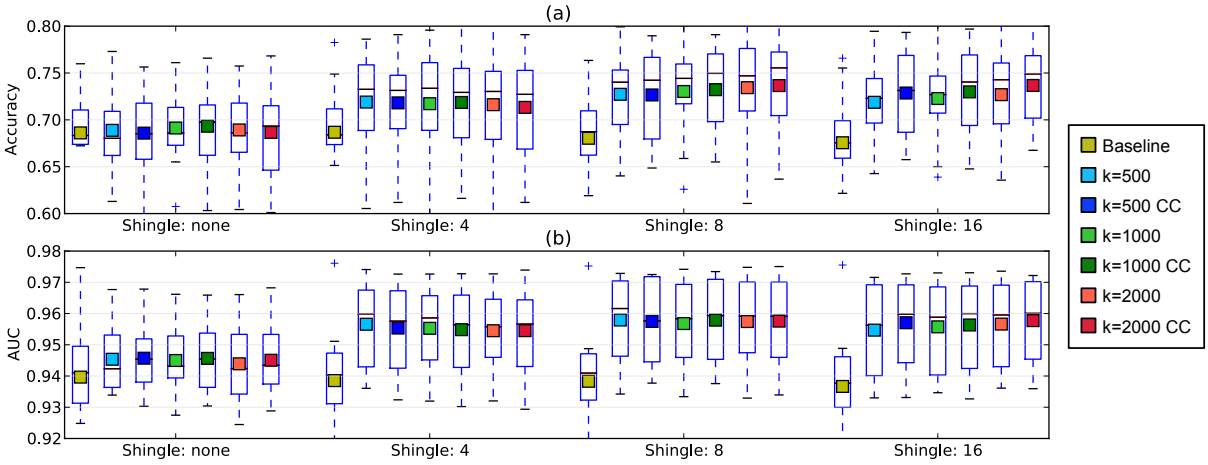

Fig. 1: Classification accuracy (a) and AUC (b) as a function of shingling, k and class-conditional (CC) codebook learning.

from field recordings crawled from the Freesound online archive[1]. Each slice contains one of 10 possible sound sources: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, street music. The sources were selected from the Urban Sound Taxonomy [19] based on the high frequency with which they appear in noise complaints as determined from the data provided by New York City's 311 service (over 370,000 complaints from 2010 to date)[2]. Since these are real field-recordings, it is possible (and often the case) for there to be other sources present in a slice in addition to the labeled source. All slices have been manually annotated with the source ID and a subjective judgement of whether the source is in the foreground or background.

To facilitate comparable research, the slices in UrbanSound8K come pre-sorted into 10 folds using a stratified approach which ensures that slices from the same recording will not be used both for training and testing, which could potentially lead to artificially high results. For every experiment we run a 10-fold cross validation using the provided stratified folds. For each fold we compute two widely used evaluation metrics: classification accuracy and Area Under the ROC Curve (AUC) [25]. Finally the result is presented as a box plot generated from the 10 per-fold scores. To gain further insight we also plot the confusion matrices for some of the experiments.

3.2. System Variants

In this study we explore two of the key parameters that can affect system performance: the size of the codebook (number of learned centroids) $k = 500, 1000, 2000$, and the number of frames we shingle together prior to learning $N_{\text{shingle}} = 1, 4, 8, 16$. When $N_{\text{shingle}} = 1$ no shingling is applied and we learn 1-dimensional features from single frames. For all other values we learn 2-dimensional features which incorporate temporal dynamics. In addition to these two parameters, we also explore the effect of class-conditional codebook learning (Section 2.2.4) by evaluating two codebooks for each combination of parameters, one learned from all the training data combined and one learned using class-conditional learning. For the latter we set the size of each per-class codebook to k divided by the number of classes (10), such that the final size of the codebook is equal to that of the one learned from all the training data combined. Altogether this gives us $3 \times 4 \times 2 = 24$ experimental configurations to compare.

In addition to the 24 configurations proposed in this study we also compare the results to the non-learning baseline system described in [19] which computes 25 MFCC coefficients per frame and summarizes each coefficient over time using 11 summary statistics. We also evaluate 3 extensions of this baseline approach where we apply frame shingling (4, 8 and 16) to assess the influence of shingling alone, without unsupervised learning.

For completeness, we briefly mention some of the variants experimented with whose results are not reported here: the number of Mel bands (128 instead of 40), the encoding scheme (n-Hot with $n = 1, 8, 16$ instead of a linear scheme) and other pooling functions (e.g. maximum). As noted in Section 2, none of these alternatives provided any improvement over the results reported in this study.

4. RESULTS AND DISCUSSION

The classification accuracy results for the 24 proposed configurations and the baseline approach [19] are presented in Figure 1(a). Each group of 7 box plots represents a different shingling strategy: no shingling, 4 frames, 8 frames and 16 frames. The means are indicated by the filled squares. The legend indicates the codebook size k and whether class-conditional (CC) codebook learning was applied.

We see that all proposed configurations perform at least as well as the baseline approach. When we learn our codebook from single frames (no shingling), we see the proposed feature learning performs comparably but does not outperform the baseline approach. However, once we group several consecutive frames into 2D patches (shingling) and apply the feature learning on those, the learned features outperform the baseline. Indeed, for the best performing configurations (patches of 8 or 16 frames with a class-conditional codebook of size $k = 2000$) we obtained a 5% accuracy improvement over the baseline (statistically significant according to a paired t-test with $p < 0.05$). Incorporating temporal context through shingling illustrates the importance of short-term temporal structure (beyond delta MFCCs) for the characterization of sources that are common to urban environments. Furthermore, we see that this gain in performance does not occur when we shingle the baseline features, confirming that the improvement comes from the combination of shingling and feature learning. This stands in contrast to birdsong classification for instance, where the authors did not observe a similar

[1] http://www.freesound.org
[2] https://nycopendata.socrata.com/data

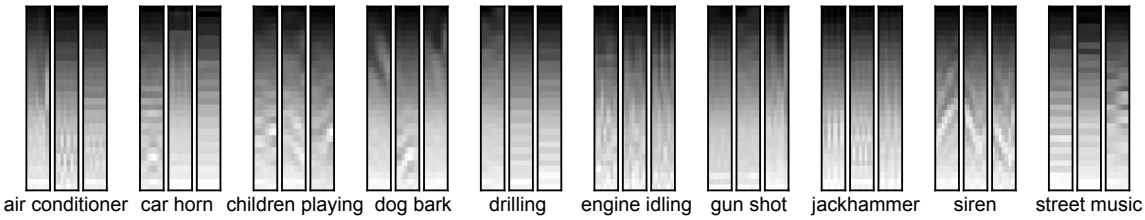

Fig. 2: Features learned using class-conditional spherical k-means applied to 8-frame patches of PCA whitened log-mel-spectrograms.

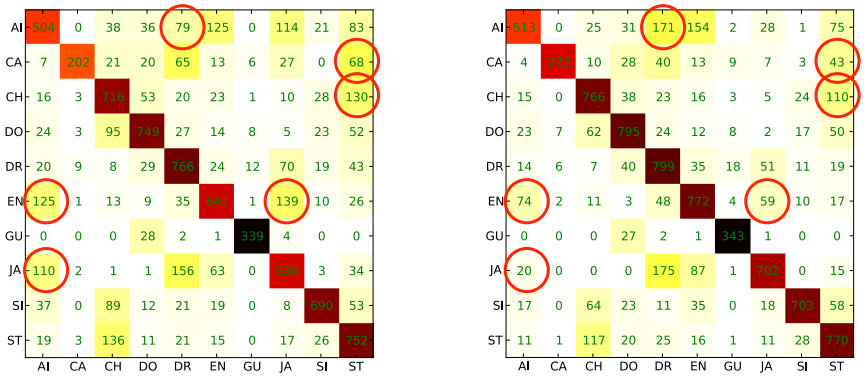

Fig. 3: Confusion matrices for the baseline (left) and proposed feature learning approach (right). Classes: air conditioner (AI), car horn (CA), children playing (CH), dog bark (DO), drilling (DR), engine idling (EN), gun shot (GU), jackhammer (JA), siren (SI), street music (ST).

improvement from frame shingling [13]. The downside to our solution however is that we need a separate basis (codeword) to encode every phase shift within a shingle, thus requiring a larger codebook (larger k). Indeed, the evaluation shows that for patches of 8 and 16 frames the accuracy increases monotonically with codebook size.

The AUC results for the same set of experiments are presented in Figure 1(b). Whilst the influence of codebook size is less clear here, overall the results are consistent with those discussed above: the classifier trained with learned features is more stable than the baseline, most so when learning patches of 8 or 16 consecutive frames.

Environmental recordings can be dominated by background noise, which motivated us to experiment with class-conditional codebook learning. Our underlying assumption was that in this way we might avoid learning noise codewords (that are common to all classes) and instead learn codewords that are truly indicative of each sound source. From Figure 1(a) we see that whilst it does improve classification accuracy (in particular for patches of 16 frames), the improvement over learning a single global codebook is marginal. The difference might be more significant for unbalanced datasets, but such a test is beyond the scope of this paper. Still, class-conditional learning allows us to inspect the features learned for the different classes. In Figure 2 we provide 3 examples of features learned for each class using class-conditional codebook learning on patches of 8 frames. Whilst it is not the case for all classes, for some it is straightforward to provide a qualitative interpretation: for car horns we learn stationary harmonic series, for children playing we learn different patterns of human speech, jackhammer features display a rapidly alternating pattern and siren features clearly show increasing and decreasing harmonic tones.

Finally, to gain further insight into the classification errors of the proposed approach and how they compare to the baseline, in Figure 3 we provide the confusion matrices for the baseline (left) and best performing learning-based approach (right). Confusion values of interest in both matrices are highlighted with red circles. We see that the confusion is reduced for all classes when using the learned features compared to the baseline. In particular, the most dramatic improvement is for the engine idling and jackhammer classes, both of which had high confusion with the air conditioner class in the baseline approach. A possible explanation is that whilst the three classes may have similar statistics when summarized over time, they actually have distinct short-term temporal patterns, and thus by learning 2D patches the confusion between the sources is reduced. Some improvement is also noted for more harmonic classes such as the reduced confusion between car horns, children and street music. Finally, we note that in some cases the confusion is actually increased, in particular between the air conditioner and drilling classes, both of which include temporally-stationary noise-like sounds.

5. SUMMARY

In this paper we studied the application of unsupervised feature learning to urban sound classification. We showed that classification accuracy can be significantly improved by feature learning if we take into consideration the specificities of this domain, primarily the importance of capturing the temporal dynamics of urban sound sources. In the future we intend to explore the use of multiscale time-frequency representations [14] and modulation spectra [26] as alternatives to shingling for encoding temporal dynamics, and the application of techniques from text-IR such as *tf-idf* weighting to increase the predictive power of our codebook [17]. Finally, we intend to extend our approach from single to multi-label classification so that our classifier can identify multiple concurrent sound sources, and investigate its application for detecting sound events in continuous audio recordings.

6. REFERENCES

[1] S. Chu, S. Narayanan, and C.-C.J. Kuo, "Environmental sound recognition with time-frequency audio features," *IEEE TASLP*, vol. 17, no. 6, pp. 1142–1158, 2009.

[2] R. Radhakrishnan, A. Divakaran, and P. Smaragdis, "Audio analysis for surveillance applications," in *IEEE WASPAA'05*, 2005, pp. 158–161.

[3] M. Xu, C. Xu, L. Duan, J. S. Jin, and S. Luo, "Audio keywords generation for sports video analysis," *ACM TOMCCAP*, vol. 4, no. 2, pp. 1–23, 2008.

[4] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li, "A user attention model for video summarization," in *10th ACM Int. Conf. on Multimedia*, 2002, pp. 533–542.

[5] D. Steele, J. D. Krijnders, and C. Guastavino, "The sensor city initiative: cognitive sensors for soundscape transformations," in *GIS Ostrava*, 2013, pp. 1–8.

[6] D. P. W. Ellis and K. Lee, "Minimal-impact audio-based personal archives," in *1st ACM workshop on Continuous archival and retrieval of personal experiences*, New York, NY, USA, Oct. 2004, pp. 39–47.

[7] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Audio context recognition using audio event histograms," in *18th EUSIPCO*, 2010, pp. 1272–1276.

[8] S. Chaudhuri and B. Raj, "Unsupervised hierarchical structure induction for deeper semantic analysis of audio," in *IEEE ICASSP*, 2013, pp. 833–837.

[9] L.-H. Cai, L. Lu, A. Hanjalic, H.-J. Zhang, and L.-H Cai, "A flexible framework for key audio effects detection and auditory context inference," *IEEE TASLP*, vol. 14, no. 3, pp. 1026–1039, 2006.

[10] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP JASMP*, vol. 2013, no. 1, 2013.

[11] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative evaluation of various MFCC implementations on the speaker verification task," in *10th Int. Conf. on Speech and Computer*, Patras, Greece, Oct. 2005, vol. 1, pp. 191–194.

[12] C. V. Cotton and D. P. W. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *IEEE WASPAA'11*, 2011, pp. 69–72.

[13] D. Stowell and M. D. Plumbley, "Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning," *PeerJ*, vol. 2, pp. e488, Jul. 2014.

[14] S. Dieleman and B. Schrauwen, "Multiscale approaches to music audio feature learning," in *14th Int. Soc. for Music Info. Retrieval Conf.*, Curitiba, Brazil, Nov. 2013.

[15] P. Hamel, S. Lemieux, Y. Bengio, and D. Eck, "Temporal pooling and multiscale learning for automatic annotation and ranking of music audio.," in *12th Int. Soc. for Music Info. Retrieval Conf.*, Miami, USA, Oct. 2011, pp. 729–734.

[16] Y. Vaizman, B. McFee, and G. Lanckriet, "Codebook-based audio feature representation for music information retrieval," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1483–1493, Oct. 2014.

[17] E. Amid, A. Mesaros, K. J Palomaki, J. Laaksonen, and M. Kurimo, "Unsupervised feature extraction for multimedia event detection and ranking using audio content," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 5939–5943.

[18] A. Coates and A. Y. Ng, "Learning feature representations with K-means," in *Neural Networks: Tricks of the Trade*, pp. 561–580. Springer, 2012.

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

[20] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra, "ESSENTIA: an audio analysis library for music information retrieval," in *14th Int. Soc. for Music Info. Retrieval Conf.*, Curitiba, Brazil, Nov. 2013, pp. 493–498.

[21] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[22] I.S. Dhillon and D.M. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, no. 1, pp. 143–175, 2001.

[23] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[25] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[26] D. P. W. Ellis, X. Zeng, and J. H. McDermott, "Classifying soundtracks with audio texture features," in *IEEE ICASSP*, 2011, pp. 5880–5883.