Musical audio analysis using sparse representations

Mark Plumbley, Samer Abdallah, Thomas Blumensath, Maria Jafari, Andrew Nesbit, Emmanuel Vincent, Beiming Wang

Musical Audio Analysis using Sparse Representations

M. D. Plumbley, S. A. Abdallah, T. Blumensath, M. G. Jafari, A. Nesbit, E. Vincent, and B. Wang

Centre for Digital Music, Department of Electronic Engineering, Queen Mary University of London, Mile End Road, London E1 4NS, UK.
mark.plumbley@elec.qmul.ac.uk

Summary. Sparse representations are becoming an increasingly useful tool in the analysis of musical audio signals. In this paper we will given an overview of work by ourselves and others in this area, to give a flavour of the work being undertaken, and to give some pointers for further information about this interesting and challenging research topic.

1 Introduction

Musical audio signals contain a large amount of underlying structure, due to the process through which music is generated. Human hearing is usually very good at analysing the structure of audio signals, a process known as *auditory scene analysis* [8]. To build machines able to analyse audio signals, one approach would be to build in knowledge about human hearing into a *computational auditory scene analysis* (CASA) system [10]. For example, a blackboard system could be used to integrate knowledge sources concerned with tracking particular frequency 'partials' into hypotheses about the notes present in a musical signal [5, 24].

In contrast, in this paper we adopt a data-driven approach. Here we use information about the statistics of musical audio signals to perform our analysis. In particular, we describe an approach to musical audio analysis based on a search for *sparse* representations, where any coefficient in such a representation has only a small probability of being far from zero [11, 27].

For music, it is not surprising that a musical audio signal would be generated from a small number of possible notes active at any one time, and hence allow a sparse representation [7, 30]. For example, for a standard piano there are 88 possible notes that could be played, with each note producing a particular sound at a particular pitch. However, in most piano pieces only a few (e.g. up to 4–6) of the notes are played at any one time, typically limited

by the chords (sets of simultaneous notes) desired by the composer, as well as the physical limit on the number of fingers available to the pianist [29]. This leads to the idea that music is *sparse*, in the sense that at a given time instant most of the available notes are not sounding.

Recently a number of techniques have been developed which aim to find sparse representations of signals and data [9, 12, 15, 23]. If we apply these techniques to analysing musical signals we may be able to recover the sparse 'objects' that produced the musical audio signal. Such a sparse representation could be applied to automatic music transcription (identifying the notes from the musical audio), source separation, or efficient coding of musical audio.

In this paper we will given an overview of work by ourselves and others in this area, to give a flavour of the work being undertaken, and to give some pointers for further information. The paper is organized as follows. In Section 2 we describe a probabilistic approach to inference of sparse components and learning of a representation dictionary, and in Section 3 we show some applications to music transcription, for both synthesized harpsichord music and real piano music. Finally, in Section 4 we mention the application of sparse representations to source separation, before concluding.

2 Finding Sparse Representations

2.1 Linear Generative Model

Suppose that we have a sequence of observation vectors $\mathbf{x} = (x_1, \ldots, x_m)^T$ where we assume that each representation vector $\mathbf{x}$ can be approximately represented using a linear generative process

$$\mathbf{x} \approx \mathbf{A}\mathbf{s} = \sum_{j=1}^{n} \mathbf{a}_j s_j \tag{1}$$

where $\mathbf{A}$ is an $m \times n$ *dictionary* matrix, and $\mathbf{s}$ is a vector of source components s_j. We typically interpret (1) as telling us that $\mathbf{x}$ is approximately given by a linear superposition of scaled basis vectors $\mathbf{a}_j$, with the corresponding scaling coefficients given by s_j. In our musical interpretation, $\mathbf{x}$ might be a short-time Fourier transform (STFT) power spectrum, approximately composed of scaled amounts of the spectra $\mathbf{a}_j$ of the musical notes available in the piece. The task is then to infer the amounts s_j of each note present, given the STFT power spectrum $\mathbf{x}$.

If the dictionary $\mathbf{A}$ is known, and we have the same number of source components s_j as the number of observation components x_i (i.e. $n = m$), then if $\mathbf{A}$ is invertible we can exactly solve (1) for $\mathbf{s}$ giving $\mathbf{s} = \mathbf{A}^{-1}\mathbf{x}$. If we have fewer source components than observations ($n < m$), we cannot guarantee to find an exact solution to (1), but we can find a least square approximation using the Moore-Penrose Pseudoinverse $\mathbf{A}^{\dagger} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$

to give $\hat{\mathbf{s}} = \mathbf{A}^{\dagger}\mathbf{x}$ [14]. If the number of source components is more than the number of observations, $n > m$, then we have an *overcomplete* system. In this case, and if $\mathbf{A}$ has full rank m, there is a whole $(n-m)$-dimensional subspace of possible solution vectors $\mathbf{s}$ which solve $\mathbf{x} = \mathbf{A}\mathbf{s}$. We could, for example, choose any m linearly independent columns from $\mathbf{A}$, and form those into the $m \times m$ matrix $\tilde{\mathbf{A}}$. This would give us $\tilde{\mathbf{s}} = \tilde{\mathbf{A}}^{-1}\mathbf{x}$ for the vector of corresponding elements of $\mathbf{s}$, with the remaining elements of $\mathbf{s}$ set to zero. However, none of these approaches yet incorporate our requirement for the source components s_j to be *sparse*.

2.2 Inference of Source Components

To build in the required sparsity, we invoke a probabilistic approach [21, 3]. We assume the source components s_j are independent, each with a probability density $p(s_j)$ peaked around zero. We suppose the observation vector $\mathbf{x}$ is generated according to

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{e} \tag{2}$$

where $\mathbf{e} = (e_1, \ldots, e_m)^T = \mathbf{x} - \mathbf{A}\mathbf{s}$ is a random vector of zero mean additive Gaussian noise. This implies a conditional density for $\mathbf{x}$ of

$$p(\mathbf{x}|\mathbf{A}, \mathbf{s}) = \sqrt{\frac{\det \Lambda_{\mathbf{e}}}{(2\pi)^m}} \exp\left(-\frac{1}{2}\mathbf{e}^T \Lambda_{\mathbf{e}} \mathbf{e}\right) \tag{3}$$

where $\Lambda_{\mathbf{e}} = \langle \mathbf{e}\mathbf{e}^T \rangle^{-1}$ is the inverse noise covariance.

With $\mathbf{A}$ and $\mathbf{s}$ assumed independent, for the maximum a posteriori (MAP) estimate $\hat{\mathbf{s}}$ of $\mathbf{s}$ given $\mathbf{A}$ and $\mathbf{x}$ we need

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} \log p(\mathbf{s}|\mathbf{A}, \mathbf{x}) \tag{4}$$

$$= \arg\max_{\mathbf{s}} \log p(\mathbf{x}|\mathbf{A}, \mathbf{s}) + \log p(\mathbf{s}) + \text{constant}. \tag{5}$$

Typically we assume equal variance noise $\langle \mathbf{e}\mathbf{e}^T \rangle = \sigma_e^2 \mathbf{I}$ giving

$$\log p(\mathbf{x}|\mathbf{A}, \mathbf{s}) = -\frac{1}{2\sigma_e^2}\|\mathbf{e}\|_2^2 + \text{constant} \tag{6}$$

and the sources components are independent with joint density $p(\mathbf{s}) = \prod_j p(s_j)$ leading to

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} -\frac{1}{2\sigma_e^2}\|\mathbf{x} - \mathbf{A}\mathbf{s}\|_2^2 + \sum_j \log p(s_j) \tag{7}$$

where the priors $p(s_j)$ are assumed to be more strongly peaked and heavy-tailed than a Gaussian. Equation (7) can be interpreted as a trade-off between preserving information (minimizing $\|\mathbf{x} - \mathbf{A}\mathbf{s}\|_2^2$) and maximizing sparsity (minimizing $\log p(s_j)$) [27]. Equation (7) can be solved using gradient descent, and we have also used a special algorithm designed for sharply-peaked priors [3].

A common choice for $p(s_j)$ is to use a Laplacian prior $p(s_j) \propto e^{-|s_j|}$, so that $\log p(s_j) = -|s_j| + \text{constant}$, giving the special case [23, 9]

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{s}\|_2^2 + \sigma_e^2 \sum_j |s_j| \tag{8}$$

$$= \arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{s}\|_2^2 + \sigma_e^2 \|\mathbf{s}\|_1 \tag{9}$$

which has a particularly convenient structure that can be solved by modern quadratic programming methods [9, 13].

As the assumed noise variance σ_e^2 is increased, these algorithms have a greater tendency to reduce the representation size $\|\mathbf{s}\|_1$ while permitting increased error $\|\mathbf{x} - \mathbf{A}\mathbf{s}\|_2^2$. This leads to a 'shrinkage' effect in the representations [18] related to the Lasso regression method [32].

2.3 Dictionary Learning

In some cases the dictionary $\mathbf{A}$ is given, such as an overcomplete ($n > m$) dictionary composed of unions of orthonormal bases [16]. However, in many of the cases we are interested in we wish to construct a dictionary $\mathbf{A}$ which is adapted to a set of observed vectors $\mathbf{X} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots]$. We might like to find the maximum likelihood estimate

$$\hat{\mathbf{A}}_{\mathrm{ML}} = \arg\max_{\mathbf{A}} \langle \log p(\mathbf{x}|\mathbf{A}) \rangle_{\mathbf{X}} \tag{10}$$

where $\langle \cdot \rangle_{\mathbf{X}}$ represents the mean over the set of observations $\mathbf{X}$. There are practical difficulties with integrating out the hidden variables $\mathbf{s}$ in $p(\mathbf{x}|\mathbf{A}) = \int p(\mathbf{x}|\mathbf{A}, \mathbf{s}) p(\mathbf{s}) \mathrm{d}\mathbf{s}$ [21]. Under certain assumptions, gradient ascent methods lead to update rules such as those by Olshausen and Field [27]

$$\mathbf{A} \leftarrow \mathbf{A} + \eta \langle (\mathbf{x} - \mathbf{A}\hat{\mathbf{s}})\hat{\mathbf{s}}^T \rangle_{\mathbf{X}} \tag{11}$$

or Lewicki and Sejnowski [23]

$$\mathbf{A} \leftarrow \mathbf{A} + \eta \mathbf{A} \langle \gamma(\hat{\mathbf{s}})\hat{\mathbf{s}}^T - \mathbf{I} \rangle_{\mathbf{X}} \tag{12}$$

where $\gamma(\mathbf{s}) = -\nabla \log p(\mathbf{s})$ is the negative gradient of the log prior. We have ourselves introduced a 'decay when active' modification to (12) for priors including an 'exactly zero' element [3]. For details of other dictionary learning algorithms see e.g. [21].

3 Sparse Representations for Music Transcription

3.1 Sparse Coding of Synthesized Spectra

The sparse coding approach outlined above can discover a dictionary and a sparse representation that show a correspondence to the activity of notes

playing during a musical piece. In one experiment, we analysed a MIDI synthesized version of Bach's *Partita in A Minor, BWV827*, sampled from audio at 11025 Hz, digitized at 16 bits per sample, with the signal amplitude normalized over a 5 s timescale. STFT frames of 512 samples (46ms) with 50% overlap were generated from the sampled audio, with the magnitudes of the first 256 STFT bins of each frame forming the observed vector $\mathbf{x}$. We chose a 'sparsified Laplacian' prior $p(s_j)$ and a corresponding tailored 'active set' optimizer to find the sparse solutions to (7). For the dictionary, a 256×96 matrix $\mathbf{A}$ was initialized to a diagonal matrix with 1s on the diagonal, and a mixture of algorithms (11) and (12) were used to learn the dictionary. For more details of the method see [3].

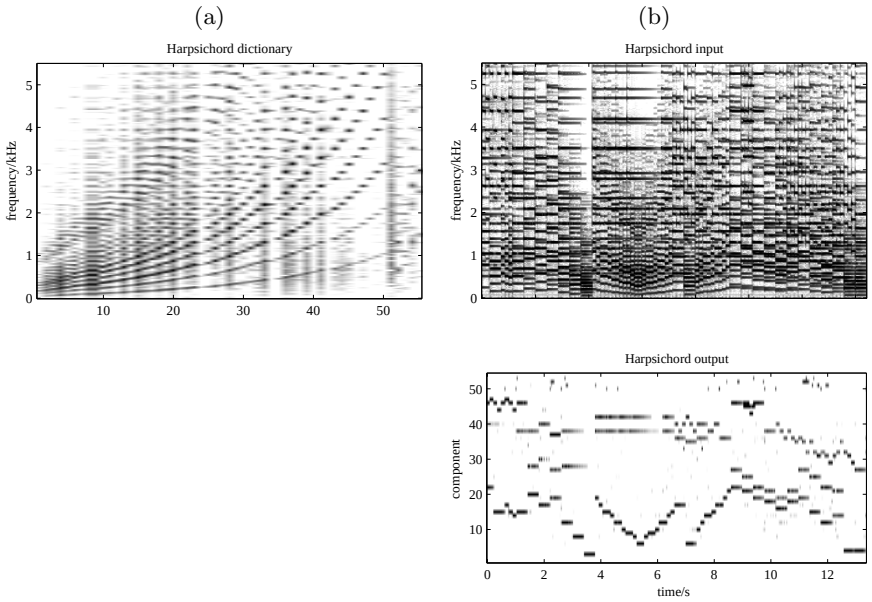

Fig. 1. Dictionary matrix $\mathbf{A}$ after learning (a), with (b) the original spectrum sequence $\mathbf{x}$ (top) decomposed to the sparse representation $\hat{\mathbf{s}}$ (bottom).

After learning, some of the dictionary vectors had decayed to zero, leaving 54 non-zero vectors (Fig. 1(a)). Fig. 1(b) clearly shows that the spectrum sequence $\mathbf{x}$ has been decomposed into a much sparser sequence $\hat{\mathbf{s}}$. On average, the source components are non-zero about 5% of the time. Comparing to the original MIDI, 94.3% of the 4684 notes in the evaluation set were correctly detected (allowing for 50ms tolerance), while 2.2% of the notes were false positives that were not present in the original. Fig. 2 shows an example of a spectrum corresponding to a three-note chord being approximated by a weighted sum of dictionary vectors.

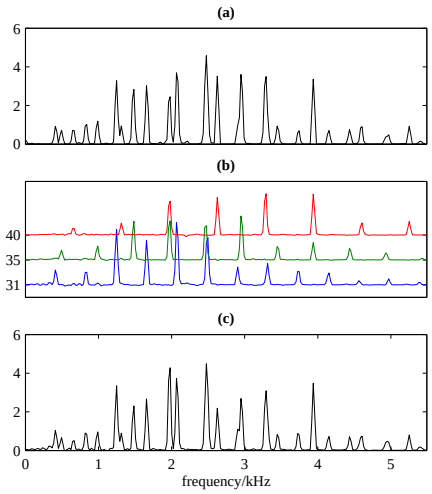

Fig. 2. A three note chord original spectrum (a) decomposed into (b) three weighted dictionary spectra, which combine to give the reconstruction (c).

3.2 Real Piano

We used a similar method to investigate sparse representations of real piano recordings [3]. For these real recordings, we found that individual notes were no longer represented mostly as single vectors, as in the synthesized harpsichord, but that a larger dictionary (e.g. 256×256) would allocate several dictionary vectors to represent each note. Fig. 3 shows the spectra of one

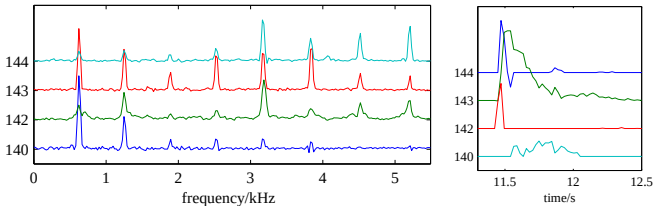

Fig. 3. Pitch group of E♭5 obtained from sparse coding of real piano spectra, showing (left) the spectra of the pitch group elements, and (right) the activation of each element during playing of a note.

of these 'pitch groups'. Some musical instruments such as the piano have a tendency to produce 'bright' notes, containing high frequencies, when first played, which change in timbre (spectral content) as the notes progress. It appears that the pitch groups are being used to represent this changing spectral content as each note is played (Fig. 3) [3].

3.3 Sparse Coding Variations

Assuming that power spectra are constrained to be positive, and that notes make a positive contribution to the total power spectra, we can build in further constraints. In fact, it is possible to tackle this problem using the positivity condition alone, by searching for positive matrices **A** and **S** which approximately factorize the observation sequence matrix using *nonnegative matrix factorization* (NMF) [22]. Lee and Seung [22] give algorithms to find a pair of nonnegative matrices **A** and **S** in

$$\mathbf{X} \approx \mathbf{AS} \tag{13}$$

that minimize some distortion between **AS** and **X**. Hoyer [17] added a 'sparsity' factor to the optimization that tends to produce nonnegative matrix factors with sparse **S**.

For the particular case of power spectra, we also developed a non-negative sparse coding (NNSC) method by constructing a slightly different generative model [2]. Here we consider that a set of time-dependent variances is generated according to

$$\mathbf{v} = \mathbf{As} \tag{14}$$

where $\mathbf{v} = (v_1, \ldots, v_m)$ is a vector of power spectrum variances, **A** is a matrix whose column vectors are the power spectra of the different sources, and **s** is the vector of source strengths. The observed power spectrum bin value x_i is the mean square of $d = 2$ (real plus imaginary) frequency bin variables with variances v_i. Thus x_i has a gamma distribution, with probability density [2]

$$p(x_i|v_i) = \frac{1}{x_i \Gamma(d/2)} \left(\frac{d}{2} \cdot \frac{x_i}{v_i} \right)^{d/2} \exp\left(-\frac{d}{2} \cdot \frac{x_i}{v_i} \right) \tag{15}$$

with $d = 2$, and where $\Gamma(\cdot)$ is the gamma function. To find the MAP estimate $\hat{\mathbf{s}}$ we derive a multiplicative learning rule [2, 30]

$$s_j \leftarrow s_j \frac{\sum_i (a_{ij}/v_i)(x_i/v_i)}{(2/d)\phi(s_j) + \sum_i (a_{ij}/v_i)} \qquad 1 \le j \le n \tag{16}$$

where $\phi(s_j) = -\frac{\mathrm{d}}{\mathrm{d}s_j} \log p(s_j)$. Similarly, to search for a maximum likelihood estimate for the dictionary we eventually derive the learning rule

$$\text{Step 1:} \quad a_{ij} \leftarrow a_{ij} \left(\frac{\langle (x_i/v_i)(s_j/v_i) \rangle_{\mathbf{X}}}{\langle s_j/v_i \rangle_{\mathbf{X}}} \right)^{\eta} \qquad 1 \le i \le m, 1 \le j \le n \tag{17a}$$

$$\text{Step 2:} \quad \mathbf{a}_j \leftarrow \mathbf{a}_j / ||\mathbf{a}_j||_2 \qquad 1 \le j \le n \tag{17b}$$

where $0 < \eta < 1$ is an update factor. The second step ensures the columns of **A** retain unit 2-norm [2].

As an alternative to the frequency-domain sparse coding methods, we have also developed a time-domain shift-invariant sparse coder (SISC) [6]. Here we

return to the mixing model (2), but this time $\mathbf{x}$ is a time-domain frame and the dictionary matrix $\mathbf{A}$ contains time-shifted versions of 'mother' dictionary vectors. Due to the shift-invariant nature of $\mathbf{A}$ we can take advantage of Fast Convolution. Nevertheless, the large size of the model does mean that we typically need to use some heuristic methods, such as a subset selection step, to speed up our searching. For full details see [6].

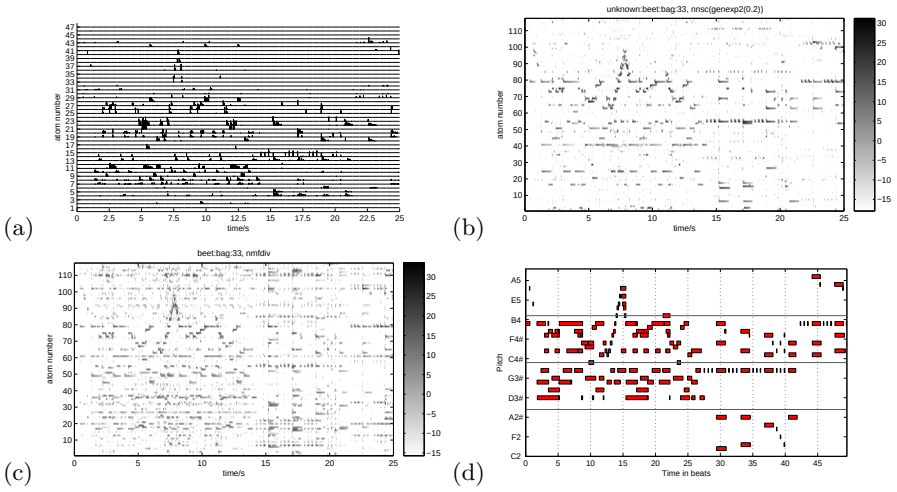

Fig. 4. Decomposition coefficients for the first 25s of the piece, found with (a) time-domain sparse coding, (b) the spectral-domain non-negative sparse coding, and (c) nonnegative matrix factorization with (d) showing the original MIDI score for comparison (50 beats = 25s at 120 bpm). Dictionary atoms have been ordered semi-automatically, and values corresponding to unpitched dictionary elements are omitted. For the time-domain method, the activities are rectified for display.

Figure 4 compares these alternative sparse decomposition methods applied to a recording of Beethoven's Bagatelle, Opus 33 No. 1 in E♭ Major. To give us a MIDI reference, but at the same time producing a 'real' piano sound, we used a MIDI-controlled acoustic piano to produce the musical audio.

We can see that all of these methods generate some decomposition of the signal that is related to the original MIDI sore. We also find both the NNSC and time domain SISC methods result in 'pitch groups' of several dictionary atoms corresponding to any given note. The time-domain representation produces sparse 'spikes' in time as well as across dictionary elements, reminiscent of the spikes found in biological neural systems [28]

4 Source Separation

We can also use sparse coding methods for source separation problems. In the simplest case we have our mixing model (2) where $\mathbf{x}$ is our vector of m observations, $\mathbf{A}$ is an unknown mixing matrix we wish to identify, and $\mathbf{s}$ is a vector of n source signals we wish to extract. For the noiseless case ($\mathbf{e} = 0$) and with $n = m$ we can use independent component analysis (ICA) [4] to identify and invert the matrix.

However, if there are more sources than observations ($n > m$) then again a linear matrix inversion is insufficient to recover the source vector $\mathbf{s}$. Instead, we can use a sparse representation approach. For example, by transforming into the frequency domain, many audio signals (particularly speech) have a very sparse representation. Often only one source will have significant activity in a given time-frequency (TF) bin, allowing the sources to be extracted [7]. When one source always dominates, we can use *time-frequency masking* to extract each of the sources, whereby the activity in each TF bin is allocated to the source that dominates in that bin. The sources are then reconstructed from the active TF bins allocated to that source, with zero assumed for all other TF bins for that source. One well-known method using this approach is the DUET algorithm [20]. For further discussion of this and other audio source separation methods see e.g. [26, 33].

The success of these methods depends on the sparsity of the representation. One way to get as sparse a representation as possible is to learn a set of transforms which are tailored to the data: for separation of convolved sources (those with time delay and reverberation) we have used a sparse ICA algorithm [1, 19] to directly learn basis vectors from the stereo signals. The

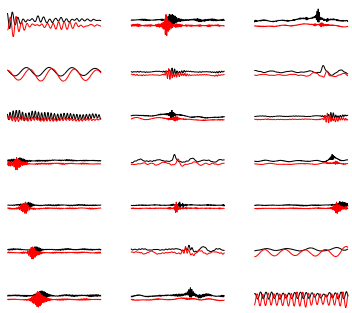

Fig. 5. Examples of stereo basis functions extracted with the sparse ICA algorithm.

relative delays visible on the basis vectors examples in Fig. 5 illustrate that the information about the delays in the mixing process has been incorporated into the basis vectors.

We are also investigating methods which promote sparsity but avoid having to learn the dictionary matrix $\mathbf{A}$. For example, we have investigated the

use of a cosine packet tree adapted to maximize the sparsity of the signal representation. We found that this can give better separation results when compared to e.g. the STFT as used in the DUET algorithm [25].

Finally, we should mention that sparse time-frequency or transform methods have also been applied to audio source separation from single channel audio, relying on the non-negativity of the spectrum. This is a very challenging problem, although some progress has been made for simple musical audio signals [31, 34, 35].

5 Conclusions

Sparse representations are becoming an increasingly useful tool in the analysis of musical audio signals. With musical signals generated from a sparse process, having only a small number of 'active' notes at any one time, it is natural to try to find a representation in which this sparsity can be exploited.

For a linear generative model $\mathbf{x} = \mathbf{A}\mathbf{s}$ we have described a probabilistic approach to estimating sparse source component $\mathbf{s}$, and to learning a suitable dictionary matrix $\mathbf{A}$ from the data. We have described the application of this sparse coding method to the problem of music transcription, and have seen that there are a variety of related sparse coding methods available, including extraction of spiking representations by exploiting sparsity in the time domain. We have also mentioned the application of sparse representations in audio source separation from multiple or single channels.

There is much further work still to be done in this interesting and challenging area, and we believe that sparse representations have significant potential for further applications in analysis, transcription, and encoding of musical audio.

6 Acknowledgements

This work was partially supported be Grants GR/R54620/01, GR/S75802/01, GR/S82213/01, GR/S85900/01 and EP/D000246/1 from the Engineering and Physical Sciences Research Council, and by EU-FP6-IST-507142 project SIMAC (Semantic Interaction with Musical Audio Contents).

References

1. S. A. Abdallah and M. D. Plumbley. Application of geometric dependency analysis to the separation of convolved mixtures. In C. G. Puntonet and A. Prieto, editors, *Independent Component Analysis and Blind Signal Separation: Proceedings of the Fifth International Conference, ICA 2004, Granada, Spain, September 2004*, pages 540–547. Springer, Berlin, September 22-24 2004. LNCS 3195.

2. S. A. Abdallah and M. D. Plumbley. Polyphonic transcription by non-negative sparse coding of power spectra. In C. L. Buyoli and R. Loureiro, editors, *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004), Barcelona, Spain*, pages 318–325. Audiovisual Institute Popeu Fabra University, October 10–14 2004.
3. S. A. Abdallah and M. D. Plumbley. Unsupervised analysis of polyphonic music by sparse coding. *IEEE Transactions on Neural Networks*, 17(1):179–196, Jan. 2006.
4. A. J. Bell and T. J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.
5. J. P. Bello. *Towards the Automated Analysis of Simple Polyphonic Music: A Knowledge-Based Approach*. PhD thesis, Department of Electronic Engineering, Queen Mary, University of London, 2003.
6. T. Blumensath and M. E. Davies. Unsupervised learning of sparse and shift-invariant decompositions of polyphonic music. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, volume 5, pages V:497–V:500, May 2004.
7. P. Bofill and M. Zibulevsky. Underdetermined blind source separation using sparse representations. *Signal Processing*, 81(11):2353–2362, Nov. 2001.
8. A. S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, Cambridge, MA, 1990.
9. S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.
10. D. P. W. Ellis. *Prediction-Driven Computational Auditory Scene Analysis*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, June 1996.
11. D. J. Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994.
12. M. A. T. Figueiredo. Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1150–1159, Sept. 2003.
13. J.-J. Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, 50(6):1341–1344, 2004.
14. G. H. Golub and C. F. van Loan. *Matrix Computations*. North Oxford Academic, Oxford, England, 1983.
15. I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, 45(3):600–616, Mar. 1997.
16. R. Gribonval and M. Nielsen. Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, 49(12):3320–3325, December 2003.
17. P. O. Hoyer. Non-negative sparse coding. In *Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing)*, pages 557–565, Martigny, Switzerland, 2002.
18. A. Hyvärinen. Sparse code shrinkage: Denoising of non-Gaussian data by maximum-likelihood estimation. *Neural Computation*, 11(7):1739–1768, 1999.
19. M. G. Jafari, S. A. Abdallah, M. D. Plumbley, and M. E. Davies. Sparse coding for convolutive blind audio source separation. In *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Source Separation (ICA 2006), Charleston, SC, USA*, pages 132–139, 5-8 March 2006.

20. A. Jourjine, S. Rickard, and Ö. Yilmaz. Blind separation of disjoint orthogonal signals: Demixing N sources from 2 mixtures. In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP'2000)*, volume 5, pages 2985–2988 vol.5, 2000.
21. K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15:349–396, 2003.
22. D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 21 October 1999.
23. M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12:337–365, 2000.
24. K. D. Martin. *Sound-Source Recognition: A Theory and Computational Model*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, June 1999.
25. A. Nesbit, M. E. Davies, M. D. Plumbley, and M. B. Sandler. Source extraction from two-channel mixtures by joint cosine packet analysis, 2006. Submitted for publication.
26. P. D. O'Grady, B. A. Pearlmutter, and S. T. Rickard. Survey of sparse and non-sparse methods in source separation. *International Journal of Imaging Systems and Technology*, 15(1):18–33, 2005.
27. B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive-field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.
28. B. A. Olshausen and D. J. Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14:481–487, 2004.
29. M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler. Automatic music transcription and audio source separation. *Cybernetics and Systems*, 33(6):603–627, Sept. 2002.
30. M. D. Plumbley, S. A. Abdallah, T. Blumensath, and M. E. Davies. Sparse representations of polyphonic music. *Signal Processing*, 86(3):417–431, March 2006.
31. P. Smaragdis. Non-negative matrix factor deconvolution: Extraction of multiple sound sources from monophonic inputs. In *Independent Component Analysis and Blind Signal Separation: Proceedings of the Fifth International Conference (ICA 2004)*, pages 494–499, Granada, Spain, September 22–24 2004.
32. R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B*, 58(1):267–288, 1996.
33. E. Vincent, M. G. Jafari, S. A. Abdallah, M. D. Plumbley, and M. E. Davies. Blind audio source separation. Technical Report C4DM-TR-05-01, Centre for Digital Music, Queen Mary University of London, 24 November 2005.
34. T. Virtanen. Separation of sound sources by convolutive sparse coding. In *Proceedings of the ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing (SAPA 2004)*, Jeju, Korea, 3 October 2004.
35. B. Wang and M. D. Plumbley. Musical audio stream separation by non-negative matrix factorization. In *Proceedings of the DMRN Summer Conference, Glasgow, UK*, 23–24 July 2005.