TOWARDS MUSICAL INSTRUMENT SEPARATION USING MULTIPLE-CAUSE NEURAL NETWORKS

*J Klingseisen*and M D Plumbley†*

Department of Electronic Engineering
King's College London, Strand, London WC2R 2LS, UK
Email: mark.plumbley@kcl.ac.uk

ABSTRACT

Over the last few years, interest has been growing in neural network circles in the separation of independent sources, using techniques such as blind source separation and independent component analysis (ICA). A related technique is the 'Multiple-Cause Model' of Saund [Neural Computation, 7, 51-71, 1995]. In this technique, a neural network is trained to model the observed pattern as a composition of several underlying 'causes', in contrast to the more traditional 'winner-takes-all' neural networks which can handle only a single 'cause'. In this paper, we report on experiments working towards the use of a simple multiple-cause model with constraints to separate different instruments and notes from audio spectral representations.

1. INTRODUCTION

Human perception of sounds is much more advanced than any technical system so far created. A human listener is able to distinguish different tones in a complex sound structure such as a number of different human voices or musical instruments.

In this paper, we report on an approach where we attempt to separate musical sounds using Saund's Multiple Cause Model [17]. This model searches for representations of the underlying causes of the input data, together with amounts of each 'cause', which take account of the input data as closely as possible. Different notes, such as a violin playing the note 'A', are to be presented in the form of an audio signal. The audio signal would be pre-processed using a suitable transform (e.g. fast fourier transform (FFT) or wavelets) before being passed to the multiple cause model. The goal of the system would be to model and recognize these tones, without any prior knowledge, and without an explicit 'teacher'.

*JK was supported by the EC via the Socrates placement scheme.

† Corresponding author

Starting from the original multiple-cause model, we gradually work towards analysis of audio sounds in a step-by-step manner. First we relax the binary constraints of the original multiple-cause model, by testing the system on artificial continuous-valued patterns, but with a binary 'volume'. We then test these patterns with continuous volumes on the unit interval. We then use patterns derived from spectra of audio signals, but combined using linear addition. Finally, we use audio signals, added in the time domain, with the resulting signal passed through an FFT.

2. SINGLE CAUSE AND MULTIPLE-CAUSE MODELS

Many different types of neural network models have been developed for pattern recognition applications. For this application, we are looking for an *unsupervised* neural network model that will learn to identify and separate sounds from musical instruments, but without any 'ground-truth', i.e. without having to be told the identity or extent of the musical instruments during training.

One of the more popular types of unsupervised neural network is the Kohonen self-organized feature map (SOM). In a well known speech processing application, this network has been used to learn a feature map of Finnish vowels from FFT spectra [10]. However, the output units of the Kohonen SOM operate in a winner-take-all mode, so that an output unit attempts to represent all of the input signal as if it were caused by a single factor. While this is reasonable for phoneme recognition from one speaker at a time, or a single monophonic musical instrument, this would not be suitable for separating multiple causes, such as two or more musical instruments playing at once.

At the other extreme from single-cause winner-take all networks are principal component analysis (PCA) and principal subspace networks [14]. These networks can be viewed as attempting to represent the input

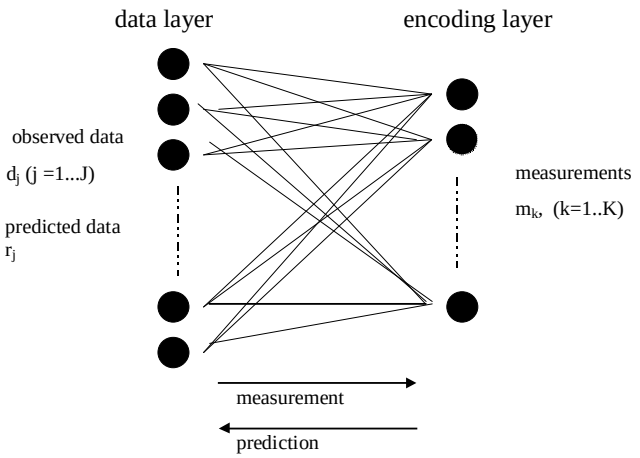

Figure 1: Multiple cause model architecture.

signal as a linear sum of orthogonal factors. These networks can be justified from information theory, and they have well-behaved global convergence [16].

However, PCA networks typically require the output units $\mathbf{y} = [y_1, \ldots, y_m]^T = \mathbf{W}^T\mathbf{x}$ to form an uncorrelated orthonormal representation set for the input data $\mathbf{x}$. These networks would therefore only find orthogonal causes. We would not expect these networks to extract patterns which have a significant overlap, such as we might expect for spectra of musical instruments.

More recently, much interest has focused on the technique of independent component analysis (ICA), particularly following the information-theoretical approach introduced by Bell and Sejnowski [1]. This technique uses output non-linearities to form a non-orthogonal (although still linear) separation of independent components (i.e. causes), and has been applied to separation of sounds [1, 11] and images [2].

For this work, we would like to be able to analyze audio signals from a frequency domain representation, from just one microphone. For ICA, we typically need N microphones to separate N sources, although recent work on overcomplete basis functions has indicated that this restriction can be relaxed [12]. However, in this paper we will use an alternative approach based on Saund's multiple-cause model.

3. SAUND'S MULTIPLE-CAUSE MODEL

The multiple-cause model [17] (Figure 1) is designed to cope with input data which is composed of several causes active at the same time. This network does not operate in a simple feed-forward manner: rather the encoding layer and connections are adjusted until the encoding forms a good reconstruction of the observed data. The model operates as follows.

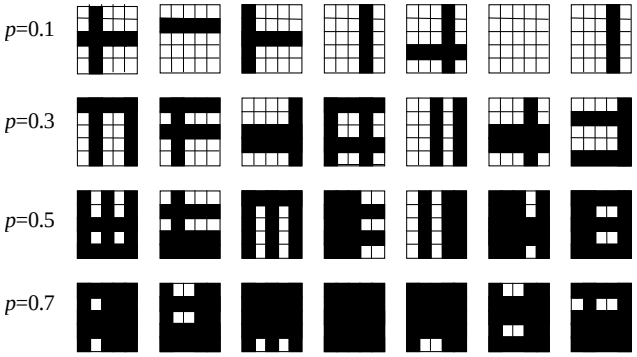

Figure 2: Bars problem.

An input data vector $\mathbf{d} = (d_1, \ldots, d_J)$ is presented to the input. This is compared to a prediction vector $\mathbf{r} = (r_1, \ldots, r_J)$, which is calculated from a measurement output vector $\mathbf{m} = (m_1, \ldots, m_K)$ and a weight matrix c_{ij}, (which represents the causes that the network is trying to model), via a mixing model. For details of the algorithms used, see [17, 9].

As one example, the multiple-cause model was originally demonstrated on the Bars problem, introduced by Földiák [6]. Here an image is composed of a white (0) background with horizontal and vertical black (1) bars, each of which may appear with some probability p. Where two bars overlap, black (1) is the result: this is a non-linear OR-type 'write-black' imaging model (Figure 2).

For this problem, the prediction r_j is calculated according to

$$r_j = 1 - \prod_k (1 - c_{jk} m_k) \qquad (1)$$

yielding a soft-or function.

On each input pattern presentation, the network searches for the measurement set m_k which minimises some error function, e.g. $g = \sum_i (d_j - r_j)^2$ (other measures such as the negative log likelihood can also be used). Over many presentations, the weight matrix c_{ij} is adjusted to minimize the expected error.

4. DEALING WITH NON-BINARY DATA

Our musical application differs from the binary version of the multiple-cause model in the following ways:

1. The mixing of sounds is approximately linear in power spectrum (for independent sources), instead of OR (write-black) mixing;

2. The basic patterns (spectra) corresponding to the sources are continuous, rather than binary;

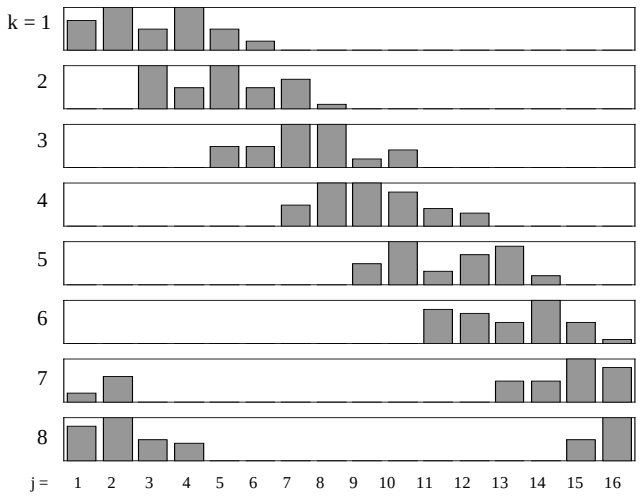

Figure 3: Continuous basic patterns

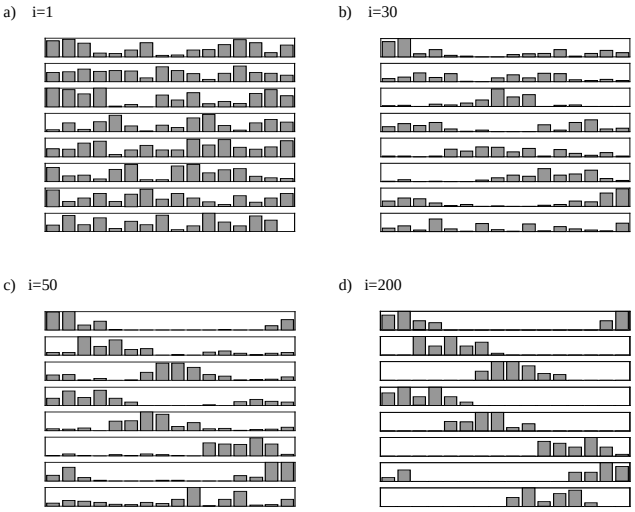

Figure 4: Weight matrix during the learning process.

3. Each basic pattern can have a continuous amount (volume), not just a binary on/off value.

4.1. Linear mixing of grey-level patterns

Our first step was to change the OR mixing model for a linear mixing model $r_j = \sum_k c_{jk} m_k$ and to use grey-level patterns in the range $[0, 1]$. For example, we tried k=8 patterns with J=16 inputs (Figure 3) mixed with probability p of each pattern appearing in the input.

Some prior knowledge, in the form of constraints, can be used to help feedback networks such as this. For example, Harpur and Prager [7] describe a network with threshold-linear output units, where the network finds basis vectors that reconstruct the input from non-negative amounts of these factors. Charles and Fyfe [3] introduced a related network with constraints of positive weights and/or outputs to analyze the bars problem, gaussian mixtures, and sine waves. We also use these positivity constraints, since we expect a positive (or zero) volume of each sound, and a positive power spectrum to be associated with each note in our final system.

In the patterns we created, we also know that the maximum value for each pattern element c_{jk} is 1, and the mean value for each pattern,

$$\bar{c}_k = \frac{1}{J} \sum_{j=1}^{J} c_{jk} \qquad (2)$$

should be between 0.2 and 0.3 for all patterns k. If any mean $\bar{c}_k$ falls outside this range, the whole vector $\mathbf{c}_k$ is scaled up or down by a small factor (we used 1.1). We also observed that this constraint helps to prevent a given weight vector $\mathbf{c}_k$ from representing more than one basic pattern, and is particularly useful when the data has been generated with high probability of each pattern occurring ($p > 0.5$).

Using a simple steepest-descent search, typical learning time for these grey patterns, consisting of 8 patterns, each of 16 components, is 20 minutes to an error level of 0.1 using Matlab on a Pentium II (350 MHz). Figure 4 shows the evolution of weight vectors as this network learns, for $p = 0.4$. Separation of 8 patterns with overlap of up to about 50% is fairly reliable. Above this, some patterns come to be recognized as a partial activation of other patterns, leaving a small error that is insufficient to drive the learning algorithm [9].

Varying the probability of appearance of each pattern had a significant effect on the learning time for each dataset. For a given error threshold, we observed that both high and low probabilites of occurrence gave rise to longer learning times than probabilities around $p = 0.4$ to $p = 0.6$ (Figure 5).

4.2. Continuous pattern volumes

So far the measurements m_k (volumes) have all been binary. For real music signals we would need to release this constraint to allow constantly varying volumes. To this end, we released some of the measurements m_k (33%, 50% and 100%) so that they could vary between 0 and 1, while the c_{jk} values were still constrained to lie in the interval $[0, 1]$. Note that since the predicted input is given by $r_j = \sum_k c_{jk} m_k$ the problem of identifying the c_{jk} and m_k parameters is clearly underdetermined, since any scaling on c and inverse scaling on m will leave the result unchanged. These range con-

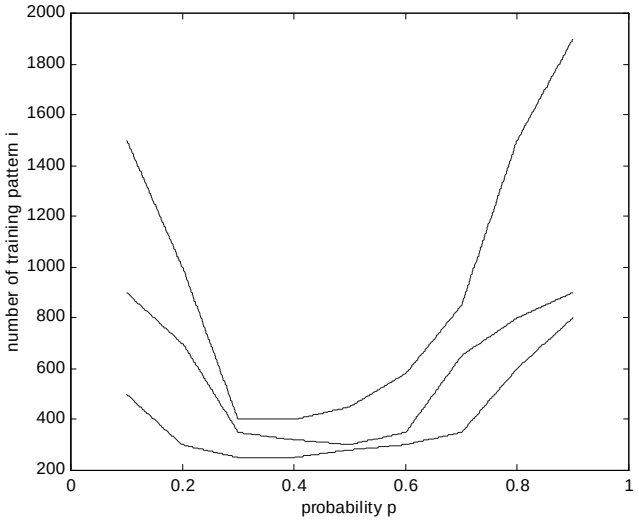

Figure 5: Dependence of learning time on pattern probability (p) and volume variability (lower curve: 33%, middle curve: 50%, upper curve: 100%).

straints help to overcome this problem.

In our experiments, we found that the more of the measurements that were allowed to vary, the longer was the time taken to learn the patterns. Fastest learning is obtained when as many measurements as possible are fixed as 0 or 1 only, and the probability of occurrence is in the approximate interval $p = 0.3$ to $p = 0.6$ (Figure 5).

5. MUSIC-BASED SIGNALS

In our next step towards audio signals, we next applied the multiple-cause model to artificial spectra produced from synthesized sounds. In the first instance, we trained the model on mixtures of spectra, downsampled to 30 bins, of a synthesized clarinet playing one of 8 notes (G3, C4, A3, D4, F4, G4, A4, E4) (Figure 6). The training set was composed of linear additions of these basic spectra (not mixed in the time domain), with probability $p = 0.4$ of each spectrum.

About 800 presentations of training patterns were necessary for successful learning. This was also tested on notes from a violin and an alto recorder, with similar results.

Separation of patterns composed from spectra of different synthesized instruments playing the same note was also attempted. For six instruments, about 600 presentations were needed to separate the patterns, and about 3000 presentations for 10 instruments. This appeared to take longer to learn than the experiments with different notes on the same instrument. This is

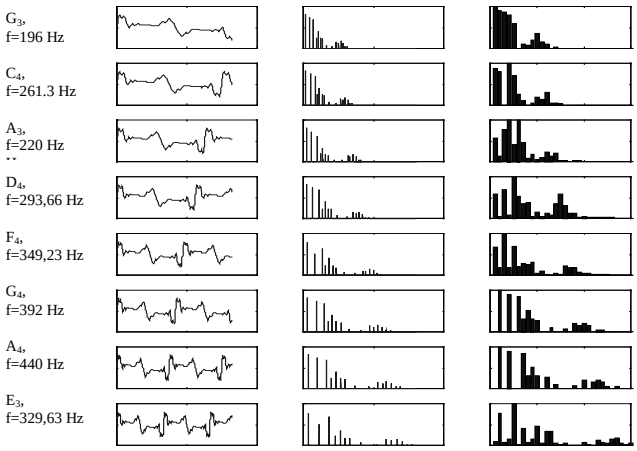

Figure 6: Eight notes played on a clarinet, showing the audio waveform, FFT, and downsampled FFT for each.

probably due to the similarity between the patterns representing the same note, particularly the alignment of the fundamental and first few harmonics into the same bins in each pattern.

Combining these two approaches, patterns composed from the spectra of three instruments (Clarinet, Oboe, Trumpet) playing each of three notes were used for the training patterns. The spectra used to form the training patterns were downsampled into log-scaled bins, with a relative scaling of $2^{1/12}$ between the bins. With synthesized instruments, different notes on the same instrument would then appear simple as shifted along this log-frequency scale, with 1 bin shift equivalent to 1 semitone (Figure 7).

A multiple-cause model with $K = 9$ measurement units learned the patterns after about 700 presentations of the training patterns. After this, the patterns were post-processed to identify which patterns were relative shifts of each other. This correctly identified that 3 instruments were used, with relative semitone shifts of (0, +2, +4), (0, +1, +5) and (0, +2, +3). Note that in the current network this does not make the learning any easier, since the relative shifts are only extracted after the basic patterns have been extracted. However, in future, it would be interesting to experiment with a non-linear mixing model that formed the prediction from shifted versions of the same underlying spectra.

6. REAL SOUNDS

In the experiments reported on so far, we analysed synthesized sounds, with artificial "spectra" composed by linear addition of underlying spectra. We also assumed that the spectra are essentially unchanged by volume or tone changes.

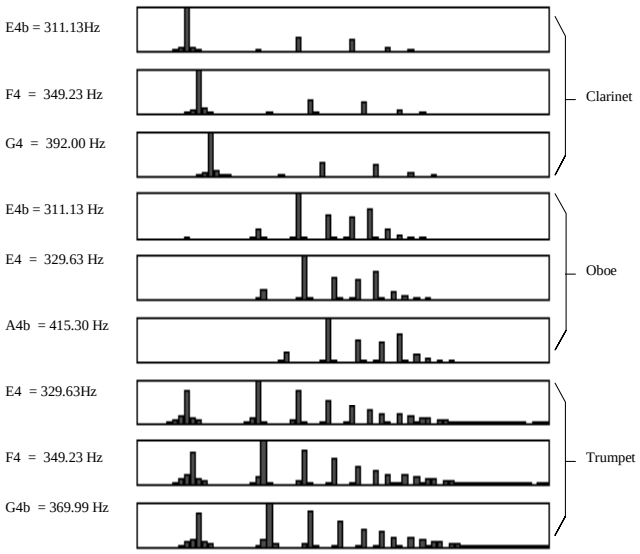

Figure 7: Log scaled spectra of three instruments each playing three notes.

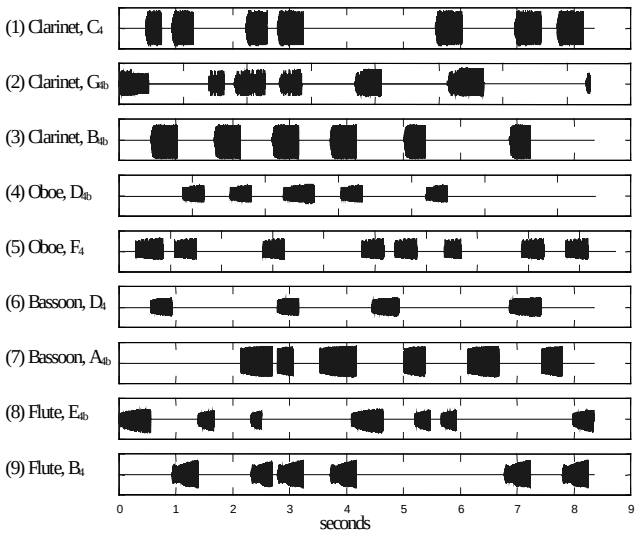

Figure 8: Waveforms of nine notes.

For the sounds of real instruments, the situation is more complicated. Volume and tone can both change the spectra, so that simple shifting or scaling is not sufficient. The spectral envelope does not stay the same for different notes, even when they belong to the same octave. Also, the spectrum changes for different volumes. For example, we observed that a loud note (fortissimo) often contains more higher harmonics than a note played quietly (pianissimo). This means that a simple linear additive multiple-cause model may not be able to cope with the full range of variability expected from a real sound source.

To control the audio signal more closely, we therefore constructed an audio signal composed of a linear time-domain addition of pulses of notes played on different instruments from the University of Iowa musical instrument samples web page [13] (Figure 8). This resulted in an audio signal of 8.3s sampled at 44.1kHz, and fourier transformed with a window width of 4096 samples, yielding 90 spectra of about 0.09s duration each.

The algorithm found most of the nine underlying patterns after 300 presentations of the set of training patterns, equivalent to 20 hour's learning using Matlab on a 350MHz Pentium II. In Figure 9 we see that the sounds have been separated such that the input sound is represented by a small number of measurement units, with other units off. This indicates that the output units have found a sparse coding Field [5], even without penalty terms that have been used to encourage sparse output distributions (see e.g. [8, 15]).

There is some instruments that are not completely separated: in particular, it seems that separate of flute and clarinet is difficult. In this example, Flute E_{4b} (input 8) and Clarinet G_{4b} (input 2) are poorly separated, with parts of each instrument found in the corresponding outputs (the label in Fig 9 indicates the closest instrument/note). Also, some of the Flute B_4 (input 9) is still mixed with the Clarinet B_{4b} (input 3), with the attack phase of the flute being 'picked up' by the Clarinet output. This difficulty may be due to the relatively pure waveforms, and therefore dominant fundamentals, that these instruments have, although more investigation is needed to confirm this.

7. DISCUSSION

In this initial work, we consider the use of appropriate mixing model and constraints to be important in the successful operation of this type of algorithm. Many of these constraints, and the flavour of algorithm used, depends on the application field.

As well as the OR mixing model used by Saund, others have proposed networks where the underlying factors compete to explain particular features, such as pixels in image analysis [4, 18]. Although these would not seem to be appropriate in our current audio problem, where we have linear mixing, it would be interesting to speculate if the phenomenon of auditory masking indicates that human hearing uses such a scheme.

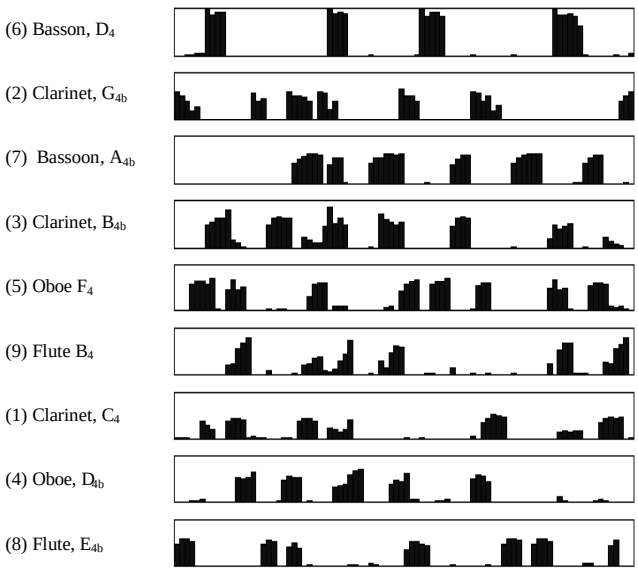

Figure 9: Activation of measurement units m_i

8. CONCLUSIONS

In this paper, we reported on a series of experiments working towards the application of Saund's multiple-cause model to the separation of audio music signals.

We introduced non-binary patterns and volumes into the model, and used a series of constraints on the range of measurement outputs and weights to help learning to be successful. The results are encouraging so far, especially considering the simplicity of the approach.

9. REFERENCES

[1] A. J. Bell and T. J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[2] A. J. Bell and T. J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37:3327–3338, 1997.

[3] D. Charles and C. Fyfe. Modelling multiple-cause structure using rectification constraints. *Network: Computation in Neural Systems*, 9:167–182, 1998.

[4] P. Dayan and R. Zemel. Competition and multiple cause models. *Neural Computation*, 7:565–579, 1995.

[5] D. J. Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994.

[6] P. Földiák. Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics*, 64:165–170, 1990.

[7] G. F. Harpur and R. W. Prager. Techniques for low entropy coding. Technical Report CUED/F-INFENG/TR. 197, Engineering Department, Cambridge University, UK, 1995.

[8] G. F. Harpur and R. W. Prager. Development of low entropy coding in a recurrent network. *Network: Computation in Neural Systems*, 7:277–284, 1996.

[9] J. Klingseisen. Audio analysis using multiple cause neural networks. Project Report. Audio & Music Technology Lab, Department of Electronic Engineering, King's College London, September 1999.

[10] T. Kohonen. The 'neural' phonetic typewriter. *IEEE Computer*, 21(3):11–22, March 1988.

[11] T.-W. Lee, A. J. Bell, and R. Orglmeister. Blind source separation of real world signals. In *Proceedings of the IEEE International Conference on Neural Networks, Houston, Texas*, pages 2129–2135, 1997.

[12] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12:337–365, 2000.

[13] University of Iowa. Musical instrument samples web page. URL http://theremin.music.uiowa.edu/~web/sound/, January 1999.

[14] E. Oja. Neural networks, principal components, and subspaces. *International Journal of Neural Systems*, 1:61–68, 1989.

[15] B. A. Olshausen and D. J. Field. Natural image statistics and efficient coding. *Network: Computation in Neural Systems*, 7:333–339, 1996.

[16] M. D. Plumbley. Lyapunov functions for convergence of principal component algorithms. *Neural Networks*, 8:11–23, 1995.

[17] E. Saund. A multiple cause mixture model for unsupervised learning. *Neural Computation*, 7:51–71, 1995.

[18] J. B. Tenenbaum and E. V. Todorov. Factorial learning by clustering features. In G. Tesauro, D. Tourestsky, and T. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 561–568, Cambridge, MA, 1995. MIT Press.