Sparse and Shift-Invariant Representations of Music

Thomas Blumensath and Michael E. Davies

Sparse and Shift-Invariant Representations of Music

Thomas Blumensath, *Member, IEEE,* and Mike Davies, *Member, IEEE*

***Abstract*—Redundancy reduction has been proposed as the main computational process in the primary sensory pathways in the mammalian brain. This idea has led to the development of sparse coding techniques, which are exploited in this article to extract salient structure from musical signals. In particular, we use a sparse coding formulation within a generative model that explicitly enforces shift-invariance. Previous work has applied these methods to relatively small problem sizes. In this paper, we present a subset selection step to reduce the computational complexity of these methods, which then enables us to use the sparse coding approach for many real world applications. We demonstrate the algorithm's potential on two tasks in music analysis: the extraction of individual notes from polyphonic piano music and single-channel blind source separation.**

***Index Terms*—Blind source separation, independent component analysis (ICA), shift-invariance, sparse coding, time–series analysis, unsupervised learning.**

I. Introduction

MANY SIGNAL processing tasks such as source separation and object recognition are fundamental operations solved successfully by the mammalian brain. Psychologically or physiologically inspired models of perception have, therefore, been proposed to solve similar problems in engineering. These methods often try to simulate and copy individual aspects of perception such as the nonlinearities in the human ear, but do not try to exploit fundamental theoretical ideas that might underlie perception and that have led to the evolution of the specific physiological and neurological systems.

One possible fundamental principle underlying the neurological processes of interpreting and recognising sensory stimuli was proposed by Barlow [1], who suggested that the main aim of mammalian primary perceptual processing is redundancy reduction. This idea has led to the development of sparse coding techniques to discover structure in natural stimuli such as images [2]–[4] and sound [5]. In these methods, redundancy reduction is achieved by representing the signal by a combination of a small number of features taken from a set of elementary waveforms called a dictionary. Mathematically, the principle of redundancy reduction can be formulated in information theoretic terms so that the problem can be analyzed using statistical methods. From such an information theoretical point of view, redundancy reduction can be seen as finding a representation in which the individual values are sparse and independent. With this set of constraints it is theoretically possible to find optimal solutions even when the dictionary is larger than the dimension of the input space. Redundancy reduction for the encoding of a certain set of stimuli is then achieved by adapting the features in the dictionary for optimal average performance. This adaptation leads to the discovery of structure in the signal, with the dictionary elements representing salient signal features.

Natural stimuli are generally located in time, and this localization has to be reflected in the representation of these stimuli. For example, natural sounds such as human language have clear time-locations of acoustic features such as vowels and plosives. The original sparse coding algorithms proposed in the literature [2]–[4] do not take account of this uncertainty of features in time and have to be adapted to incorporate such constraints. Such an adaptation can be based on the neurophysiological principles recently suggested in Rieke *et al.* [6], where a generative model of perception was proposed in which the stimulus is reconstructed by a convolution of a neural impulse train with a function describing a certain feature coded by the neuron under study. Such methods led to the development of shift-invariant sparse coding proposed in [7]–[10].

Section II reviews the concept of sparse coding and its extension to shift-invariant sparse coding. The shift-invariant sparse coding formulation does, however, not scale with the problem size, making the computational requirements for many real world problems prohibitively large. For example, in [8], the optimization problem to be solved had 6528 dimensions, while for the experiment reported in Section IV-B, the dimension of the optimization problems is 383 500, which is two orders of magnitudes higher. In order to solve such large problems, we introduce a subset selection procedure in Section III that reduces the computational demands and allows us to use the shift-invariant sparse coding method to extract salient features from musical signals in an unsupervised manner. Two applications of shift-invariant sparse coding to music are studied in Section IV. In Section IV-A, we show that the shift-invariant sparse coding model leads to the emergence of note- and score-like structures from piano recordings and in Section IV-B, we analyze a mixture of guitar and vocal and show that the found sparse representation contains enough information to separate the two sources.

II. Theoretical Background

A. Sparse Coding

In signal processing and coding, one fundamental operation is the efficient representation or approximation of a signal. The efficiency of such representations is based on the exploitation of signal features. A common and useful way to find such representations is to assume a generative signal model that describes the signal as a linear combination of atomic functions or features.

Manuscript received January 26, 2005; revised September 20, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Paris Smaragdis.

The authors are with the Digital Signal Processing Group, Department of Electronic Engineering, Queen Mary, University of London, London E14NS, U.K. (e-mail: thomas.blumensath@elec.qmul.ac.uk; michael.davies@elec.qmul.ac.uk).

Digital Object Identifier 10.1109/TSA.2005.860346

This model can be expressed algebraically as $\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\epsilon}$, where the observation vector $\mathbf{x}$ is a linear combination of the columns of the matrix $\mathbf{A}$ scaled by the coefficients in the vector $\mathbf{s}$. From our point of view, $\mathbf{A}$ is the dictionary matrix whose columns $\mathbf{a}_k$ represent the individual features to be extracted. The vector $\boldsymbol{\epsilon}$ represents observation noise or approximation error. In order to find efficient representations of a set of observations $\{\mathbf{x}\}$, both $\mathbf{A}$ and $\mathbf{s}$ have to be determined. As a measure of the efficiency of a representation, it is often suggested to use independence and sparsity of the coefficients $\mathbf{s}$ together with a term measuring the approximation error $\boldsymbol{\epsilon}$. The linear generative signal model together with these three constraints is the sparse coding model as proposed in [2]–[4].

From a Bayesian point of view, independence and sparsity can be enforced by using a factorial prior $p(\mathbf{s}) = \prod_n p(s_n)$, with $p(s_n)$ being sparse distributions as proposed by Lewicki in [4]. The sparsity constraint can further guarantee the existence of a unique solution of the problem even in the case when $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $N > M$.

With this prior formulation, learning of the matrix $\mathbf{A}$ can be achieved by finding the maximum likelihood estimate of the marginal likelihood $\mathcal{Z} = p(\mathbf{x}|\mathbf{A}) = \int p(\mathbf{x}|\mathbf{A}, \mathbf{s}) p(\mathbf{s})\, d\mathbf{s}$, which can be done by stochastic gradient optimization. The gradient can be estimated using a single data observation and, following [11], can be written as

$$\frac{\partial \log \mathcal{Z}}{\partial \mathbf{A}_{\mathrm{mn}}} = \left\langle \frac{\partial}{\partial \mathbf{A}_{\mathrm{mn}}} \log p(\mathbf{x}, \mathbf{s}|\mathbf{A}) \right\rangle_{p(\mathbf{s}|\mathbf{A}\mathbf{x})}$$

where $\langle \cdot \rangle$ denotes expectation.

Taking the derivative of $\log p(\mathbf{x}, \mathbf{s}|\mathbf{A})$ with respect to the elements in $\mathbf{A}$ and assuming a Gaussian error term $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\sigma}_\epsilon \mathbf{I})$, this can be written as [11]

$$\frac{\partial \log \mathcal{Z}}{\partial \mathbf{A}_{\mathrm{mn}}} \propto \left\langle \boldsymbol{\sigma}_\epsilon^{-1} (\mathbf{x} - \mathbf{A}\mathbf{s}) \mathbf{s}^T \right\rangle_{p(\mathbf{s}|\mathbf{A}, \mathbf{x})} \tag{1}$$

where the derivative is again with respect to the individual entries in the matrix $\mathbf{A}$.

As this expectation with respect to $p(\mathbf{s}|\mathbf{A}, \mathbf{x})$ cannot be evaluated analytically, different strategies have been proposed including Markov Chain Monte Carlo methods [12], [9] and first-order Laplace approximations [4].

Olshausen and Field [3] proposed a method that uses a delta approximation of the distribution $p(\mathbf{s}|\mathbf{A}, \mathbf{x})$ at its maximum *a posteriori* (MAP) estimate. In this case, the gradient simplifies to

$$\Delta \mathbf{A} \propto \boldsymbol{\sigma}_\epsilon^{-1} \boldsymbol{\epsilon} \hat{\mathbf{s}}^T \tag{2}$$

where $\hat{\mathbf{s}}$ is the MAP estimate of $p(\mathbf{s}|\mathbf{A}, \mathbf{x})$ and $\boldsymbol{\epsilon} = \mathbf{x} - \mathbf{A}\hat{\mathbf{s}}$ is the reconstruction error.

B. Shift-Invariant Model

In the standard sparse coding formulation as introduced above, the observations $\mathbf{x}$ are vectors. Many signals of interest in engineering, however, are time–series. In order to deal with such time–series, it is customary to partition the sequence into smaller blocks. These blocks can then be used as the observations $\mathbf{x}$ in the sparse coding model. However, one motivation for the use of the sparse coding model is to represent the observations as a linear combination of salient features. In time–series such as audio, it is not generally known *a priori* at which time-locations features occur. The features present in a particular observation block are then randomly shifted with respect to the beginning of the block. In order to model this uncertainty, the standard sparse coding model has to include several copies of each feature at all possible time-locations.

This structure can be learned from the observations themselves, which requires that the model includes enough free parameters so that the features can be learned at different locations. It is, however, of advantage to keep the number of free parameters low, which can be done by explicitly enforcing the shift-invariant structure in the dictionary as suggested in [6]–[10], and [13]. To state the model used in these references, we introduce the following notation.

From now on, we differentiate between the structured matrix $\hat{\mathbf{A}}$ in which all features occur at all possible shifted positions and the general unstructured matrix $\mathbf{A}$ used in the standard sparse coding model. The index k labels a particular feature, while the index l denotes the corresponding shift relative to the beginning of the data-block analyzed. $\mathcal{K}$ and $\mathcal{L}$ are the sets of indices of all features and shifts, respectively, while we denote the length of the features $\mathbf{a}_k$ as L. From now on, we let l be zero to denote no shift, i.e., $\mathbf{a}_{k0} = [a_{k1}, a_{k2}, \cdots, a_{kL}, 0, \ldots, 0]^T$ while, for example, $\mathbf{a}_{k-4} = [a_{k5}, a_{k6}, \cdots, a_{kL}, 0, \cdots, 0]^T$ and so forth. Note that for all $p - l \notin [0, L]$, the elements of $\mathbf{a}_{kl}$ are set to zero and that for $l < 0$ and $l > M - L$ the features $\mathbf{a}_k$ have to be truncated. We use a_{kp} to denote the pth component of a feature, which should not be confused with the notation $\mathbf{a}_{kl}$ that refers to a shifted feature. With this notation, we can write $\hat{\mathbf{A}} = [\mathbf{a}_{1,-L}, \mathbf{a}_{1,-L+1}, \ldots, \mathbf{a}_{k,M-1}, \mathbf{a}_{k+1,-L}, \ldots, \mathbf{a}_{K,M-1}]$.

$\hat{\mathbf{A}}$ is shown graphically for $M = 4, N = 12, L = 3, K = 2$ as follows:

$$\begin{bmatrix} \star_3 & \star_2 & \star_1 & 0 & 0 & 0 & \circ_3 & \circ_2 & \circ_1 & 0 & 0 & 0 \\ 0 & \star_3 & \star_2 & \star_1 & 0 & 0 & 0 & \circ_3 & \circ_2 & \circ_1 & 0 & 0 \\ 0 & 0 & \star_3 & \star_2 & \star_1 & 0 & 0 & 0 & \circ_3 & \circ_2 & \circ_1 & 0 \\ 0 & 0 & 0 & \star_3 & \star_2 & \star_1 & 0 & 0 & 0 & \circ_3 & \circ_2 & \circ_1 \end{bmatrix}.$$

Here, the two features are shown as stars $\star$ and circles $\circ$, respectively, with the subscripts labeling the sample.

If we use s_{kl} as the coefficient multiplying feature $\mathbf{a}_{kl}$, then the data model can be written as

$$\mathbf{x} = \sum_{k \in \mathcal{K}, l \in \mathcal{L}} \mathbf{a}_{kl} s_{kl} + \boldsymbol{\epsilon} = \hat{\mathbf{A}}\mathbf{s} + \boldsymbol{\epsilon}.$$

Note that this model is now a mixture of convolutions.

Another possible approach to the analysis of time–series would be to use a phase-blind spectral model. The power spectrum of a time–series is less affected by the exact positions of the block locations and has, therefore, been proposed for feature extraction [14], [15]. A detailed comparison between phase-blind spectral methods and the shift-invariant time-domain approach for music analysis can be found in [16], where the differences and similarities in the representations found with these two approaches are studied.

C. Learning in the Shift-Invariant Model

In the shift-invariant sparse coding model, the elements of the features $\mathbf{a}_k$ are repeated along the diagonals of the matrix $\hat{\mathbf{A}}$. When updating the values of $\hat{\mathbf{A}}$, this structure has to be taken into account, which is achieved by calculating the gradient of $\log p(\mathbf{x}|\hat{\mathbf{A}}, \mathbf{s})$ with respect to the pth component of the feature $\mathbf{a}_k$. As proposed in [7]–[10], the gradient learning rule (1) then becomes

$$\Delta a_{kp} = \sigma_\epsilon^{-1} \left\langle \sum_m \epsilon_m s_{k,p-m} \right\rangle_{p(\mathbf{s}|\hat{\mathbf{A}},\mathbf{x})}. \qquad (3)$$

Here, $\epsilon_m = x_m - \sum_{k \in \mathcal{K}, l \in \mathcal{L}} a_{km+l} s_{kl}$. Note that the term in the expectation is now a convolution.

Again, the expectation in (3) cannot be evaluated analytically. Making the same approximations that led to (2), the following update is found [7], [8]:

$$\Delta a_{kp} = \sigma_\epsilon^{-1} \sum_m \epsilon_m \tilde{s}_{k,p-m}. \qquad (4)$$

The convolution in (4) suggests the use of all coefficients $\mathbf{s}$, i.e., $\tilde{s}_{k,p-m} = \hat{s}_{k,p-m}$. By using the delta approximation of the posterior $p(\mathbf{s}|\hat{\mathbf{A}}.\mathbf{x})$, information about the distribution is, however, lost. This is especially critical for those feature shifts for which only part of the feature contributes to the current observation $\mathbf{x}$ (in the example above, these are columns one, two, five to eight, eleven, and twelve). For example, at the extreme shift positions, where a feature only overlaps with the observation block by one sample (i.e., columns one, six, seven and twelve in the above example), there is no information in the observation block to guide the selection of a specific feature. Any error in modeling the first and last sample in the observation block can, therefore, be reduced to exactly zero by selecting any feature at such an extreme shift with an appropriate coefficient value. This uncertainty would be reflected in the full posterior $p(\mathbf{s}|\hat{\mathbf{A}}, \mathbf{x})$ by an increased variance for the coefficients associated with these features. This information is not available in the delta approximation in (4) and, as suggested in [7], only those coefficients are used in the feature update for which the entire feature contributes to the observation. We, therefore, use

$$\tilde{s}_{k,p-m} = \begin{cases} \hat{s}_{k,p-m}, & \text{if } 0 \leq p - m \leq M - L \\ 0, & \text{otherwise.} \end{cases}$$

Note that ϵ_m is still calculated using all $\hat{s}_{kl}$. In the example above, we would, therefore, only use the coefficients associated with the third, fourth, ninth, and tenths columns. This does not bias the estimate of the features if the data blocks $\mathbf{x}$ are selected at random locations during learning.

D. Finding the MAP Estimate

In order to calculate the estimate of the gradient given in (4), the maximum of the posterior $p(\mathbf{s}|\hat{\mathbf{A}}, \mathbf{x})$ has to be found. Different optimization methods have been proposed for a range of prior distributions. If the prior $p(\mathbf{s})$ is assumed to be Laplacian, this optimization can be achieved using convex optimization routines such as linear programming as suggested in [3] and [4]. For more general sparse prior distributions, the posterior is not guaranteed to be unimodal. For these distributions, a gradient descent procedure can be applied. Another approach, which is the method used in the experiments reported below, is to use the expectation maximization (EM) algorithm proposed by Figueiredo *et al.* in [17], which for certain prior formulations is equivalent to the FOCUSS algorithm proposed by Rao in [18] and Kreutz-Delgado *et al.* in [19].

III. Approximate Inference Using A Subset Selection Step

Many engineering problems of interest suffer from high dimensionality. In the problems studied here, the length of the expected features can often be of the order of a few thousand and the number of features often in the hundreds. In the shift-invariant model this leads to a matrix $\hat{\mathbf{A}}$ of substantial size, which means that the calculation of the maximum of the posterior $p(\mathbf{s}|\hat{\mathbf{A}}, \mathbf{x})$ becomes prohibitively costly. This forbids a direct implementation of the above algorithms. Therefore, we propose the use of a subset selection step that offers a fast way to select a small subset of features depending on their correlation with the observation. After this selection, the optimization routines mentioned in the previous section can be used by ignoring features not contained in the subset. With this approach, results can be obtained even for problems of very high dimension.

Most of the coefficients s are zero with high probability and, therefore, most columns of $\hat{\mathbf{A}}$ do not contribute to any one observation. In order to speed up the optimization required to find the maximum of $p(\mathbf{s}|\hat{\mathbf{A}}, \mathbf{x})$, we propose to exclude a large set of the columns of $\hat{\mathbf{A}}$ from the optimization. Information about which features to keep and which to exclude has to be taken from a particular observation $\mathbf{x}$. It is further required that this selection process can be performed efficiently. This can be done by calculating the correlation between the observation $\mathbf{x}$ and all columns of $\hat{\mathbf{A}}$. Due to the structure in the matrix $\hat{\mathbf{A}}$, this correlation can be evaluated efficiently using fast convolution. Based on this correlation, it is possible to only select those features, for which this correlation is high. However, an additional constraint has to be imposed. As smooth features shifted only slightly are similar to themselves, the same feature would be selected several times at adjacent locations. This can be avoided by constraining the selected subset to only include shifted versions of the same features if these are shifted by more than a certain distance, i.e., by selecting $\mathbf{a}_{kl}$ and $\mathbf{a}_{k\tilde{l}}$ only if $((|l - \tilde{l}|)/L) > Q$ for some $Q < 1$.

The iterative selection procedure then selects the function and shift with the highest correlation as

$$\{k_i, l_i\} = \arg \max_{\{k,l\} \in \mathcal{K}_i \times \mathcal{L}_i} \langle \mathbf{a}_{kl}, \mathbf{x} \rangle$$

where the product space of indices $\mathcal{K}_i \times \mathcal{L}_i$ is defined iteratively by removing subsets from the set of all features and shifts as

$$\mathcal{K}_i \times \mathcal{L}_i = \mathcal{K} \times \mathcal{L} \backslash \bigcup_{\tilde{i} < i} k_{\tilde{i}} \times [l_{\tilde{i}} - QL; l_{\tilde{i}} + QL].$$

IV. Music Analysis

Previous work on shift-invariant sparse coding has focused on the problem-domain of image [10] and video [7]–[9] analysis.

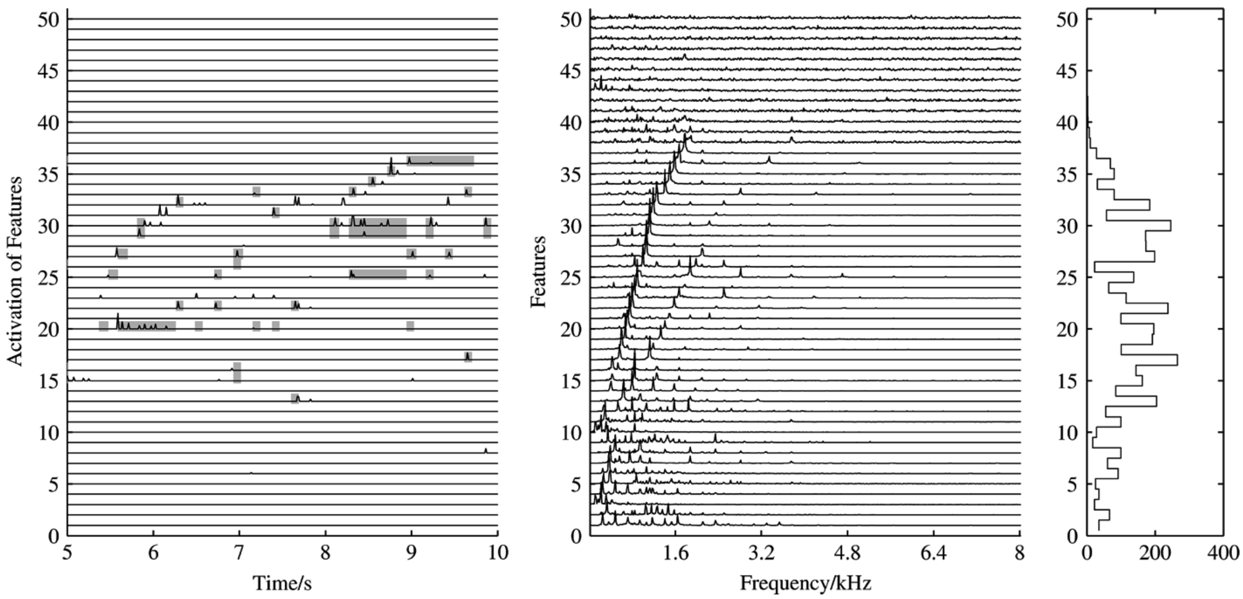

Fig. 1. Extract of the rectified activation pattern of the features represented by spikes, and notes in the original score represented with grey blocks (left), magnitude-spectrum of features (middle), and their number of occurrence in the decomposition (right).

In the following section, we demonstrate two possible applications of the method for polyphonic music analysis: piano note extraction and single-channel blind source separation.

A. Emergence of Musical Structures

In this section, we study the ability of the shift-invariant sparse coding method to discover note- and score-like structures in music signals. To this end, we use a polyphonic piano recording as this signal has a clear structure, and we assume that such a signal at least approximately follows the generative model used here. For such a signal, we would hope that the features would converge to note-like structures. We use a recording of Beethoven's Sonata for Piano no. 12, in A flat, Scherzo (Allegro molto). The original stereo recording was summed to mono and resampled at 8000 Hz. The number of possible features was set to 50, a feature length of 1024 samples was chosen, and the maximally allowed amount of overlap Q of one feature with a shifted version of itself was set to 50%.

The middle panel of Fig. 1 shows the magnitude-spectrum of the 50 learned features after 100 000 iterations. (It is worth noting that we here only show the magnitude-spectrum; the algorithm, however, extracted the features in the time-domain and, therefore, also learned phase information.) The features have been ordered by their approximated fundamental frequencies. Features 38–50 could not be assigned to a certain frequency as they had no clear peaks in their spectrum. Four of the features with clear spectral peaks did contain more than one harmonic series of peaks, i.e., they represented chord-like structures.

A small number of features had similar fundamental frequencies, however, features with similar fundamental frequencies differed in their harmonic structure. Those features with a well defined fundamental frequency were found to correspond to notes of the western equally tempered 12 tone scale in a range from C#2 to A5.

As most of the features $\mathbf{a}_k$ can be assigned to individual notes, the coefficients $\mathbf{s}$ contain information about the occurrence of the notes in the piano recording. This can be seen in the left panel of Fig. 1, where we show an extract of the rectified coefficients $\mathbf{s}$ associated with each of the features. In grey, we show the position and length of notes with the same pitch as they occur in the original score of the sonata. It can be seen that many of the occurrences of the features correspond to notes in the score. In the left panel of Fig. 1, we only show the notes of the original score for which a feature has been found. Some of the notes in the performance, however, did not have associated features and are, therefore, omitted. It is also clear that some of the notes in the score have no associated nonzero coefficients, and that some nonzero coefficients do not correspond to notes in the score. Some of these errors seem to be due to a feature modeling a different note to the one assigned to it here. This can be seen in the left panel of Fig. 1, where nonzero coefficients in the activation of feature 23 correspond to notes that in the assignment here should have been modeled by feature 20.

It was noted that some features emerged with equal fundamental frequency but with different harmonic structure. An example for this are features 29 and 30 in Fig. 1. In the left panel, it can be seen that these different features model different parts of a note, and it was found that the note onset was often modeled by a feature with higher high-frequency content, while the latter part of the note was often modeled with a feature with less high-frequency content.

In the right panel of Fig. 1, the number of occurrences of each feature in the decomposition is given. It is evident that a high number of features do not occur at all (seven features), and that other features only occur a few times (12 features occur less then ten times each) in the entire training signal. The features that did not occur in this particular decomposition are those features shown on the top in Fig. 1. It can, therefore, be assumed that these features have not been updated during learning and, therefore, cannot represent salient features of the signal.

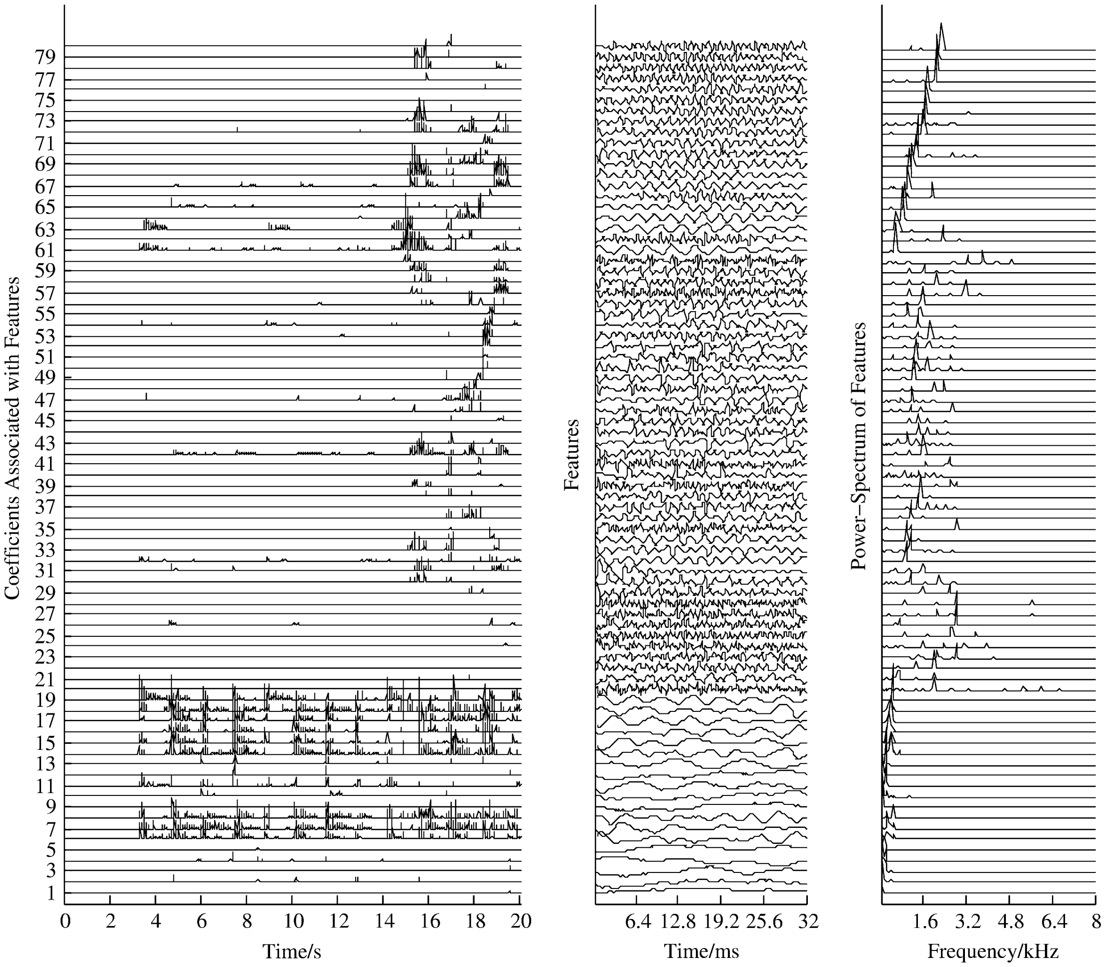

Fig. 2. Decomposition coefficients for the first 20 s of the piece (left), the associated time-domain features (center), and their power-spectrum (right).

B. Single-Channel Blind Source Separation

The separation of several source signals from a single observation requires the weighted assignment of time-frequency points to each source. If different sources overlap in frequency, linear transforms such as the Fourier transform cannot be used directly for this assignment. Sparse coding methods, however, offer such an assignment and can be used to learn source models with overlapping frequency support.

Previous approaches to single-channel blind source separation reported in the literature either rely on prior knowledge of a source model for each source to be recovered (see, for example, [20] and [21]) or treat the extracted features as individual sources (see, for example, [14] and [15]). The models in [14], [15], and [20] are further based on phase-blind spectral models that recover the sources by Wiener filtering methods.

In this section, we investigate the performance of the shift-invariant sparse coding algorithm for single-channel blind source separation in the case where it cannot be assumed in general that individual notes have similar waveforms each time they occur. We nevertheless show that for more general musical signals, features can be extracted that can be assigned to individual sources in the mixture. This classification then leads to a reconstruction of a signal using only those features corresponding to a single source.

For this experiment we recorded two separate signals; a vocal and a guitar track, which were mixed linearly and resampled to 8 kHz. It is important to stress that these signals were musically related, i.e., both guitar and singing where performing the same musical piece in harmony and with the same tempo so that both sources had much structure in common. We used this single-channel mixture as a training sequence for the algorithm. We learned 500 features of length 256 samples in a similar fashion to the experiments reported above. Of the 500 features, 126 had converged after 500 000 iterations, while the remaining features had not been updated substantially. In this experiment, all of the converged features had a clear harmonic structure. This can be seen in Fig. 2, where we show an extract of the coefficients $\mathbf{s}$ (left) associated with the learned features shown in the time-domain (middle) and the spectral domain (right). Here, we only show those features which could be clearly associated with a certain source using prior information (see below).

1) Oracle Clustering: In order to analyze the possible performance of the shift-invariant sparse coding method for blind source separation, we first perform separation of the sources by assigning the learned features to each source based on knowledge of the actual sources themselves, i.e., we use a nonblind (oracle) method.

The oracle assignment of features to sources was done depending on the energy each feature contributed to the representation of the individual sources, which was determined as

$$p_{k,\mathrm{vox}} = \frac{\|s_{k,\mathrm{vox}}\|}{\|s_{k,\mathrm{vox}}\| + \|s_{k,\mathrm{guitar}}\|}$$

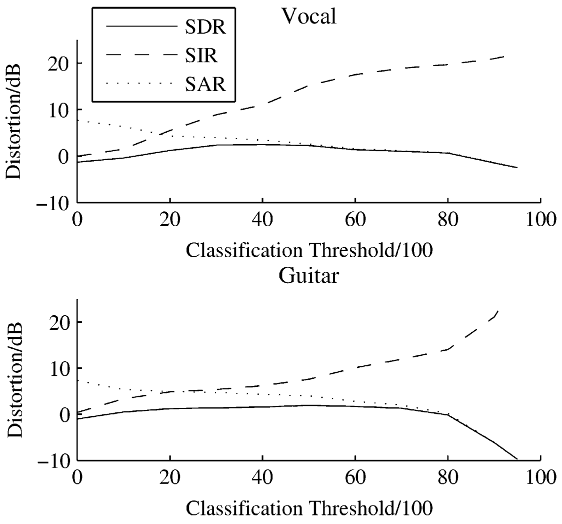

Fig. 3. Distortion (in decibels) for the separated sources. Vocal (top) and Guitar (bottom) and their associated distortions; SDR (solid), SIR (dashed), and SAR (dotted).

with $s_{k,\mathrm{vox}}(s_{k,\mathrm{guitar}})$ denoting the coefficients associated with feature k when analyzing the original vocal (guitar) signal. Different clusters could then be built by assigning features to a source whenever $p_k > P$ for some P. The results below are given for different values of $P < 1$. Note that $P = 0$ corresponds to the case in which all features are assigned to both sources, $P = 0.5$ corresponds to the case where each feature is assigned to a single source, and $P > 0.5$ corresponds to the case where some features are not assigned to any of the sources. For $P = 0.9$, we could assign 80 of the 129 features to a single source. These are the 80 features shown with their coefficients in Fig. 2.

After this clustering, we used the coefficients $\mathbf{s}$ from the decomposition of the mixture to reconstruct the sources using only those features assigned to each individual source. The performance of this separation was then measured using the method proposed in [22]. This gives us a measure of the signal to interference ratio (SIR), i.e., the ratio of the true source to the interference of the other sources in the estimated source as well as the signal to artefact ration (SAR), i.e., a measure of the artefacts introduced by the method. We can also calculate the overall signal to distortion ratio (SDR). For further details, the reader may refer to [22].

The top panel of Fig. 3 shows the SDR (solid), the SIR (dashed), and the SAR (dotted) results for the vocal reconstruction, while the lower panel gives the results for the guitar reconstruction. The SIR increases when fewer features are assigned to a source, while the overall SDR peaks at around 40% (vocal) and 50% (guitar) but is generally quite insensitive to the threshold. It is also clear that as fewer features are used in the reconstruction, the SAR decreases as more artefacts are introduced. The SIR (at $P = 0.9$) for the vocal reconstruction was 21 dB while the SIR for the guitar reconstruction at this value was also 21 dB. This means that the guitar track was suppressed by 21 dB in the vocal reconstruction. However, this reduction in interference between the sources leads to the introduction of artefacts. For the SIR levels reported above, the signal to artefact ratios were −1.4 and −6.1 dB, respectively. It can also be seen that even the reconstruction of the signal with all features is not artefact-free, and the highest SAR is 7 dB for this example.

2) Unsupervised Clustering: In real situations, the information used for clustering in the previous subsection is not available, and other methods for assigning features to sources are required. In previous methods (e.g., [20]), the features and models of the sources were learned from training sequences. However, different recordings of even the exact same instrument might change the recorded waveforms if the microphone position is changed or the recording done in another acoustic environment so that such a prior assignment is not feasible. Instead, it is required to cluster the features based only on the information available from the single mixture that was used in the feature learning procedure.

To facilitate clustering, we exploit higher level dependencies not modeled in the shift-invariant sparse coding model. In particular, we exploit the residual dependencies found in the coefficients $\mathbf{s}$ as well as dependencies between the features $\mathbf{a}_k$.

The coefficients $\mathbf{s}$ have been modeled as independent and identically distributed variables. However, for real sources, observations are not independent from previous observations, and the coefficients $\mathbf{s}$ are not independent over time.

In order to exploit temporal information in the coefficients $\mathbf{s}$ that has been ignored by the shift-invariant sparse coding algorithm, we estimate the probability of occurrence of a feature during a short time interval

$$p_t(\tilde{k}, i) = p(l \in [l_i, l_{i+1} - 1] : s_{kl} \neq 0, k = \tilde{k})$$

by

$$p_t(\tilde{k}, i) \approx \frac{1}{\sum_i \sum_{l \in [l_i, l_{i+1}-1]} |s_{\tilde{k}l}|} \sum_{l \in [l_i, l_{i+1}-1]} |s_{\tilde{k}l}|.$$

This histogram estimation does not only count the occurrence of the features but also takes their strength into account. This can be justified by assuming that a larger coefficient s is a sum of smaller "quantum" coefficients. This feature is based on the activation patterns of the coefficients s which are assumed to be similar for features $\mathbf{a}_k$ associated with a single source. Other features, such as a histogram estimate of the distribution of the coefficients s associated with each feature $\mathbf{a}_k$ or features based on the autocorrelation of or the cross-correlation between the coefficients s associated with each feature $\mathbf{a}_k$, were found not to work well for unsupervised clustering.

Individual instruments often have fixed physical characteristics that shape the spectrum of the produced sounds in a characteristic manner. The features $\mathbf{a}_k$ associated with the same source can, therefore, be assumed to have a similar spectral envelope. This similarity can be measured based on a spectral feature calculated by smoothing the power-spectrum of the features $\mathbf{a}_k$. This is done here by calculating the energy in the spectrum in several frequency bands.

In [5], the statistics of natural sounds have been shown to lead to efficient codes that have a wider frequency support at high frequencies. It was further argued in [5] that for speech, music, and some natural sounds the average power spectrum is approximately $1/f$ so that in order for each frequency band to have equal average power, the width of the frequency bands has to increase linearly with frequency. This is reflected in the frequency-discrimination found in the human auditory system, which is known to roughly follow a logarithmic scale. Therefore, we use a logarithmic frequency-domain partitioning of each feature $\mathbf{a}_k$ and calculate the feature as

$$p_f(\tilde{k}, i) \propto \sum_{l\in[2^i,2^{i+1}-1]} |\tilde{\mathbf{a}}_{\tilde{k}}(l)|^2$$

where $\tilde{\mathbf{a}}_{\tilde{k}}$ is the Fourier transform of feature $\mathbf{a}_{\tilde{k}}$. A linear partitioning is possible, however, for the experiments reported here, the results obtained were slightly worse to those obtained with the logarithmic partitioning.

Clustering of the features $\mathbf{a}_k$ can be performed using standard clustering algorithms. Here, we use the standard K-means algorithm with the symmetric Kullback–Leiber metric

$$KL(p(k,i), p(\tilde{k},i)) = 0.5 \sum_i p(k,i) \log \frac{p(k,i)}{p(\tilde{k},i)} + 0.5 \sum_i p(\tilde{k},i) \log \frac{p(\tilde{k},i)}{p(k,i)}$$

where $p(k,i)$ and $p(\tilde{k},i)$ are the two features to be compared.

In addition to the features p_t and p_f, we can also use a combination of these two features for clustering. The results obtained with these different features is shown in Table I. It is evident that for the example studied here, the feature p_t outperforms the feature p_f; a combination of both features, however, offers the best overall performance.

To show the tradeoff between the SIR and the SAR, it is again possible to assign a feature to more than one source or to assign some features to no source at all. This can be done by introducing a margin (positive, to assign some features to no sources and negative, to assign some features to more than one source). The SIR, SAR, and SDR values are given in Fig. 4 for different margins. Here, we show the results for clustering based on the combined features. The values obtained with a margin of zero are those shown in the right column of Table I. Again, the SDR is quite insensitive to the used margin; however, the change in SIR and the inverse change in SAR are less pronounced.

TABLE I
COMPARISON BETWEEN THE FEATURES FOR CLUSTERING

	p_f	p_t	$[p_t, p_f]$	Oracle P=0.5
SIR vocal	11.5	12.6	11.8	15.2
SIR guitar	4.7	9	9.9	7.6
SAR vocal	-0.2	3	3.2	2.6
SAR guitar	3.7	3.3	3	4.0
SDR vocal	-0.8	2.3	2.4	2.3
SDR guitar	0.4	1.8	1.8	1.9

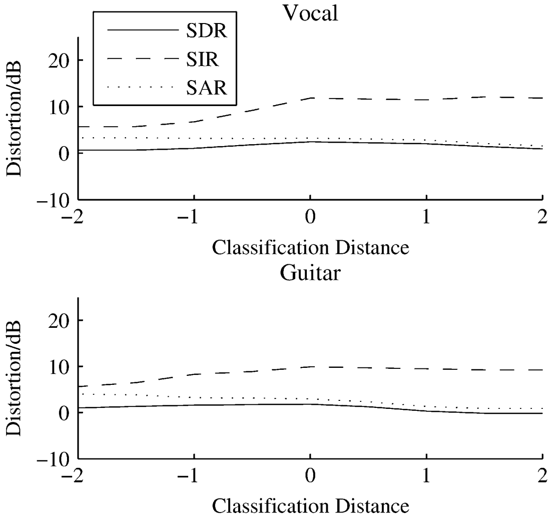

Fig. 4. Distortion (in decibels) for the blindly separated sources. Vocal (top) and Guitar (bottom) and their associated distortions; SDR (solid), SIR (dashed), and SAR (dotted).

V. Discussion and Conclusion

Shift-invariant sparse coding methods are able to extract salient features from time–series data. The abstract representations found with these methods resemble a series of spike trains, which suggests similarities of the shift-invariant sparse coding model to coding of perceptual signals in mammalian neural circuits. Such a model was proposed by Rieke *et al.* in [6] to reconstruct perceptual stimuli from recordings of neural spike trains. Lewicki and Sejnowski [13] also proposed the generative model used in shift-invariant sparse coding in order to find sparse codes of time-series for a given set of features $\mathbf{a}_k$. The shift-invariant sparse coding model can be seen as an extension of these ideas and learns both $\mathbf{a}_k$ and $\mathbf{s}$.

The shift-invariant sparse coding model can extract note- and score-like features from a polyphonic piano recording and is able to separate the sources from a mixture of human singing voice and guitar. The human singing voice does not fit the shift-invariant sparse coding model as well as the piano signal. It could nevertheless be shown that a decomposition of such a signal could be used to separate the singing voice from the accompanying guitar. These results suggest that a shift-invariant decomposition can capture and model certain aspects of individual sources when trained on a single mixture; however, not all features originate from a single source.

The proposed unsupervised clustering method required for blind source separation is based on the assumptions that the short time average activation of a features $\mathbf{a}_k$ is similar for features associated to a single source, and that the spectral shapes of features $\mathbf{a}_k$ from the same source have a similar spectral envelope. For music and vocal signals, the reported experiments seem to justify these assumptions; however, a more detailed analysis remains to be undertaken.

The linear mixture model is not necessarily the most accurate model to describe events in musical mixtures. The restriction to note prototypes of fixed length is a severe constraint. However, this model is simple enough to allow a Bayesian treatment and shows how the assumptions of independence and sparsity can extract information about the causes underlying an observation. A more complex model similar to the one in [20] might be used to model derivations of a note from its prototype; however, such a model would come with a substantial additional computational burden.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for additional references and valuable comments which greatly helped to improve this article.

REFERENCES

[1] H. B. Barlow, "Possible principles underlying the transformations of sensory messages," in *Sensory Communication*, W. A. Rosenblith, Ed. Cambridge, MA: MIT Press, 1961, pp. 217–234.

[2] P. Földiák, "Forming sparse representations by local anti-Hebian learning," *Biol. Cybern.*, vol. 64, no. 2, pp. 165–170, 1990.

[3] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 13, pp. 607–609, Jun. 1996.

[4] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural Comput.*, no. 12, pp. 337–365, 2000.

[5] M. S. Lewicki, "Efficient coding of natural sounds," *Nature Neurosci.*, vol. 5, pp. 356–363, Apr. 2002.

[6] F. Rieke, D. Warland, R. R. de Ruyter van Steveninck, and W. Bialeck, *Spikes, Exploring the Neural Code*. Cambridge, MA: MIT Press, 1997.

[7] R. Bogacz, M. W. Brown, and C. Giraud-Carrier, "Emergence of movement-sensitive neurons' properties by learning a sparse code of natural moving images," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2001, pp. 838–844.

[8] B. A. Olshausen, "Sparse coding of time-varying natural images," in *Proc. Int. Conf. Independent Component Analysis and Blind Source Separation (ICA)*, 2000, pp. 603–608.

[9] P. Sallee and B. A. Olshausen, "Learning sparse multiscale image representations," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003, pp. 1327–1334.

[10] H. Wersing, J. Eggert, and E. Körner, "Sparse coding with invariance constraints," in *Proc. Int. Conf. Artificial Neural Networks (ICANN)*, 2003, pp. 385–392.

[11] M. S. Lewicki and B. A. Olshausen, "A probabilistic framework for the adaptation and comparison of image codes," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 16, no. 7, pp. 1587–1601, 1999.

[12] B. A. Olshausen and K. Millman, "Learning sparse codes with a mixture-of-Gaussians prior," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Muller, Eds. Cambridge, MA: MIT Press, 2000, pp. 841–847.

[13] M. S. Lewicki and T. J. Sejnowski, "Coding time-varying signals using sparse, shift-invariant representations," *Advances Neural Inf. Process. Syst.*, vol. 11, pp. 730–736, 1999.

[14] P. Smaragdis, "Non-negative matrix factor deconvolution: Extraction of multiple sound sources from monophonic inputs," in *Proc. Int. Conf. Independent Component Analysis and Blind Source Separation (ICA)*, 2004, pp. 494–499.

[15] T. Virtanen, "Separation of sound sources by convolutive sparse coding," in *Proc. ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, 2004. [Online] Available: http://journal.speech.cs.cmu.edu/SAPA2004/.

[16] M. Plumbley, S. Abdallah, T. Blumensath, and M. Davies, "Sparse representations of polyphonic music," *EURASIP Signal Processing J.*, to be published.

[17] M. A. T. Figueiredo, "Adaptive sparseness for supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1150–1159, Sep. 2003.

[18] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 187–200, Jan. 1999.

[19] 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 Comput.*, vol. 15, no. 2, pp. 349–396, 2003.

[20] E. Vincent and X. Rodet, "Underdetermined source separation with structured source priors," in *Proc. Int. Conf. Independent Component Analysis and Blind Source Separation (ICA)*, 2004, pp. 327–334.

[21] G. J. Jang and T. W. Lee, "A probabilistic approach to single channel source separation," *Advances in Neural Information Processing Systems*, pp. 1173–1180, 2002.

[22] R. Gribonval, L. Benaroya, E. Vincent, and C. Févotte, "Proposals for Performance Measurement in Source Separation," Institut de Recherche et Coordination Acoustique/Musique, Paris, France, Tech. Rep. 1501, 2003.

Thomas Blumensath (M'05) received the B.Sc. degree (Hons.) in music technology from Derby University, Derby, U.K., in 2002 and is currently pursuing the Ph.D. degree in electronic engineering from Queen Mary, University of London, London, U.K.

His current research interests include mathematical and Bayesian methods in signal processing, unsupervised learning, sparse signal representations, neural computation, and applications to music and audio analysis.

Mike Davies (M'00) received the B.A. degree (Hons.) in engineering from Cambridge University, Cambridge, U.K., in 1989 and the Ph.D. degree in nonlinear dynamics and signal processing from University College London (UCL), London, U.K., in 1993.

He is currently a Reader at Queen Mary, University of London, in the Digital Signal Processing Research Group which he cofounded with Prof. Mark Sandler in 2001. His research interests include nonlinear dynamics, audio processing, statistical signal processing, array signal processing, and information theory.

Dr. Davies was awarded a prestigious Royal Society Research Fellowship which he held at UCL and then Cambridge.