FEATURE LEARNING WITH DEEP SCATTERING FOR URBAN SOUND ANALYSIS

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

In this paper we evaluate the scattering transform as an alternative signal representation to the mel-spectrogram in the context of unsupervised feature learning for urban sound classification. We show that we can obtain comparable (or better) performance using the scattering transform whilst reducing both the amount of training data required for feature learning and the size of the learned codebook by an order of magnitude. In both cases the improvement is attributed to the local phase invariance of the representation. We also observe improved classification of sources in the background of the auditory scene, a result that provides further support for the importance of temporal modulation in sound segregation.

Index Terms— Unsupervised learning, scattering transform, acoustic event classification, urban, machine learning

1. INTRODUCTION

Audio classification systems have traditionally relied on hand crafted features, a popular choice being the Mel-Frequency Cepstral Coefficients (MFCCs) [1]. Examples of systems that rely on manually engineered features in the domain of environmental sound source classification include [2–4]. Recent studies in audio classification have shown that accuracy can be improved by using unsupervised feature learning techniques as an alternative to manually designed features, with examples in the areas of bioacoustics [5], music information retrieval [6–8] and environmental sound classification [9–11].

Still, 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 [5]. As a consequence, even systems that use feature learning need to first transform the signal into a representation that lends itself to successful learning. For audio signals, a popular representation is the mel-spectrogram [5, 6, 11].

In [11], we studied the application of unsupervised feature learning from mel-spectrograms for urban sound source classification. We showed that the learned features can outperform MFCCs due to their ability to capture the short-term temporal dynamics of the sound sources – a particularly important trait when dealing with sounds such as idling engines or jackhammers whose instantaneous noise-like characteristics can be hard to distinguish otherwise. To achieve the learning of temporal dynamics we applied frame shingling, i.e. the features were learned from groups of several consecutive frames (2D time-frequency patches). Whilst we were able to improve classification accuracy in this way, we noted that the downside to the approach was that we had to learn a separate codeword (feature) to encode every phase-shift within a 2D patch, which consequently required learning a larger codebook (set of features) overall.

Over the years we have seen the advent of alternative signal representations (i.e. transforms) for audio classification, and in particular representations that encode amplitude modulation over time such as the modulation spectrogram [12, 13] and more recently the scattering transform [14–16]. The latter in particular, also referred to as the deep scattering spectrum, has been shown to be stable to time-warping deformations and capable of characterizing time-varying structure over (relatively) long window sizes compared to those used for computing MFCCs for instance. This suggests that the scattering transform could characterize the short-term temporal dynamics captured by 2D mel-spectrogram patches with the added advantage of being phase invariant. From the sound perception and cognition literature we know that modulation plays an important role in sound segregation and the formation of auditory images [17–19], which further motivates the exploration of representations such as the scattering transform for machine listening. It has already been shown to be a useful representation for environmental sound classification in [20], although no feature learning was applied in that study.

In this paper we study the use of the scattering transform in combination with the feature learning and classification pipeline proposed in [11] for the classification of urban sound sources. In Section 2 we describe the signal representations compared in this study. The feature learning and classification algorithms we use are described in Section 3, and our dataset and experimental design in Section 4. Results are discussed in Section 5 and a summary is provided in Section 6.

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

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

2. SIGNAL REPRESENTATIONS

2.1. Mel Spectrogram

The mel-spectrogram is obtained by taking the short-time Fourier transform and mapping its spectral magnitudes onto the perceptually motivated mel-scale [21] using a filterbank in the frequency domain. It is the starting point for computing MFCCs [1], and a popular representation for many audio analysis algorithms including ones based on unsupervised feature learning [5–7, 11]. As in [11], we compute log-scaled mel-spectrograms with 40 bands between 0-22050 Hz using a 23 ms long Hann window (1024 samples at a sampling rate of 44.1 kHz) and a hop size of equal length. The representation is computed using the Essentia library [22] which provides Python bindings to a C++ implementation.

In a feature learning framework (cf. Section 3), we can choose to learn features from individual frames of the mel-spectrogram or alternatively group the frames into 2D patches (by concatenating them into a single longer vector) and apply the learning algorithm to the patches. In [11] we showed that the latter approach facilitates the learning of features that capture short-term temporal dynamics, outperforming MFCCs for classification. Following this result, we group consecutive frames to form 2D patches with a time duration of roughly 370 ms. Given our analysis parameters this corresponds to grouping every 16 consecutive frames.

2.2. Scattering Transform

As noted in the introduction, grouping mel-spectrogram frames captures temporal structure at the cost of having to learn a larger number of features to cover all possible phase shifts of a sound pattern within a 2D patch. An alternative solution would be to use a transform that can characterize amplitude modulations in a phase-invariant way: the scattering transform [16].

The scattering transform can be viewed as an extension of the mel-spectrogram that computes modulation spectrum coefficients of multiple orders through cascades of wavelet convolutions and modulus operators. Given a signal x, the first order (or "layer") scattering coefficients are computed by convolving x with a wavelet filterbank ψ_{λ_1}, taking the modulus, and averaging the result in time by convolving it with a low-pass filter $\phi(t)$ of size T:

$$S_1 x(t, \lambda_1) = |x * \psi_{\lambda_1}| * \phi(t). \quad (1)$$

The wavelet filterbank ψ_{λ_1} has an octave frequency resolution Q_1. By setting $Q_1 = 8$ the filterbank has the same frequency resolution as the mel filterbank, and this layer is approximately equivalent to the mel-spectrogram. The second order coefficients capture the high-frequency amplitude modulations occurring at each frequency band of the first layer and are obtained by:

$$S_2 x(t, \lambda_1, \lambda_2) = ||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \phi(t). \quad (2)$$

The octave resolution of the second order filterbank is determined by Q_2. Following [16] we set $Q_2 = 1$. Higher order coefficients can be obtained by iterating over this process, but it has been shown that for the value of T used in this study (see below) most of the signal energy is captured by the first and second order coefficients [16]. Furthermore, adding higher-order coefficients would blow-up the dimensionality of the representation. Consequently, in this study we use the first and second orders for our scattering representation.

It is beyond the scope of this paper to describe the scattering transform in greater detail, and we refer the interested reader to [14–16] for further information. It will suffice to note that the important parameters of the transform relevant to our experiments are the filterbank resolutions Q_1 and Q_2 and the duration T of the averaging filter which also represents the duration of the modulation structure we hope to capture using the second order coefficients. As noted above, we fix $Q_1 = 8$ and $Q_2 = 1$. The filterbank is constructed of 1D Morlet wavelets. We set T to the same duration covered by the 2D mel-spectrogram patches, i.e. 370 ms (for a sampling rate of 44.1 kHz this implies $T = 1024 \times 16$). To compute the scattering transform we use the ScatNet v0.2 Matlab software[1]. By default ScatNet computes the scattering coefficients without any oversampling, meaning the time-difference between consecutive frames is $T/2$. ScatNet provides an oversampling parameter that allows us to obtain a finer temporal representation of the transform. By modifying this parameter we can emulate different hop sizes ranging from $T/2$ down to the hop size we use for the mel-spectrograms ($T/16$). Importantly, note that the hop size is in inverse relationship to the number of analysis frames and consequently to the amount of data used for unsupervised feature learning. For each frame we concatenate the first order coefficients with all of the second order coefficients into a single feature vector. The second order coefficients are normalized using the previous order coefficients as described in [16]. Note that we also experimented with using the second order coefficients only, but this option resulted in significantly lower classification accuracies and is hence not explored further in this study.

3. FEATURE LEARNING & CLASSIFICATION

3.1. Feature learning with spherical k-means

Given our representation (log-mel-spectrogram patches or scattering transform frames), we learn a codebook of representative codewords from the training data. The samples in the dataset are then encoded against this codebook and the resulting code vectors are used as feature vectors for training / testing a classifier. To learn the codebook we use the *spherical k-means* algorithm [23]. Unlike the traditional k-means clustering algorithm [24], the centroids are constrained to have unit L2 norm (they must lie on the unit sphere), the

[1] http://www.di.ens.fr/data/software/scatnet/

benefits of which are discussed in [23, 25]. The algorithm has been shown to be competitive with more complex (yet considerably slower) techniques such as sparse coding, and has been used successfully to learn features from audio for music [6], birdsong [5] and urban sound classification [11]. As a feature learning technique, we use the algorithm to learn an over-complete codebook, so k is typically much larger than the dimensionality of the input data. For further details about the algorithm the reader is referred to [23].

Before passing the log-mel-spectrogram patches or scattering transform frames to the clustering algorithm we first reduce the dimensionality of the input data by decorrelating the input dimensions using PCA whitening. This has been shown to significantly improve the discriminant power of the learned features [23]. Following the procedure proposed in [6], we apply PCA whitening and keep enough components to explain 99% of the variance. Empirically, this reduces the dimensionality of a mel-spectrogram patch from 640 to 250, and of a scattering frame from 427 to 230. The clustering produces a codebook matrix with k columns, where each column represents a codeword. Every sample in our dataset is encoded against the codebook by taking the scalar product between each of its frames (or patches) and the codebook matrix. This approach was shown to work better than e.g. vector quantization [8] for the dataset studied in this paper [11]. Finally, we have to summarize the per-frame (or per-patch) values produced by the encoding over the time axis to ensure that all samples in the dataset are represented by a feature vector of the same dimensionality. Following [11] we summarize the encoded values over time with two statistics: the mean and standard deviation. The resulting feature vectors are thus all of size $2k$, and we standardize them across samples before passing them on to the classifier.

3.2. Classification

For each representation we experimented with two different classification algorithms: a random forest classifier [26] with 500 trees and a support vector machine [27] with a radial basis function kernel. In this way we can select the classifier that is best suited for each representation and focus on comparing the best results obtained by the two representations. Based on these experiments, we use the random forest classifier for the mel-spectrograms and the support vector machine for the scattering transform. For both classifiers we use the implementation provided in the *scikit-learn* Python library [28] with their default hyper-parameter values.

4. EXPERIMENTAL DESIGN

We use the UrbanSound8K [29] dataset for our experiments. It contains 8732 audio samples of up to 4 seconds in duration taken from real field recordings. The samples include sounds from 10 classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren and street music; and come divided into 10 stratified subsets for unbiased cross-validation. Since the samples come from field recordings, there are often other sources present in addition to the labeled source. For each sample the dataset indicates whether the labeled source is in the foreground or background (as subjectively perceived by the dataset annotators).

Each experiment is run using 10-fold cross validation based on the stratified subsets. We compute the classification accuracy for each fold and present the results as a box plot.

5. EXPERIMENTS & RESULTS

5.1. Experiment 1: Mel-spectrogram versus Scattering

In our first experiment, we compare the classification accuracy obtained when using the mel-spectrogram as input versus the scattering transform, varying the size k of the codebook from 200 to 2000. For this experiment we over-sample the scattering transform such that it contains the same number of analysis frames as the mel-spectrogram. In this way both representations produce the same amount of training data (frames) for the feature learning algorithm.

The results are presented in Figure 1. As reference, note that a baseline system using MFCCs with a random forest (and no learning) obtained a mean accuracy of 0.69 for the same dataset [11]. The first thing we see is that the scattering transform produces comparable results to the shingled mel-spectrogram (it actually outperforms the mel-spectrogram in the best case, but the difference is not statistically significant). Both outperform the MFCC baseline (statistically significant according to a paired t-test with $p < 0.05$). Whilst we had to group frames into 2D patches to capture the temporal structure of sounds with mel-spectrograms, we see that this information is captured by a single frame of the scattering transform (with first and second order coefficients), as one might expect. The more remarkable result is that whilst the performance using the mel-spectrogram increases as we increase k, using the scattering transform we actually get better performance with smaller values of k (the best being $k = 500$). This can be explained by the local phase invariance of the latter: when using mel-spectrogram patches we need to learn features to encode every phase-shift of a sound within a patch, whereas a scattering transform frame is invariant to this shift. This means we can obtain the same classification accuracy whilst reducing the size of the codebook by an order of magnitude. This could have a significant influence on the computational cost of the approach as we scale the problem to larger datasets with more classes and more training samples. Note that in practice computing the scattering transform will take longer than computing the mel-spectrogram by a multiplicative factor proportionate to the dimensionality of the scattering output, so the effective gain in efficiency will depend on the specific parameterization used.

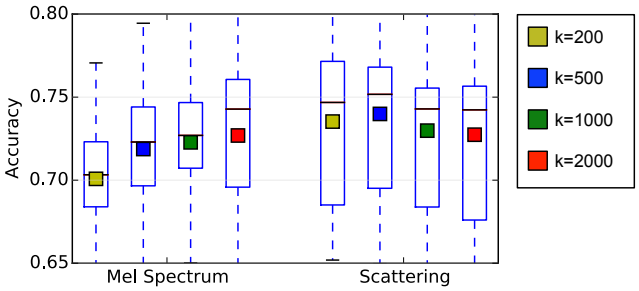

Fig. 1. Classification accuracy results: mel-spectrogram with feature learning vs. scattering transform with feature learning. The codebook size k is varied from 200 to 2000.

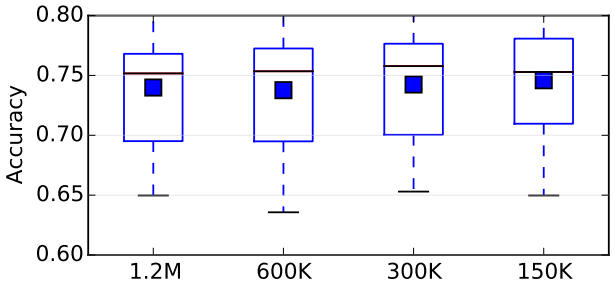

Fig. 2. Classification accuracy for the scattering transform (with $k = 500$) as a function of the training data size.

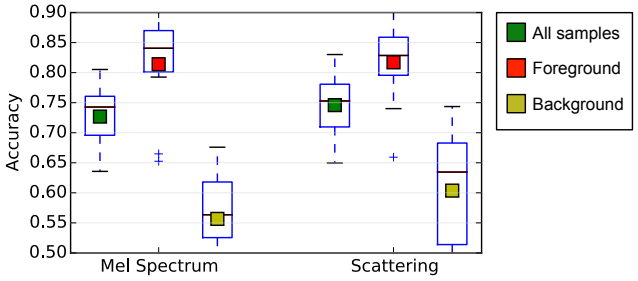

Fig. 3. Best classification accuracy obtained using the mel-spectrum and scattering transform with a breakdown into foreground and background.

5.2. Experiment 2: Scattering Oversampling

In our first experiment we oversampled the scattering transform so that the number of analysis frames equaled the number of frames in the mel-spectrogram. In our second experiment, we compute the scattering transform with different degrees of oversampling: from 43 frames per second (equivalent to the analysis hop size of the mel-spectrogram of 23 ms) down to the critical sampling rate of the transform coefficients of 5.5 frames per second (a hop size of 185 ms). This in turn affects the total number of training frames produced for the feature learning stage: from approximately 1.2M down to 150K. Based on the results of the previous experiment, we fix $k = 500$.

In Figure 2 we present the classification accuracy obtained as a function of the size of the data used for unsupervised feature learning. The results are somewhat surprising – rather than observing a decrease in performance as we decrease the size of data, the results remain stable. As with the previous experiment, this is likely due to the local phase invariance of the scattering transform: there is no need to represent every phase shift of a 2D patch in the training data since a single scattering frame represents all possible shifts simultaneously. Together with the findings of the previous experiment, the results are highly encouraging – by replacing the mel-spectrogram with the scattering transform we are able to reduce both the amount of training data and the size of the learned codebook by an order of magnitude whilst maintaining (or even improving) the classification accuracy.

5.3. Experiment 3: Foreground versus Background

Finally, recall that each sample in the dataset is also labeled as being in the foreground or background. In Figure 3 we plot the best classification accuracy for the mel-spectrogram and the scattering transform alongside a breakdown into foreground accuracy and background accuracy. Whilst for the foreground sounds the representations produce very similar results (mean accuracies of 0.814 and 0.817 respectively), we observe a considerable improvement for background sounds using the scattering transform: 0.557 vs. 0.604. Though further investigation would be required to make any hard claims, our results align nicely with those of the sound perception and cognition literature [17–19], suggesting that the scattering transform representation better facilitates the (machine) segregation of sound sources due to its characterisation of modulation, in particular when the source of interest is masked by other sounds.

6. SUMMARY

In this paper we evaluated the use of the scattering transform as an alternative to the mel-spectrogram in the context of unsupervised feature learning for environmental sound classification. We showed that we can obtain comparable (or slightly better) performance using the scattering transform whilst reducing both the amount of training data required for feature learning and the size of the learned codebook by an order of magnitude. In both cases the improvement is attributed to the local phase invariance of the representation. We also noted that the observed increase in performance was primarily due to the improved classification of sources in the background of the auditory scene, a result which aligns with the evidence found in the sound perception and cognition literature about the importance of temporal modulation for sound segregation.

REFERENCES

[1] 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*, Greece, Oct. 2005, pp. 191–194.

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

[3] 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.

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

[5] 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.

[6] S. Dieleman and B. Schrauwen, "Multiscale approaches to music audio feature learning," in *14th ISMIR*, Curitiba, Brazil, Nov. 2013.

[7] P. Hamel, S. Lemieux, Y. Bengio, and D. Eck, "Temporal pooling and multiscale learning for automatic annotation and ranking of music audio.," in *12th ISMIR*, Oct. 2011, pp. 729–734.

[8] Y. Vaizman, B. McFee, and G. Lanckriet, "Codebook-based audio feature representation for music information retrieval," *IEEE TASLP*, vol. 22, no. 10, pp. 1483–1493, Oct. 2014.

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

[10] 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 ICASSP*, Italy, May 2014, pp. 5939–5943.

[11] J. Salamon and J. P. Bello, "Unsupervised feature learning for urban sound classification," in *IEEE ICASSP*, Brisbane, Australia, Apr. 2015.

[12] L. Atlas and Shihab A. Shamma, "Joint acoustic and modulation frequency," *EURASIP J. on Applied Signal Processing*, vol. 2003, pp. 668–675, Jan. 2003.

[13] N. H. Sephus, A. D. Lanterman, and D. V. Anderson, "Modulation spectral features: In pursuit of invariant representations of music with application to unsupervised source identification," *J. of New Music Research*, vol. 44, no. 1, pp. 58–70, 2015.

[14] J. Andén and S. Mallat, "Multiscale scattering for audio classification," in *12th Int. Soc. for Music Info. Retrieval Conf.*, Miami, USA, Oct. 2011, pp. 657–662.

[15] J. Andén and S. Mallat, "Scattering representation of modulated sounds," in *15th DAFx*, UK, Sep. 2012.

[16] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Trans. on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, Aug. 2014.

[17] S. McAdams, *Spectral fusion, spectral parsing and the formation of auditory images*, Ph.D. thesis, Stanford University, Stanford, USA, 1984.

[18] W.A. Yost, "Auditory image perception and analysis: The basis for hearing," *Hearing Research*, vol. 56, no. 1, pp. 8–18, 1991.

[19] R.P. Carlyon, "How the brain separates sounds," *Trends in cognitive sciences*, vol. 8, no. 10, pp. 465–471, 2004.

[20] C. Bauge, M. Lagrange, J. Anden, and S. Mallat, "Representing environmental sounds using the separable scattering transform," in *IEEE ICASSP*, Vancouver, Canada, May 2013, pp. 8667–8671.

[21] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *JASA*, vol. 8, no. 3, pp. 185–190, 1937.

[22] 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.*, Brazil, Nov. 2013, pp. 493–498.

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

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

[25] 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.

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

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[28] 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.

[29] 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.