Sparse Coding for Drum Sound Classification and its Use as a Similarity Measure

Simon Scholler[1,2]
[1]Bernstein Center for Computational Neuroscience, Berlin, Germay
simon.scholler@googlemail.com

Hendrik Purwins[2]
[2]Music Technology Group, Barcelona, Spain
Barcelona, Spain
hendrik.purwins@upf.edu

ABSTRACT

Although rare in the sound recognition literature, previous work using features derived from a sparse temporal representation has led to some success [8, 2, 9]. A great advantage of deriving features from a temporal representation is that such an approach does not face the trade-off problem between time and frequency resolution. Here, we present a biologically inspired two-step process for audio classification: In the first step, efficient basis functions are learned in an unsupervised manner [13] on mixtures of percussion sounds (drum phrases). In the second step, features are extracted by using the learned basis functions to decompose percussion sounds (base drum, snare drum, hi-hat) with matching pursuit [7]. The classification accuracy in a 3-class database transfer task is 91.5% as opposed to 70.7% when using MFCC features. Further, we show that a MP-feature representation preserves sound similarity to a greater extent than MFCC-features, i.e. an artificial mixture of two sounds of equal energy normally lies in the middle between the two single sound distributions in feature space. An MP-representation thus inherently contains a similarity measure between different sounds.

1. GENERAL TERMS

Theory, Verification

2. KEYWORDS

Sparse Coding, Audio Recognition, Matching Pursuit

3. INTRODUCTION

The large majority of popular features used in audio classification are spectral methods, such as mel-frequency spectral coefficients (MFCCs) [10], spectral moments [14], or band-energy ratio [14]. However, as these features require the calculation of a spectrogram, they suffer from the trade-off between frequency and time resolution. Also, phase information which is crucial for consonant recognition and auditory grouping of sounds [12] gets lost when calculating features based on the magnitude spectrum only. Sparse temporal representations have become popular recently and first efforts have been made to use such models for audio recognition [8, 9].

The basic idea of modeling sparse coding [13] is simple: Given a set of input signals $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, the aim is to find a number of basis functions $\phi_1, \phi_2, \ldots, \phi_m$, such that each input signal can be approximated sparsely by a linear combination of a relatively small number of basis functions (also called atoms):

$$\mathbf{x}_i = \sum_{j=1}^{m} s_j \phi_j = \mathbf{s}^{\mathbf{T}} \Phi \quad \forall i = 1, 2, ..., n$$

where the activity vector $\mathbf{s}$ is sparse, i.e. units are either active or inactive (not significantly different from zero), where the fraction of active units is generally quite small given the total number of units. The complete set of the elementary basis functions is called a dictionary. In sparse coding, overcomplete dictionaries, i.e. dictionaries containing more atoms than the dimensionality of the signal, are normally used.

There is quite some evidence from computational neuroscience models that sparse coding might be a strategy that is pursued by the brain in order to extract relevant information out of sensory data.

For instance, in [13], natural sounds were encoded through a sum of shiftable and scalable basis functions, i.e. an (analogue) spike-code model. In an iterative process, they enforced sparse coding of environmental sounds and speech data on a set of basis functions. The resulting basis functions showed a close resemblance to the response properties of auditory nerve fibers and their distribution in the frequency-bandwidth plane roughly matched the distribution found by experimental studies in the cat.

Previously, the sparse optimization model of [13] has been applied to a music genre recognition task [8]. Short pieces of music were decomposed sparsely from a genre-recognition database by matching pursuit (MP) [7] and features from the resulting spike representation were computed. Although they demonstrated the general applicability and advantages of this temporal method for audio classification over spectral methods, they did not yield a significantly higher classification performance compared to classifying widely applied mel frequency cepstral coefficients (MFCCs). Also, the results with sparse coding were not as good as using an equal amount of gammatone atoms. Their results however increased slightly when using the combined set of MFCC and MP-features.

A similar approach was taken by [2]. They used features obtained by decomposing audio files with a sparse decomposition method (matching pursuit) and a dictionary of gammatone atoms. Using these features alone, the results on classifying environmental ambient sounds were lower than using MFCCs, but the performance of the combined feature set was significantly higher than

MML'10, October 25, 2010, Firenze, Italy.

using MFCCs alone. Other methods such as independent subspace analysis and non-negative matrix factorization have also been successfully applied to the classification of drum sounds (cf. e.g. [6]). In this paper, we will use sparse coding for the same purpose. Although we will only deal with drum data due to its relative simplicity (stable pitch), this is a general approach that can be applied to arbitrarily structured sound data.

4. METHODS

4.1 Data

We used isolated drum data samples from two databases, the ENST database [3] and the RWC musical instrument database [4], from which the following datasets were created:

- BSH: A subset of isolated drum sounds from the ENST drum database: base drum ('bd', 60 sounds), snare drum ('sd', 68 sounds), closed hi-hat ('chh', 46 sounds), open hi-hat ('ohh',62 sounds). This set was also used for all subsequent sets where the hi-hat sounds are merged to one class ('hh').
- AMIX: Artificial pairwise mixtures of the classes 'bd', 'sd' and 'hh' from BSH (100 samples each) and the single classes were used. We chose three energy ratios (0.5, 1, 2) for mixing the sounds. Sounds were mixed with a random offset up to a maximum distance of 50 ms between their amplitude peaks.
- DBT: Database transfer data. Classes 'bd, 'sd', 'hh' from BSH are used as training data, and testing was done on isolated drum samples ('bd', 'sd', 'hh') taken from the RWC music instrument database (12/34/36 samples respectively).

4.2 Features

Three different classes of features were used for classification. Two of these aim at expressing the signal sparsely in the temporal domain via a set of elementary basis functions. Since the problem of finding the sparsest representation using a dictionary of basis functions is NP-hard, we employed matching pursuit (MP) [7] which decomposes a signal in a sparse manner and is computationally feasible. MP is an iterative method. In each step, the basis functions of the dictionary are correlated with the signal and the basis function with the highest correlation is subtracted from the (residual) signal. This process is repeated until a stopping criterion is reached (e.g. signal-to-residual ratio, a fixed number of iterations, or a spiking threshold, cf. explanation below).

4.2.1 Matching pursuit using a sparse coding dictionary (SC-MP)

We generated matching pursuit features using a dictionary previously learned by sparse coding optimization [13]. Learning was done on a subset of the ENST drum database ('phrases'). The signal is described as a linear superposition of a number of shiftable basis functions ϕ_m, each having an associated temporal location τ^m and amplitude s^m [12]. The length of the basis functions is only restricted by the length of the signal. Further, each basis function can appear multiple times at different timepoints τ_i^m. Formally, we yield the following decomposition:

$$x(t) = \hat{x}(t) + \epsilon(t) = \sum_{m=1}^{M} \sum_{i=1}^{n_m} s_i^m \phi_m(t - \tau_i^m) + \epsilon(t),$$

where M is the total number of basis functions (the size of the dictionary), n_m is the total number of times the basis function ϕ_m is used in the current decomposition and ϵ represents the residual (gaussian noise in the ideal case).

A "spiking threshold" was used as the stopping criterion for matching pursuit, i.e. the matching pursuit decomposition was stopped when the correlation of a newly picked atom fell below 0.2.

Learning of the basis functions was done by gradient ascent with respect to the samples of the basis functions. The update rule is

$$\phi_m^{t+1} = \phi_m^t + \lambda \sum_i \hat{s}_i^m [x - \hat{x}]_{\tau_i^m}$$

i.e. each used basis function gets updated according to the decomposition residual $[x - \hat{x}]$ over the temporal extent of the basis function τ_i^m. The learning rate λ was set to $\lambda = 0.005$ in our case. Note that the value of the learning rate is highly dependent on the stopping criterion of the matching pursuit because it influences the decomposition error and thereby has a direct impact on the learning rule. Learning of the basis function is activity based, since after each decomposition of a sound, only those basis functions used in the MP-decomposition are updated. After each update step, the basis functions (vectors) were normalized to unit length.

The size of the atoms was kept variable in order to allow basis functions with low frequency components to grow over the initial length and basis functions with high frequencies to shrink. Each atom was initialized with a zero-padding on both ends that was 10% of the length of the basis function. If the vector norm of the padding exceeded a predefined extension threshold, the basis function was extended. If it is smaller than the shrinkage threshold the atoms were shortened.

For learning, 2000 iterations over a database of 134 drum phrases (8-33s duration, average: 17s) have been performed. Each decomposition on average consisted of more than 1200 atoms.

4.2.2 Matching pursuit using a gammatone dictionary (GT-MP)

In this case, matching pursuit features using a gammatone dictionary of the same size as the SC-MP dictionary were used. Gammatones functions are obtained by the product of a gamma distribution and a sinudoidal. To obtain the dictionary, we decomposed the ENST-subset that was also used for learning sparse basis functions ('phrases') with a dictionary of 64 gammatone atoms (equally spaced on the equivalent-rectangular bandwith (ERB) scale). The gammatone atoms that contributed most to the decomposition of the signals were then selected and used as the GT-MP dictionary. The reason for introducing GT-MP features is that they can be directly compared to SC-MP and thus serve as a test whether the computationally expensive preprocessing step of learning sparse basis function out of data is necessary to achieve a high classification performance.

4.2.3 Calculating matching pursuit features

Each decomposition (i.e. each audio file) gave one feature vector which was then used for classification. The MP-feature vector consists of two parts: The first part is a vector containing summed coefficients ("spike weights") of each dictionary atom in the sound decomposition normalized to unit length. The following part is a vector containing the total frequency each atom has been used ("spike counts") normalized by the frequency of the most often used atom in the decomposition.

4.2.4 MFCCs

We used mel-frequency cepstral coefficients (MFCCs) to compare the computed features with a classical approach. MFCCs

describe the spectral shape and are widely used in audio classification problems, especially in speech recognition. We obtained them using Slaney's Auditory Toolbox [11] and normalized to a maximum (absolute) amplitude of 1. Additionally to applying the three classes of features individually (MFCC, SC-MP, GT-MP), the combinations of MP-features with MFCC-features were also tested (MFCC + SC-MP, MFCC + GT-MP)

4.3 Classifier

To classify, we chose the random forest method [1] which is implemented in the machine learning environment WEKA [5]. Random forest creates an ensemble of decision trees using bootstrapped samples of the training data. We used random forest since on our data set it performed better in comparison to the standard approach of classifying with a support vector machine. Further, this method eases the investigation of feature interactions and the importance of features for classification.

4.4 Investigating sound similarity

Since sparse coding should make the higher statistical dependencies explicit in the data (e.g. the co-occurance of two frequencies), we investigated how the data is distributed in the feature space. We computed the feature centroids of each audio class and then projected the centroids on the first two principle components of the centroids. When dealing with sound mixtures, we can thereby investigate whether e.g. the class centroid of a mixture of two sounds lies in between the distribution of the isolated sounds.

5. CLASSIFICATION RESULTS

The classification results of the different feature classes are shown in Table 1. For the simple task of classifying the basic drum sounds ('bd', 'sd', 'hh') from the same database, all classifiers perform well (classification accuracies over 85%).

	BSH	AMIX	DBT
MFCC	86.3%	31.4%	70.7%
SC-MP	97.3%	70.7%	**91.5%**
GT-MP	89.5%	50.2%	69.3%
MFCC + SC-MP	**97.6%**	**75.3%**	87.8%
MFCC + GT-MP	94.1%	61.7%	79.3%
Baseline	33.3%	8.3%	33.3%

Table 1: Drum classification accuracy for all dataset and feature combinations. The MP-features were obtained using a dictionary of 16 atoms. The results of the first two columns were obtained by 5-fold cross-validation, the last column are results from a database transfer task. The cross-validation was performed such that samples from one recording were all either in the training or test set.

In the other two cases (artificial mixtures and databank transfer), SC-MP features greatly outperform MFCCs and even the MP features obtained with a gammatone dictionary. A combination of MFCCs with matching pursuit features leads to a further increase of the classification accuracy in almost all cases, suggesting that MP and MFCC features capture different kind of information about the signal.

The classification rate certainly is highly dependent on the dictionary used. Table 2 shows the classification accuracy for different sizes of the gammatone and sparse-coding dictionaries. For our data set, the use of 16 basis functions yields the best results for both SC-MP and GT-MP features. This is consistent with the finding that when learning 32 basis functions, only a small subset adopts a meaningful structure. However, the appropriate number of basis functions may depend on the complexity of the sounds. Further, SC features lead to a significantly better performance than GT features in all cases, independent of the size of the dictionary.

	BSH	AMIX	DBT
SC-MP 8	93.7%	59.8%	81.7%
GT-MP 8	87.2%	45.6%	79.3%
SC-MP 16	**97.3%**	**70.7%**	**91.5%**
GT-MP 16	89.5%	50.2%	69.5%
SC-MP 32	97.2%	61.3%	70.7%
GT-MP 32	88.4%	48.6%	67.2%
Baseline	33.3%	8.3%	33.3%

Table 2: Drum classification results for three different datasets and SC-MP and GT-MP features using a dictionary of size 8, 16, or 32. Again, the results of the first two columns were obtained by 5-fold cross-validation; the last column shows results from a database transfer task.

Figure 1 shows a sparse coding dictionary along with the relative use of the basis functions for the three drum sound classes depicted as pie charts. The basis function most closely resemble gammatone functions except for two basis functions, a sinusoidal and a high-frequency transient.

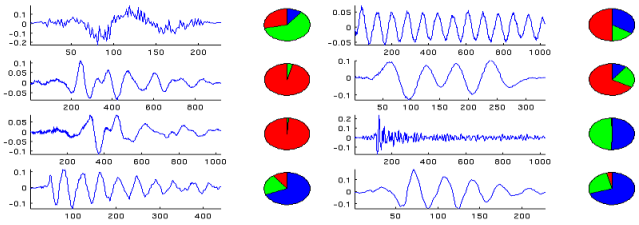


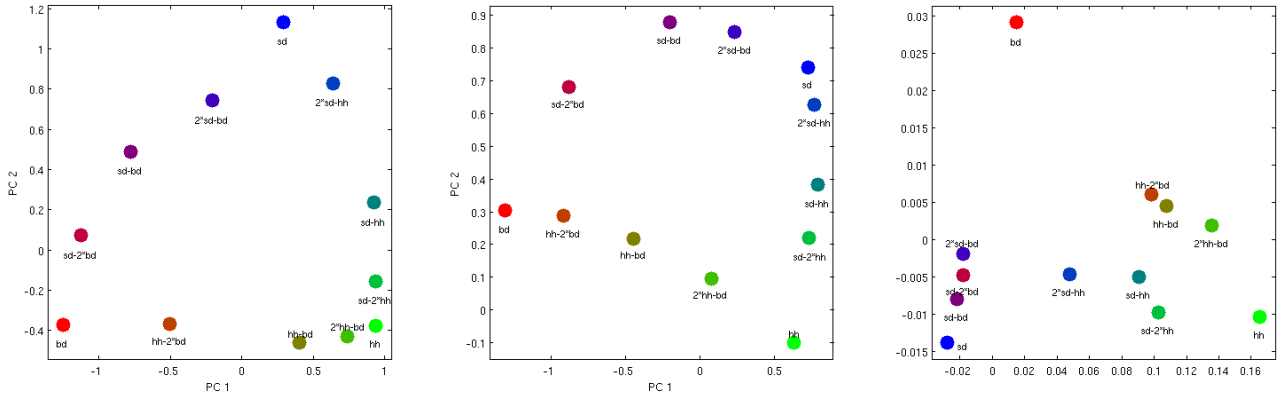
Figure 1: Learned basis function dictionary with 8 atoms. The pie charts depict the (relative) summed weights for the BSH dataset (red: 'bd', green: 'sd', blue: 'hh').

As the pie charts reveal, some of the basis functions are almost exclusively used in the decomposition of one sound class. These atoms therefore seem to be specialized to respond primarily to sounds of its preferred class indicating that they respond sparsely to percussion sounds.

Notice also that the spike count features of the specialized atoms seem to separate the classes more clearly than the weight features. This suggests that a binary spiking of the units (as done in real neurons) alone is sufficient to distinguish between the sound classes.

6. DRUM MIXTURES AND SIMILARITY

Figure 2 shows the centroids of the classes projected onto the two largest principal components. SC-MP and GT-MP features preserve sound similarity almost perfectly in the sense that similar mixed sound distributions lie in between the single sound distributions in feature space. Also, the distribution of the mixed sounds moves towards one single sound as the energy of the single sound becomes more dominant in the mixture. The MFCCs however do not preserve sound similarity as good as the MP-features since the mixtures do not lie equidistant between the original sounds and sometimes the expected order is not retained (centroids 'sd-bd',



(a) SC-MP (b) GT-MP (c) MFCC

Figure 2: Centroids of the class distributions of dataset AMIX projected onto the first and second principal component. The labels of the mixed classes contain the energy ratios of the mixtures, i.e. the sound distribution centroid consisting of a base drum sound and a snare drum sound of twice the energy as the base drum sound has the label 'bd-2*sd'. The color codes further exemplify the neighbourhood relations.

'sd-2*bd' and '2*sd-bd'). This is not only the case for the PC1-PC2 plane but is also visible in the pairwise distance matrix of the cluster centroids.

7. DISCUSSION

We have presented a biologically inspired approach to classify arbitrary sound classes by learning sparse basis function on unlabelled data and supervised learning of the classes using features derived from a temporal representation of the signal. This approach has certain advantages over other music-information retrieval methods. First, it is a general method that does not make any assumptions about the data except that the data can be sparsely represented by a small number of basis functions. In contrast to Fourier or wavelet analysis, no predefined basis functions are used but the most efficient ones from the data are learned. This approach is therefore probably superior to the Fourier or wavelet transform when dealing with heterogenous data, e.g. data consisting of transient and tonal components (such as speech or music). Selecting those basis functions that account for most of the energy leads to a sparse representation.

However, the presented method also has some drawbacks. The number of basis functions to be learned has to be set manually and it is not easy to determine what a reasonable number for given classes of sounds would be. The temporal structure of a sound decomposition such as the temporal sequence of the basis functions or the co-ocurrances of two basis functions is not considered in the feature extraction process.

Using a gammatone dictionary instead of a SC-dictionary decreases the classification significantly, even when using data from a different database for testing. An adapted dictionary thus seems to be highly beneficial for the classification although a recent music genre recognition study could not demonstrate this [8]. The usefulness of a sparse coding approach therefore seems to be highly dependent on the recognition task in question.

In our future work, we want to investigate if we obtain similar results for mixed sounds when using real mixtures. If the results from the artificial mixtures are representative, then this method has a wide variety of applications in music transcription systems.

8. REFERENCES

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

[2] S. Chu, S. Narayanan, and C.C.J. Kuo. Environmental sound recognition with time-frequency audio features. *Trans. Audio, Speech and Lang. Proc.*, 17(6):1142–1158, 2009.

[3] O. Gillet and G. Richard. Enst-drums: an extensive audio-visual database for drum signals processing. In *Proceedings of the 7th International Symposium on Music Information Retrieval (ISMIR 2006)*, pages 156–159. Citeseer, 2006.

[4] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Music genre database and musical instrument sound database. In *Proc. ISMIR*, pages 229–230. Citeseer, 2003.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[6] M. Helén and T. Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *Proc. EUSIPCO*, volume 2005. Citeseer, 2005.

[7] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[8] P. Manzagol, T. Bertin-Mahieux, D. Eck, et al. On the use of sparse time-relative auditory codes for music. In *Proc. International Conference on Music Information Retrieval (ISMIR). Philadelphia, PA: ISMIR*, pages 14–18, 2008.

[9] M.D. Plumbley, S.A. Abdallah, T. Blumensath, and M.E. Davies. Sparse representations of polyphonic music. *Signal Processing*, 86(3):417–431, 2006.

[10] L.R. Rabiner and B.H. Juang. *Fundamentals of speech recognition*. Prentice hall, 1993.

[11] M. Slaney. Auditory toolbox version 2. *Interval Research Corporation*, 10:1998, 1998.

[12] E. Smith and M.S. Lewicki. Efficient coding of time-relative

structure using spikes. *Neural Computation*, 17(1):19–45, 2005.

[13] E.C. Smith and M.S. Lewicki. Efficient auditory coding. *Nature*, 439(7079):978–982, 2006.

[14] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.