BINAURAL SCENE CLASSIFICATION WITH WAVELET SCATTERING

Vincent Lostanlen[*]

Département d'Informatique
École normale supérieure
Paris, France
vincent.lostanlen@ens.fr

Joakim Andén[*]

Program in Applied and Computational Mathematics
Princeton University
Princeton, NJ, USA
janden@math.princeton.edu

ABSTRACT

This technical report describes our contribution to the scene classification task of the 2016 edition of the IEEE AASP Challenge for Detection and Classification of Acoustic Scenes and Events (DCASE). Our computational pipeline consists of a gammatone scattering transform, logarithmically compressed and coupled with a per-frame linear support vector machine. At test time, frame-level labels are aggregated over the whole recording by majority vote. During the training phase, we propose a novel data augmentation technique, where left and right channels are mixed at different proportions to introduce invariance to sound direction in the training data.

***Index Terms*—** scattering transform, wavelets, auditory scene classification, orientation invariance, support vector machine

1. SYSTEM OUTLINE

The system used for the scene classification task is illustrated in Figure **??**. Each recording is decomposed using a time scattering transform, which provides a signal representation that is locally invariant to time-shifting and stable to time-warping deformation. Since small changes in timing have little relevance to the auditory scene of a particular recording, this invariance reduces the variability of the data without necessarily hampering discriminability. The averaging scale of the scattering transform is fixed to be 740 ms. In order to account for the varying orders of magnitude across frequencies, we then apply a logarithmic compression to the data. Each recording is thus represented as a sequence of logarithmically compressed scattering vectors.

Due to the stereophonic nature of the recordings, we augment the training data by constructing monophonic signals with different mixing proportions, effectively providing invariance to sound direction. At the training stage, a linear support vector machine (SVM) classifier is trained using the sequences obtained from the training data. For each recording in the testing set, the classifier is applied to all scattering vectors in the recording, yielding a class for each vector. The class of the entire recording is then determined by majority vote. Evaluating this system on the standard four train-test splits in the development data, we obtain an average accuracy of 79.4%.

[*]This work is supported by the ERC InvariantClass grant 320959. The source code to reproduce figures and experiments is freely available at `http://www.github.com/lostanlen/dcase2016`.

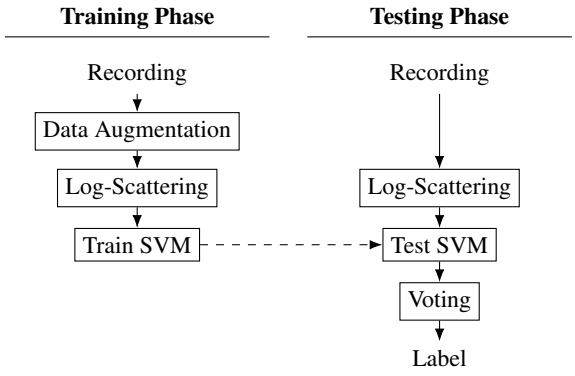


Figure 1: The scattering classification system proposed for the scene classification challenge. During the training phase, the data is augmented by mixing the left and right channels at various proportions. The log-scattering vectors are then computed and used to train a linear SVM. This SVM is then used to classify the sequence of log-scattering vectors for each test recording. The resulting sequence of labels is reduced to a single recording label through majority vote.

2. SCATTERING REPRESENTATION

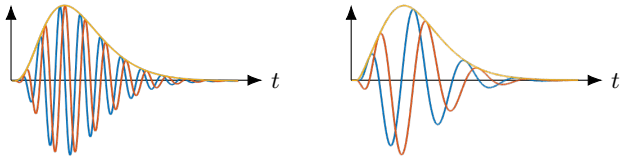
The scattering transform was introduced by S. Mallat as a signal representation that is invariant to translations and stable to deformation [**?**]. It has had success in classifying images [**?**], audio [**?**], and biomedical signals [**?**]. For audio signals, translation corresponds to time-shifting, while deforming a signal warps it in time. Both of these transformations have little to no effect on the semantic content of an audio signal, so reducing their influence enables us to train more accurate classifiers using limited training data. A brief review of the scattering transform is provided in this section.

First, for a signal x, let us define its Fourier transform $\widehat{x}$ by

$$\widehat{x}(\omega) = \int_{\mathbb{R}} x(u) e^{-i\omega u}\, du. \tag{1}$$

Given an analytic filter ψ with Fourier transform $\widehat{\psi}$ concentrated around the dimensionless frequency 1, we define a wavelet filter bank $\{\psi_\lambda\}_{\lambda>0}$ by dilating ψ, called the mother wavelet, to obtain

$$\psi_\lambda(t) = \lambda \psi(\lambda t). \tag{2}$$



(a) (b)

Figure 2: Gammatone wavelets $\psi(t)$ in the time domain with quality factors (a) $Q = 4$ and (b) $Q = 1$. Blue and red oscillations represent the real and imaginary parts. The orange envelope represents the complex modulus.

The analyticity of ψ forces its Fourier transform to be zero for negative frequencies. The wavelet filters ψ_λ are also analytic and since $\widehat{\psi}$ is centered around frequency 1, $\widehat{\psi}_\lambda$ is centered around λ.

A function ψ that fulfills our conditions is the pseudo-analytic Gammatone wavelet, illustrated in Figure **??**. Its form is given explicitly by

$$\psi(t) = \left((n-1)t^{n-2} + it^{n-1}\right) e^{-(b+i)t}\, 1_{[0,\infty)}(t), \qquad (3)$$

where the bandwidth parameter b is roughly proportional to $2^{-1/Q}$ and Q is the desired quality factor, i.e. the ratio of the center frequency to the half-maximum bandwidth in the Fourier domain. To begin with, we shall take $Q = 4$.

Given a signal x, we decompose it using the wavelet filter bank to obtain

$$x * \psi_{\lambda_1}(t) \quad \text{for } \lambda_1 > 0, \qquad (4)$$

known as the wavelet decomposition of x. We have denoted the standard convolution operator by $*$. The dilation structure of the wavelet filter bank means that we do not need to sample λ_1 continuously. Rather, it is sufficient to sample λ_1 as $2^{j/Q}$, where Q is the quality factor of the mother wavelet ψ. This means that we sample uniformly in log-frequency $\log \lambda_1$.

The wavelet decomposition itself is very sensitive to time-shifting and time-warping, which can be partially mitigated by taking the complex modulus. The result is known as the wavelet scalogram and we denote it by

$$x_1(t, \log \lambda_1) = |x * \psi_{\lambda_1}(t)|. \qquad (5)$$

The scalogram provides a useful representation of the time-frequency content of a signal. At a given point $(t, \log \lambda_1)$, it gives the intensity of x at time t and log-frequency $\log \lambda_1$[1]. However, it does not have the desired invariance and stability properties. To achieve this, we average the scalogram in time using a lowpass filter $\phi_T(t)$ of duration T to give

$$\begin{aligned} S_1 x(t, \log \lambda_1) &= x_1(\cdot, \log \lambda_1) * \phi_T(t) \\ &= |x * \psi_{\lambda_1}| * \phi_T(t). \end{aligned} \qquad (6)$$

In our configuration ϕ_T is given by a Gabor filter centered at frequency 0 with the desired bandwidth T in time. The coefficients $S_1 x$ are known as first-order time scattering coefficients and are comparable to the commonly used mel-frequency spectrogram coefficients [**?**].

The averaging by ϕ_T discards fine-scale temporal structure in the scalogram x_1. To recover this, we calculate a second wavelet decomposition on the scalogram along the time axis. Instead of the quality factor $Q = 4$ used in the first decomposition, we now use $Q = 1$. As before, we compute the complex modulus and obtain

$$\begin{aligned} x_2(t, \log \lambda_1, \log \lambda_2) &= x_1(\cdot, \log \lambda_1) * \psi_{\lambda_2}(t) \\ &= \big|\, |x * \psi_{\lambda_1}| * \psi_{\lambda_2}(t)\big|. \end{aligned} \qquad (7)$$

This second-order wavelet scalogram describes the modulation structure of the frequency band centered at $\log \lambda_1$ of the first-order scalogram x_1. It is therefore closely related to modulation spectrograms [**?**, **?**], but are defined using wavelet decompositions instead of short-time Fourier transforms.

Again, to obtain invariance, the second-order scalogram x_2 is averaged in time using the lowpass filter ϕ_T to give

$$\begin{aligned} S_2 x(t, \log \lambda_1, \log \lambda_2) &= x_2(\cdot, \log \lambda_1, \log \lambda_2) * \phi_T(t) \\ &= \big|\, |x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \phi_T(t), \end{aligned} \qquad (8)$$

which are known as second-order time scattering coefficients. Like modulation spectrograms, these coefficients provide information on the modulation structure of the signal x at log-frequency $\log \lambda_1$, but do so in a stable manner due to the wavelet construction. In this way, they are more closely related to constant-Q averaged modulation spectrograms [**?**, **?**]. We note that the above procedure can be continued for third- and higher-order scattering coefficients, but that for most applications, first- and second-order coefficients suffice.

Concatenating all the first-order scattering coefficients into one vector

$$S_1 x(t) = \{S_1 x(t, \log \lambda_1)\}_{\lambda_1 > 0}, \qquad (9)$$

and doing the same for the second-order coefficients

$$S_2 x(t) = \{S_2 x(t, \log \lambda_1, \log \lambda_2\}_{\lambda_1 > 0, \lambda_2 > 0}, \qquad (10)$$

we can combine all of them into one scattering vector at time t

$$S x(t) = \{S_1 x(t), S_2 x(t)\}. \qquad (11)$$

It is important here to remark that, although the above formulas cover continuous domains in t, $\log \lambda_1$, and $\log \lambda_2$, these variables can all be sampled discretely without great loss of information. As mentioned earlier, $\log \lambda_1$ and $\log \lambda_2$ can be sampled uniformly with a step proportional to $1/Q$. In addition, the lowpass nature of the scalogram x_1 in time ensures that many coefficients in x_2 will be negligible for large values of $\log \lambda_2$. As a result, these can be safely excluded from the transform. Finally, the lowpass filtering by ϕ_T ensures that we can sample the final scattering vector Sx along multiples of $T/4$ in time.

3. SCATTERING POST-PROCESSING

Instead of feeding the raw scattering vectors into the SVM classifier, we process them to facilitate model building by reducing their dynamic range and standardizing their variability. Specifically, we first compute the log-scattering coefficients by taking the logarithm of each value in the scattering vector to get

$$\log S_1 x(t, \log \lambda_1) = \log\left(|x * \psi_{\lambda_1}| * \phi_T(t)\right) \qquad (12)$$

in the first order and

$$\log S_2 x(t, \log \lambda_1, \log \lambda_2) = \log\left(|\, |x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \phi_T(t)\right) \qquad (13)$$

in the second order. These are combined across all log-frequencies $\log \lambda_1$ and $\log \lambda_2$ as before to yield a log-scattering vector $\log Sx(t)$.

The log-scattering coefficients are better suited for audio classification since audio amplitudes can vary across several orders of magnitude without significantly changing the content of the signal. This is often characterized as the Weber-Fechner law in psychoacoustics.

In addition, we diagonally standardize the coefficients to have mean zero and unit variance. To do this for our train-test splits, we calculate the mean and variance for each scattering coefficient across our training data and then use this to standardize both the training and testing data for the current split. This helps improve the conditioning of the SVM training since all values are in the same numerical range.

4. DATA AUGMENTATION

Most acoustic scene datasets are recorded according to a binaural protocol, i.e. with a pair of in-ear microphones [**?**]. This protocol provides a realistic description of the spatial auditory environment, as it reproduces the natural listening conditions of humans. In particular, the interaural level difference (ILD) between the left and right channels depends on the direction of each sound source that make up the scene with respect to the listener [**?**]. Yet, since the microphone location and direction vary across instances of the same class, the signal representation should be invariant to these parameters.

In order to achieve invariance to changes in direction, averaging the left and right channels into a monophonic signal is by far the most widespread approach. If these are denoted by x_{L} and x_{R}, respectively, we would thus have

$$x_{\mathsf{M}}(t) = \frac{1}{2} x_{\mathsf{L}}(t) + \frac{1}{2} x_{\mathsf{R}}(t), \tag{14}$$

where x_{M} is the mixed signal. However, it favors the center of the scene while attenuating lateral cues. Indeed, if x_{L} is zero and x_{R} contains the entire signal, the resulting mixed signal x_{M} will have a lower amplitude.

To create the desired invariance without overemphasizing the central direction, we calculate multiple combinations of the left and right channels into the monophonic signal x_α using the formula

$$x_\alpha(t) = \frac{1+\alpha}{2} x_{\mathsf{L}}(t) + \frac{1-\alpha}{2} x_{\mathsf{R}}(t), \tag{15}$$

where α determines the mixing between the channels. Note that taking $\alpha = 0$ gives us the center mixed signal x_{M} described above.

In following experiments, we set α to -1, $\frac{1}{2}$, 0, $+\frac{1}{2}$, and $+1$. This is a form of data augmentation since 5 monophonic signals are obtained from each binaural recording in the training set, increasing its size. At test time, only the center combination ($\alpha = 0$) is used to classify the recording.

5. CLASSIFIER

Given the recordings in a training set, we compute their scattering vectors as described in Section **??**. Each recording being 30 s long and the averaging window set at $T = 740$ ms, this results in 160 scattering vectors per recording, since Sx is sampled at intervals of size $T/4 = 186$ ms. From these, log-scattering vectors $\log Sx$ are computed as outlined in Section **??** and their mean and variance are computed across all recordings in the training set. These values are then used to standardize the log-scattering vectors to have mean zero and unit variance.

These log-scattering vectors are then fed, along with the class label of the recording they are extracted from, into the training algorithm for a linear SVM. An SVM is a binary linear classifier which is trained by finding the hyperplane that best separates two classes while minimizing training error [**?**]. In our system, we used the LIBLINEAR library [**?**]. Since the SVM is formulated as a binary classifier, a one-versus-all scheme is used to generalize its results to a multi-class setting.

To train the SVM, a cost parameter C is used to determine the trade-off between penalizing errors on the training data and the maximization of discriminatory margin. In our case, we saw very little difference in performance when varying this parameter, which suggests that there is little conflict between these two objectives and that the data is therefore close to linearly separable. As a result, we set C equal to 1.

Once the SVM is trained, we feed it log-scattering vectors from the testing set. As discussed in Section **??**, these are first standardized by the mean and variance derived from the training set. Again, a 30 s-recording in the testing set yields 160 log-scattering vectors, which means that the SVM returns 160 different class labels for a single recording. To obtain a label for the entire recording, we perform a majority vote among all the labels corresponding to the individual log-scattering vectors. This type of approach to aggregating representation vectors along a time series is known as a late integration scheme [**?**].

Scene	Baseline	Temporal scattering
beach	74.6 ± 18.9	83.5 ± 7.1
bus	58.2 ± 18.3	88.8 ± 11.4
cafe/restaurant	85.1 ± 10.8	64.5 ± 8.3
car	69.1 ± 20.8	94.9 ± 6.1
city_center	89.6 ± 9.2	91.7 ± 9.4
forest_path	72.4 ± 23.8	93.8 ± 7.8
grocery_store	74.1 ± 13.7	90.9 ± 6.9
home	78.1 ± 17.7	56.2 ± 23.9
library	65.1 ± 23.3	82.0 ± 13.0
metro_station	85.2 ± 15.6	96.0 ± 2.3
office	90.8 ± 16.0	87.5 ± 21.7
park	25.6 ± 11.0	75.4 ± 7.4
residential_area	75.1 ± 19.4	44.2 ± 15.0
train	34.1 ± 7.2	58.3 ± 10.0
tram	85.4 ± 13.9	83.7 ± 12.2
Average	70.8 ± 2.6	79.4 ± 3.0

Table 1: Classification results obtained through cross-validation on the development data for the DCASE 2016 scene classification challenge. The mean and standard deviation of the percentage correct in each class is provided along with the average across classes.

6. NUMERICAL RESULTS

To evaluate the time scattering system proposed above, we apply it to the development data provided with the DCASE 2016 scene classification challenge. This dataset is divided into four folds, which

can be used to form train-test splits by training on three folds and testing on the remaining fold. This cross-validation procedure provides an estimate of the mean classification accuracy along with its variability due to differing training sets.

The results for the entire system is provided in Table **??**. It is to be noted that while performance is quite good for most classes, certain classes ("cafe/restaurant", "residential_area", and "train") perform very poorly. These scenes are relatively quiet, with few characteristic sounds present, which could explain the difficulty in classifying them well. The exact cause for this difference in performance merits further study.

Despite these problems, the system provides a significant improvement over the baseline system, which consists of mel-frequency cepstral coefficients (MFCCs) modeled using Gaussian mixture models (GMMs) [**?**]. Since the scattering representation takes into account the modulation structure of the scalogram in addition to the average spectral envelope captured by MFCCs, this improvement in performance is to be expected. We expect that further increases in performance can be achieved for this task by introducing more sophisticated extensions of the scattering transform such as the joint time-frequency scattering transform [**?**].

7. REFERENCES

[1] S. Mallat, "Group invariant scattering," *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.

[2] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

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

[4] V. Chudáček, J. Andén, S. Mallat, P. Abry, and M. Doret, "Scattering transform for intrapartum fetal heart rate variability fractal analysis: A case-control study," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1100–1108, 2014.

[5] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[6] M. S. Vinton and L. E. Atlas, "Scalable and progressive audio codec," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 5. IEEE, 2001, pp. 3277–3280.

[7] H. Hermansky, "The modulation spectrum in the automatic recognition of speech," in *Proc. IEEE ASRU*, 1997, pp. 140–147.

[8] D. Ellis, X. Zeng, and J. McDermott, "Classifying soundtracks with audio texture features," in *Proc. IEEE ICASSP*, Prague, Czech Republic, May. 22-27 2011, pp. 5880–5883.

[9] J. K. Thompson and L. E. Atlas, "A non-uniform modulation transform for audio coding with increased time resolution," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 5. IEEE, 2003, pp. V–397.

[10] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006.

[11] J. Blauert, *Spatial Hearing: The psychophysics of human sound localization (revised edition)*. Cambridge, MA, USA: The MIT Press, 2004.

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

[13] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of machine learning research*, vol. 9, no. Aug, pp. 1871–1874, 2008.

[14] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On combining classifiers," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[15] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *24rd European Signal Processing Conference 2016 (EUSIPCO 2016)*, Budapest, Hungary, 2016.

[16] J. Andén, V. Lostanlen, and S. Mallat, "Joint time-frequency scattering for audio classification," in *Proc. MLSP*, 2015.