POLYPHONIC MUSIC TRANSCRIPTION BY NON-NEGATIVE SPARSE CODING OF POWER SPECTRA

Samer A. Abdallah and Mark D. Plumbley
Centre for Digital Music,
Queen Mary, University of London

ABSTRACT

We present a system for adaptive spectral basis decomposition that learns to identify independent spectral features given a sequence of short-term Fourier spectra. When applied to recordings of polyphonic piano music, the individual notes are identified as salient features, and hence each short-term spectrum is decomposed into a sum of note spectra; the resulting encoding can be used as a basis for polyphonic transcription. The system is based on a probabilistic model equivalent to a form of noisy independent component analysis (ICA) or sparse coding with non-negativity constraints. We introduce a novel modification to this model that recognises that a short-term Fourier spectrum can be thought of as a noisy realisation of the power spectral density of an underlying Gaussian process, where the noise is essentially *multiplicative* and non-Gaussian. Results are presented for an analysis of a live recording of polyphonic piano music.

1. INTRODUCTION

In this paper we describe a method of spectral basis decomposition that can be applied to polyphonic music transcription. The approach belongs to and combines elements of a family of related methods that includes independent component analysis (ICA) [8], sparse coding [6], non-negative matrix factorisation (NMF) [10], and non-negative variants of both ICA [14] and sparse coding [7]. In the context of polyphonic transcription, the overall methodology is to identify the extracted components with the spectral profiles of the different notes, and thus to achieve the decomposition of a given mixed spectrum into a sum of those belonging to the currently sounding notes. The fact that the basis is adaptive means that the spectral profile of each note is learned by training the system on examples of polyphonic music, not on isolated notes (as in, e.g., [15]).

Similar approaches have been described in [1, 2, 16]. The technical novelty in this paper is that the underlying probabilistic model specifically addresses issues to do with spectral estimation (and more generally, the estimation of variance) in a Bayesian context. Thus, quite apart from applications in polyphonic transcription and feature extraction, the model forms a theoretical basis for spectral estimation and denoising using an ICA model to provide a strong but adaptive prior, which essentially plays the role of a *schema* or statistical summary of past experience, enabling the system to produce low-variance spectral estimates from limited data.

We tested the system on a recordings of Bach's G-minor fugue (No.16) from Book I of the Well Tempered Clavier; some results from one of these experiments are presented in § 5. Before that, the following sections describe some of the theoretical aspects of adaptive basis decomposition, Bayesian estimation of variance, and the combined system for non-negative sparse coding of power spectra.

We adopt the following typographical conventions: vectors and matrices are written in boldface $\mathbf{x}, \mathbf{A}$; random variables and vectors are denoted by uppercase letters $X, \mathbf{Y}$, while their realisations are denoted by lowercase letters $x, \mathbf{y}$. Where these conventions clash, the intended meaning should be clear from the context. Angle brackets $\langle\cdot\rangle$ will denote the expectation or averaging operator.

2. ADAPTIVE BASIS DECOMPOSITION

Systems for adaptive basis decomposition generally assume a linear generative model of the form $\mathbf{x} = \mathbf{A}\mathbf{s}$, or, writing out the sum explicitly,

$$x_i = \sum_{j=1}^{m} \mathbf{a}_j s_j, \tag{1}$$

where $\mathbf{x} = (x_1, \ldots, x_n)$ denotes an n-component multivariate observation, the $\mathbf{a}_j$ (for $1 \leq j \leq m$) denote a *dictionary* of m 'atomic' features (which form the columns of the $n \times m$) dictionary matrix $\mathbf{A}$, and $\mathbf{s} = (s_1, \ldots, s_m)$ contains the weighting coefficients. The purpose of the system is to learn, from examples of $\mathbf{x}$, a dictionary matrix $\mathbf{A}$ which contains a suitable collection of atomic features, and thence to encode optimally any given $\mathbf{x}$ as an $\mathbf{s}$ such that $\mathbf{x} \approx \mathbf{A}\mathbf{s}$. The learning process can be driven by a number of desiderata for the dictionary matrix and the components of $\mathbf{s}$, some of which we outline below.

An assumption of statistical independence between the s_j, motivated by considerations of redundancy reduction

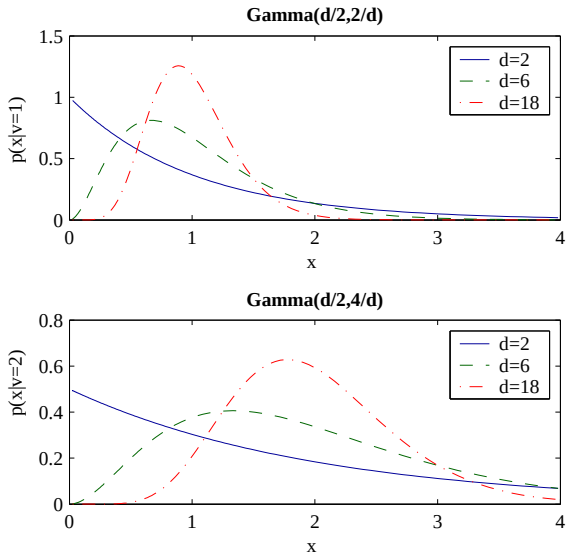

Figure 1. Some examples of Gamma distribution probability density functions. Note how shifting the mean of the distribution from 1 to 2 also doubles the standard deviation: this is because x is the product of v with a Gamma-distributed random variable, which therefore plays the role of a multiplicative noise.

and efficient representation [4], leads to ICA. Reducing the dependence between the components representing different notes should in principle reduce the need to examine several components in order to make one note onset/offset decision.

In sparse coding, we assume that most observations can be encoded with only 'a few' non-zero elements of $\mathbf{s}$, that is, only a few atomic features are required to account for typically observed patterns. This fits well with the notion that, in music, a relatively small fraction of the available notes will (usually) be sounding at any one time. Sparse coding can be facilitated in two distinct (but not mutually exclusive) ways: (a) by using a very large dictionary containing a wide variety of specialised features, or (b) by assuming a noisy generative model such that, after 'a few' dictionary elements have been activated, any remaining discrepancies can be treated as noise.

In some applications, the quantities involved are intrinsically non-negative; this is certainly the case for power spectra and variance estimates in general. Placing non-negativity constraints on the atomic features (the elements of $\mathbf{A}$), their weighting coefficients (the components of $\mathbf{s}$), or indeed both, can be enough to achieve meaningful feature detection in some applications without any additional assumptions, as demonstrated in [10].

Relationships between these different requirements have been investigated in, for example, [14, 7]. A recent and thorough treatment of sparse decomposition and dictionary learning can be found in [9].

3. BAYESIAN ESTIMATION OF VARIANCE

3.1. Univariate case

Consider a system in which we have d independent identically distributed (i.i.d.) Gaussian random variables (r.vs.) Z_k of zero mean and unknown variance v, that is, $Z_k \sim \mathcal{N}(0, v)$ for all $1 \leq k \leq d$. To estimate the variance v, one would compute the mean-square of a sample of the variables. It is a standard result (e.g., [5]) that, taken as a random variable itself, this estimate has a Gamma (or scaled Chi-squared) distribution:

$$X = \tfrac{1}{d}\sum_{k=1}^{d} Z_k^2 \sim \Gamma(\tfrac{d}{2}, \tfrac{2}{d}v) \sim \tfrac{1}{d}v\chi_d^2. \qquad (2)$$

Since $\langle X \rangle = v$, this estimator is unbiased, but noting that v appears only in the scale parameter of the Gamma distribution, we can see that, as a noisy estimate of v, it involves what is effectively a multiplicative, rather than an additive, noise model, the standard deviation of the error being proportional to v, the true variance. The probability density of the estimate given a particular variance is

$$p(x|v) = \frac{(\frac{dx}{2v})^{d/2} \exp -\frac{dx}{2v}}{x\Gamma(d/2)}, \qquad (3)$$

where Γ denotes the Gamma function. Some examples of Gamma densities are illustrated in fig. 1. When inferring v from observed values of x, we interpret the conditional density $p(x|v)$ as the likelihood of v given x; this can be combined with any prior expectations about v in the form of a prior density $p(v)$ to yield the posterior density

$$p(v|x) = \frac{p(x|v)p(v)}{p(x)}. \qquad (4)$$

The maximum a posteriori (MAP) estimate of the variance is therefore

$$\hat{v} = \arg\max_v \{\log p(x|v) + \log p(v)\}. \qquad (5)$$

Using the Gamma density (3), the log-likelihood term expands to

$$\log p(x|v) = \tfrac{d}{2}\log\tfrac{dx}{2v} - \tfrac{dx}{2v} - \log x - \log\Gamma(\tfrac{d}{2}). \qquad (6)$$

Considered as a function of v (illustrated in fig. 2), this expression plays the role of a statistically motivated error measure, or divergence, from v to x:

$$\log p(x|v) = -\mathcal{E}(v;x) + \{\text{Terms in } x \text{ and } d\}, \qquad (7)$$

$$\text{with} \quad \mathcal{E}(v;x) = \tfrac{d}{2}\left(\tfrac{x}{v} - 1 + \log\tfrac{v}{x}\right). \qquad (8)$$

The divergence $\mathcal{E}(v;x)$ reaches a minimum of zero when $v = x$, but unlike the quadratic error mesure $(v-x)^2$, it is strongly asymmetric, with a much higher 'cost' incurred when $v < x$ than when $v > x$. This expresses mathematically the intuition that samples from a Gaussian rv are quite likely to be much smaller than the standard deviation, but very unlikely to be much larger.

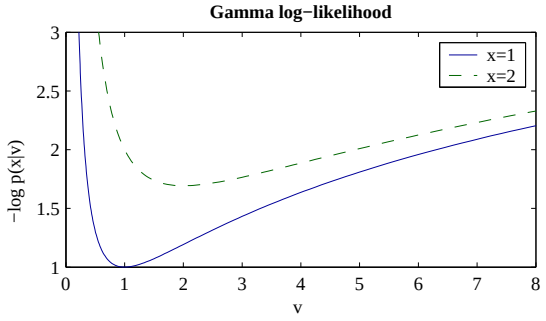

Figure 2. Gamma distribution (negative) log-likelihood function. The asymptotic behaviour is as $1/v$ as $v \to 0$ and as $\log v$ as $v \to \infty$.

3.2. Multivariate case

Assume now that $\mathbf{Y}$ is a multivariate Gaussian (i.e. a random vector) with components Y_k, $1 \leq k \leq N$. Diagonalisation of the covariance matrix $\left\langle \mathbf{Y}\mathbf{Y}^T \right\rangle = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ yields the orthogonal transformation $\mathbf{U}$ (whose columns are the eigenvectors $\mathbf{u}_i$), and the eigenvalue spectrum encoded in the diagonal matrix $\Lambda_{kl} = \delta_{kl}\lambda_k$. If there are degenerate eigenvalues (i.e. of the same value), then the corresponding eigenvectors, though indeterminate, will span some determinate subspace of $\mathbb{R}^N$ such that the distribution of $\mathbf{Y}$ projected into that subspace is isotropic (i.e. spherical or hyperspherical).

Now let us assume that the eigenvectors $\mathbf{u}_k$ and the degeneracy structure of the eigenvalue spectrum are known (i.e., the 'eigen-subspaces' are known), but the actual values of the λ_k are unknown, and are to be estimated from observations. We are now in a situation where we have several independent copies of the problem described in the preceding section. Specifically, if there are n distinct eigenvalues, then the subspaces can be defined by n sets of indices $\mathcal{I}_i$ such that the ith subspace is spanned by the basis $\{\mathbf{u}_k | k \in \mathcal{I}_i\}$. A maximum-likelihood estimate of the variance in that subspace, $v_i = \lambda_k \; \forall \; k \in \mathcal{I}_i$, can be computed as in (2):

$$\hat{v_i} = x_i = \frac{1}{|\mathcal{I}_i|} \sum_{k \in \mathcal{I}_i} (\mathbf{u}_k^T \mathbf{y})^2, \tag{9}$$

where $|\mathcal{I}_i|$, the dimensionality of the ith subspace, plays the role of d, the Chi-squared 'degrees-of-freedom' parameter in (2). Assuming the subspaces are statistically independent given the variances v_i, the rest of the derivation of § 3.1 can be extended to yield the multivariate divergence

$$\mathcal{E}(\mathbf{v}; \mathbf{x}) = \sum_{i=1}^{n} \frac{|\mathcal{I}_i|}{2} \left(\frac{x_i}{v_i} - 1 + \log \frac{v_i}{x_i} \right), \tag{10}$$

where $\mathbf{x}$ and $\mathbf{v}$ denote the arrays formed by the components x_i and v_i respectively.

3.3. Application to power spectra

One of the effects of a Fourier transform is to diagonalise the covariance of a stationary Gaussian process, the eigenvalue spectrum being in this case equivalent to the power spectral density (PSD) of the Gaussian process. The discrete short-term Fourier transform is an approximation to this (the windowing process makes it inexact) for time-varying Gaussian processes; this is how the periodogram method of spectral estimation works.

PSD estimation is a specific instance of a more general class of covariance estimation problem where the 'eigen-subspaces' happen to be known in advance: the diagonalising transformation is the Fourier transform, and the sinusoidal eigenvectors (except those encoding the DC component and possibly the Nyquist frequency) come in pairs of equal frequency but quadrature phase. These pairs of eigenvectors will have the same eigenvalue and will therefore span a number of 2-D subspaces, one for each discrete frequency. The problem of spectral estimation is equivalent to that of estimating the variance in each of these known subspaces. If no further assumptions are made about these variances (that is, if the prior $p(\mathbf{v})$ is flat and uninformative) then any estimated PSD will have a large standard-deviation proportional to the true PSD as illustrated by the $d = 2$ curves in fig. 1. Our system aims to improve these estimates by using a structured prior, but unlike those implicit in parametric methods such as autoregressive models (which essentially amount to smoothness constraints), we use an adaptive prior in the form of an explicit ICA model.

4. GENERATIVE MODEL

The prior $p(\mathbf{v})$ on the subspace variances is derived from the linear generative model used in adaptive basis decomposition (1): we assume that $\mathbf{v} = \mathbf{A}\mathbf{s}$, where the components of $\mathbf{s}$ are assumed to be non-negative, independent, and sparsely distributed. If $\mathbf{A}$ is square and non-singular, then we have $p(\mathbf{v}) = \det \mathbf{A}^{-1} \prod_{j=1}^{m} p(s_j)$, where $\mathbf{s} = \mathbf{A}^{-1}\mathbf{s}$ and $p(s_j)$ is the prior on the jth component. These priors are assumed to be single-sided ($s_j \geq 0$) and sharply peaked at zero, to express the notion that we expect components to be 'inactive' (close to zero) most of the time. In the case that $\mathbf{A}$ is not invertible, the situation is a little more complicated; we circumvent this by doing inference in the $\mathbf{s}$-domain rather than the $\mathbf{v}$-domain, that is we estimate $\mathbf{s}$ directly by considering the posterior $p(\mathbf{s}|\mathbf{x}, \mathbf{A})$, rather than $p(\mathbf{v}|\mathbf{x})$. The elements of $\mathbf{A}$, representing as they do a set of atomic power spectra, are also required to be non-negative. The complete probability model is

$$\begin{aligned} p(\mathbf{x}, \mathbf{s}|\mathbf{A}) &= p(\mathbf{x}|\mathbf{A}\mathbf{s}) p(\mathbf{s}) \\ &= \prod_{i=1}^{n} p(x_i|v_i) \prod_{j=1}^{m} p(s_j), \end{aligned} \tag{11}$$

where $\mathbf{v} = \mathbf{A}\mathbf{s}$ and $p(x_i|v_i)$ is defined as in (3). An important point is that this linear model, combined with

the multiplicative noise model that determines $p(\mathbf{x}|\mathbf{v})$, is physically accurate for the composition of power spectra arising from the superposition of phase-incoherent Gaussian processes, barring discretisation and windowing effects. It is not accurate for *magnitude* spectra or log-spectra. On the other hand, additive Gaussian noise models are not accurate in any of these cases.

4.1. Sparse decomposition

It is straightforward to extend the analysis of § 3.2 to obtain a MAP estimate of the components of $\mathbf{s}$ rather than those of $\mathbf{v}$:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \mathcal{E}(\mathbf{A}\mathbf{s}; \mathbf{x}) - \log p(\mathbf{s}), \tag{12}$$

where $p(\mathbf{s})$ is the factorial prior $p(\mathbf{s}) = \prod_{j=1}^{m} p(s_j)$. When the $p(s_j)$ have the appropriate form (see [9]), the $-\log p(\mathbf{s})$ terms plays the role of a 'diversity' cost which penalises non-sparse activity patterns. If we assume the $p(s_j)$ to be continuous and differentiable for $s_j \geq 0$, then local minima can be found by searching for zeros of the gradient of the objective function in (12). Using (10) this expands to a set of m conditions

$$\forall\ j, \quad \sum_{i=1}^{n} A_{ij} \frac{|\mathcal{I}_i|}{2} \left(\frac{v_i - x_i}{v_i^2} \right) + \varphi(s_j) = 0, \tag{13}$$

where $\varphi(s_j) \stackrel{\text{def}}{=} -\mathrm{d} \log p(s_j)/\mathrm{d}s_j$. The optimisation can be achieved by standard 2nd order gradient based methods with non-negativity constraints, but these tend to converge poorly, and for large systems such as we intend to deal with, each individual step is rather expensive computationally. Steepest descent is worse still, tending to become unstable as any component of $\mathbf{v}$ approaches zero.

We found that a modified form of Lee and Seung's non-negative optimisation algorithm [11] gives much better overall performance. Their algorithm minimises a different measure of divergence between $\mathbf{A}\mathbf{s}$ and $\mathbf{x}$, with no additional diversity cost. Our modification accommodates both the diversity cost (as in [7]) and the Gamma-likelihood based divergence (10). The iterative algorithm is as follows. (To simplify the notation, we will assume that $\forall\, i,\ |\mathcal{I}_i| = d$, i.e., that all the independent subspaces of the original Gaussian rv are of the same dimension d.) The s_j are initialised to some positive values, after which the following assignment is made at each iteration:

$$\forall\ j, \quad s_j \leftarrow s_j \frac{\sum_{i=1}^{n} A_{ij} x_i / v_i^2}{(2/d)\varphi(s_j) + \sum_{i=1}^{n} A_{ij} / v_i} \tag{14}$$

This is guaranteed to preserve the non-negativity of the s_j as long as the A_{ij} are non-negative and $\varphi(s_j) \geq 0$ for $s_j \geq 0$, though some care must be taken to trap any division-by-zero conditions which sometimes occur. States that satisfy (13) can easily be shown to be fixed points of (14), and although the convergence proofs given in [11, 7] do not apply, we have found that in practice, the algorithm converges very reliably. Note that the subspace dimensionality parameter d (i.e. the degrees-of-freedom in the Gamma-distributed noise model) reduces to controlling the relative weighting between the requirements of good spectral fit on one hand and sparsity on the other.

4.2. Learning the dictionary matrix

We adopt a maximum-likelihood approach to learning the dictionary matrix $\mathbf{A}$. Due to well known difficulties [9] with maximising the average log-likelihood $\langle \log p(\mathbf{x}|\mathbf{A}) \rangle$, (that is, treating the components of $\mathbf{s}$ as 'nuisance variables' to be integrated out) we instead aim to maximise that average *joint* log-likelihood $\langle \log p(\mathbf{x}, \mathbf{s}|\mathbf{A}) \rangle$. Let $\mathbf{x}_{1:T} \equiv (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ denote a sequence of T training examples, with $\mathbf{s}_{1:T}$ the corresponding sequence of currently estimated components obtained by one or more iterations of (14), and $\mathbf{v}_t = \mathbf{A}\mathbf{s}_t$, where $\mathbf{A}$ is the current estimated dictionary matrix. Then, the combined processes of inference and learning can be written as the joint optimisation

$$(\hat{\mathbf{A}}, \hat{\mathbf{s}}_{1:T}) = \arg \max_{(\mathbf{A}, \mathbf{s}_{1:T})} \sum_{t=1}^{T} \log p(\mathbf{x}_t, \mathbf{s}_t|\mathbf{A}), \tag{15}$$

where $p(\mathbf{x}, \mathbf{s}|\mathbf{A})$ is defined in (11). Both multiplicative and additive update algorithms were investigated. Additive updates were found to be adequate for small problems (e.g. $n = 3$), but unstable when applied to real power spectra ($n = 257$, $n = 513$). The following multiplicative update (modelled on those in [10]) was found to be effective (the sequence index t has been appended to the component indices i and j):

$$\forall\, i, j,\ A_{ij} \leftarrow A_{ij} \left(\frac{\sum_{t=1}^{T} s_{jt} x_{it} / v_{it}^2}{\sum_{t=1}^{T} s_{jt} / v_{it}} \right)^{\eta}, \tag{16}$$

$$\mathbf{A} \leftarrow \mathrm{normalise}_p\, \mathbf{A}, \tag{17}$$

where $\eta \leq 1$ is a step size parameter to enable an approximate form of online learning, and the operator $\mathrm{normalise}_p$ rescales to unit $p-$norm each column of $\mathbf{A}$ independently. For example, if $p = 1$, the column sums are normalised. The values of $\mathbf{s}$ (and hence $\mathbf{v} = \mathbf{A}\mathbf{s}$) inside the sums may be computed either by interleaving these dictionary updates with a few incremental iterations of (14), or by re-initialising the $\mathbf{s}_t$ and applying many (typically, around 100) iterations of (14) for each iteration of (17). Clearly, the latter alternative is much slower, but is the only option if online learning is required.

5. APPLICATION TO PIANO MUSIC

The system was tested on a recording of J. S. Bach's Fugue in G-minor No. 16. The input was a sequence of 1024 Fourier spectra computed from frames of 512 samples each, with a hop size of 256 samples, covering the first $9\frac{1}{2}$ bars of the piece. The only preprocessing was a spectral normalisation or flattening (that is, a rescaling of each row of the spectrogram) computed by fitting a generalised exponential distribution to the activities in each

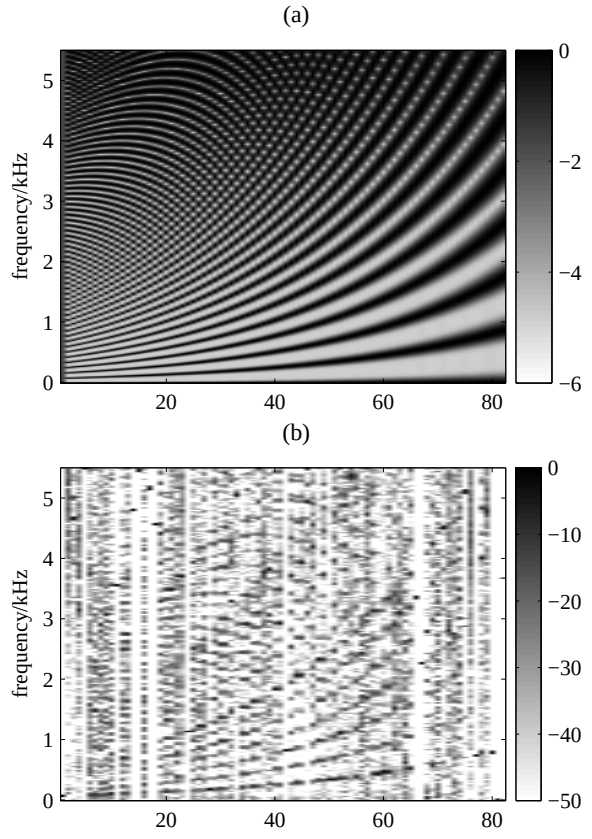

Figure 3. Dictionary matrix before (a) and after (b) training on an extract of piano music. The remnants of the original structured dictionary are still visible, but each pitched spectrum has become adapted to the actual spectrum of the piano notes in the piece. The colour scale is logarithmic and in dB.

spectral bin, as outlined in [2]. In the sparse coding system described therein, spectral flattening was used to improve the fit between the assumed additive white noise model and the data. The multiplicative noise system described in this paper does not require such a rescaling of the data, but the preprocessing was retained both as an aid to visualisation (so that, for exampe, the detail in the upper frequencies of fig. 4(a,c) is visible) and to enable future comparisons between the additive and multiplicative systems.

Experiments were performed both with random and structured initial dictionaries. The structured initial dictionary, illustrated in fig. 3(a), consisted of an ordered collection of roughly pitched spectra on a quarter-tone spacing. Each column of the dictionary matrix was a constructed as a sequence of broad harmonics increasing in width with frequency, to allow for inharmonicity or variations in intonation in the tones to be analysed. After training, both dictionaries converged to qualitatively similar solutions, but initialisation with the structured dictionary tends to result in a correctly ordered final dictionary. This ordering, however, is not essential to the system, and can be recovered after training.

Training was accomplished by alternate applications of the multiplicative update rules (14) and (17), with $p = 1$ in the column normalisation step; that is, each atomic spectrum was normalised to have unit total energy. In addition, a small constant offset was periodically added to all elements of $\mathbf{A}$ and $\mathbf{s}_{1:T}$ in order to nudge the system out of local minima caused by zero elements, which, of course, the multiplicative updates are unable to affect. The resulting dictionary is illustrated in fig. 3(b).

The next step in the process was the categorisation of the atomic spectra in the learned dictionary as either pitched or non-pitched, followed by an assignment of pitch to each of the pitched spectra. The 'pitchedness' categorisation was done by a visual inspection of the spectra in combination with a subjective auditory assessment of the sounds reconstructed from the atomic spectra by filtering white Gaussian noise, as described in [2]. We are currently investigating quantitative measures of 'pitchedness' so that this process can be automated.

Once pitches had been assigned to eached of the pitched spectra in the dictionary, we found that many pitches were represented by more than one dictionary element, which elements can therefore be arranged into groups by pitch. The different elements in a particular group represent different spectral realisations of the same pitch, which may occur during different instances of the same note or at different stages in the temporal evolution of a single note. For example, the fourth note (an F♯3) in the extract in fig. 4(b) can be seen to involve activity in two dictionary elements.

In order to obtain the pitch traces in fig. 4(d), the activities (i.e. the component values s_j) in each pitch group were summed. Specifically, if $\mathcal{P}_k$ is the set of components in the kth pitch class, then the activity of that pitch class is

$$\sigma_k = \sum_{j \in \mathcal{P}_k} s_j. \tag{18}$$

Since the dictionary matrix is column normalised using a 1-norm, each atomic spectrum has the same total energy, so the sum of the activities in each group has a direct interpretation as the total energy attributable to that note. These energies have a very wide dynamic range, so, for display purposes, we plot $\log(\sigma_k^2 + 1)$ for each pitch class in fig. 4(d).

The pitched dictionary elements corresponded to notes in a three octave range from E2 to G5. We have yet to implement the final stages of event detection and time quantisation, so an evaluation was done by visual comparison of fig. 4 with the original score. All the notes in the first $9\frac{1}{2}$ bars were correctly detected, except for a repeated G4 in bar 5, which is coalesced into the preceding G4 (circled in fig. 4(d), at time 10.5s). Given a sufficiently robust peak-picking algorithm, most of the errors would be false detections of repeated notes, (two of which are circled in fig. 4) though we cannot provide any quantitative results at this stage. The manual evaluation can be summarised as follows:

Notes in original extract	163
Correctly detected notes	162
False detections	2

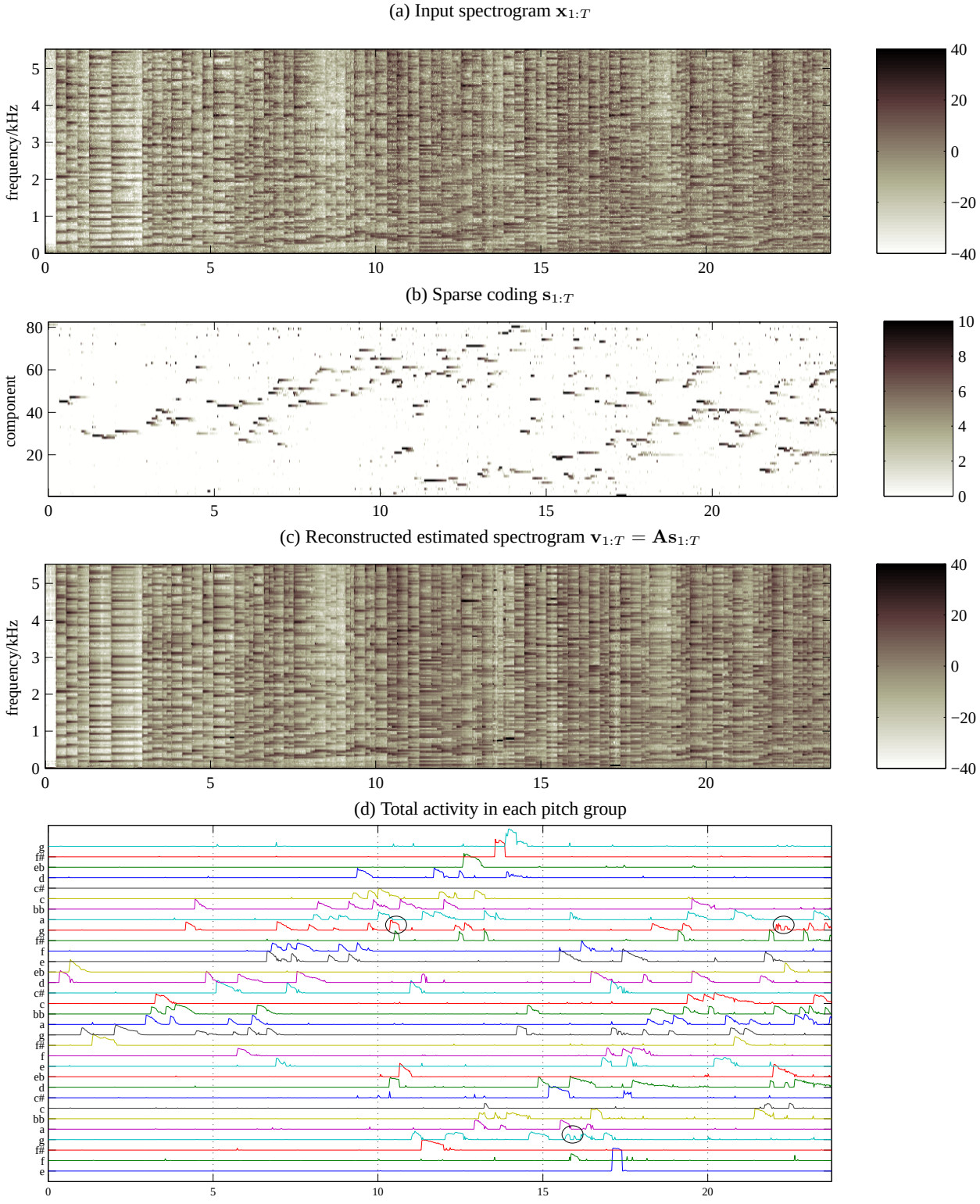

Figure 4. Analysis of the first $9\frac{1}{2}$ bars of Bach's Fugue in G-minor, BWV861: (a) input spectrogram; (b) sparse coding using the dictionary in fig. 3; (c) reconstruction, which can be thought of as a 'schematised' version of the input—note the denoising effect and the absence of beating partials. Finally (d) graphs the pitch-group activities as $\log(\sigma_k^2 + 1)$, where k ranges over the pitches and σ_k is defined in (18). Errors in transcription are circled (see text). In all figure, the x-axis indicates time in seconds.

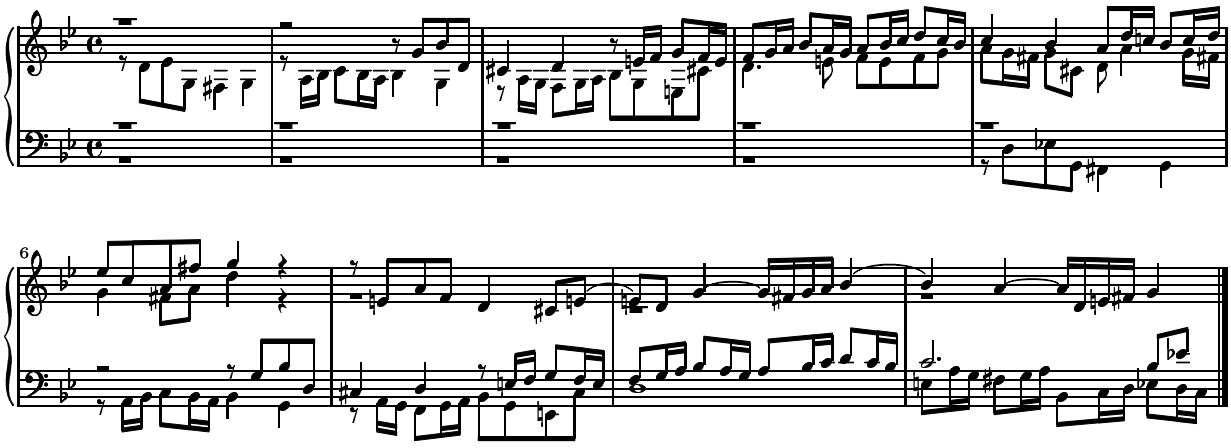

Figure 5. The first 10 bars of the Fugue in G-minor, BWV861.

6. RELATIONS WITH OTHER METHODS

The algorithm presented here is largely derived from Lee and Seung's multiplicative non-negative matrix factorisation (NMF) algorithms [10, 11], which are founded on two divergence measures, one of which is quadratic and can be interpreted in terms of an additive Gaussian noise model; the other, in the present notation, can be written as

$$\mathcal{E}_{LS}(\mathbf{v};\mathbf{x}) = \sum_{i=1}^{n} x_i \log \frac{x_i}{v_i} - x_i + v_i \qquad (19)$$

This divergence measure is related to the Kullback-Leibler divergence, and can be derived by interpreting $\mathbf{v}$ as a discrete probability distribution (over whatever domain is indexed by i) and $\mathbf{x}$ as a data distribution. The divergence measures the likelihood that that the data distribution was drawn from the underlying probability distribution specified by $\mathbf{v}$. Smaragdis [16] applies this form of NMF to polyphonic transcription and achieves results very similar to those presented in this paper. However, we would argue that power spectra are *not* probability distributions over frequency, and that the resulting system does not have a formal interpretation in terms of spectral estimation.

Hoyer [7] modified the quadratic-divergence version of Lee and Seung's method to include the effect of a sparse prior on the components $\mathbf{s}$, and applied the resulting system to image analysis. He used an additive update step (with post-hoc rectification to enforce non-negativity) for the adapting the dictionary matrix $\mathbf{A}$, rather than a multiplicative step. In the present work, additive updates were found to be rather unstable due to singularities in the divergence measure (10) as any component v_i approaches zero.

Abdallah and Plumbley [1, 2] applied sparse coding with additive Gaussian noise and no non-negativity constraints to the analysis of magnitude spectra. The algorithm was based on the overcomplete, noisy ICA methods of [13]. The system was found to be effective for transcribing polyphonic music rendered using a synthetic harpsichord sound, but less able to deal with the wide dynamic range and spectral variability of a real piano sound.

It is interesting to note some parallels between the present work and the polyphonic transcription system of Lepain [12]. His system was built around an additive decomposition of log-power spectra into a manually chosen basis of harmonic combs. This basis included several versions of each pitch with different spectral envelopes. The error measure used to drive the decomposition was an asymmetric one. If, using the current notation, we let $w_i = \log v_i$, and $z_i = \log x_i$, Lepain's error measure would be

$$\mathcal{E}_{L}(\mathbf{w};\mathbf{z}) = \sum_{i=1}^{n} (w_i - z_i), \quad w_i \geq z_i \; \forall \; i. \qquad (20)$$

For comparison, the log-likelihood $\log p(\mathbf{z};\mathbf{w})$ can be derived from the Gamma-distributed multiplicative noise model (3), yielding

$$-\log p(\mathbf{z}|\mathbf{w}) = \sum_{i=1}^{n} \tfrac{d_i}{2} \left\{\exp(z_i - w_i) + (w_i - z_i)\right\} + \{\text{Terms in } d_i\}, \qquad (21)$$

where d_i denotes the degrees-of-freedom for the ith component. The exponential term means the error measure rises steeply when $w_i < z_i$, but approximately linearly when $w_i > z_i$, and thus Lepain's error measure can be seen as a rough approximation to this, using a hard constraint instead of the exponential 'barrier' found in the probabilistically motivated measure.

7. SUMMARY AND CONCLUSIONS

A system for non-negative, sparse, linear decomposition of power spectra using a multiplicative noise model was presented and applied to the problem of polyphonic transcription from a live acoustic recording. The noise model was derived from a consideration of the estimation of the variance of a Gaussian random vector, of which spectral estimation is a special case, while the generative model for power spectra belongs to a class of ICA-based models, in which the power spectra are assumed to the result of

a linear superposition of independently weighted 'atomic' spectra chosen from a dictionary. This dictionary is in turn learned from, and adapted to, a given set of training data. These theoretical underpinnings mean that the system has a formal interpretation as a form of spectral estimation for time-varying Gaussian processes using a sparse factorial linear generative model as an adaptive prior over the power spectra.

The learned dictionary can be thought of as an environmentally determined 'schema', a statistical summary of past experiences with power spectra, which enables the system to make better inferences about newly-encountered spectra. When exposed to polyphonic music, this schema quickly adapts to the consistent presence of harmonically structured notes. The internal coding of spectra (i.e. the components of $\mathbf{s}$) therefore reflects the presence or absence of notes quite accurately, while the reconstructed spectra (the vectors $\mathbf{v} = \mathbf{A}\mathbf{s}$) are essentially a 'schematised' (cleaned up, denoised, and 'straightened out') version of the input ($\mathbf{x}$).

The encoding produced by the system, though not a finished transcription, should provide a good basis for one, once the final stages of (a) automatic grouping of dictionary elements into subspaces by pitch and (b) event detection on the per-pitch total energy traces, have been implemented. The manual evaluation ($\S\,5$) suggests that a transcription accuracy of 99% could be achievable given a sufficiently robust and adaptable 'peak-picking' algorithm; we refer the interested reader to [3] for an overview of our initial efforts in that direction.

8. ACKNOWLEDGMENTS

This work was supported by EPSRC grant GR/R54620/01 (Automatic polyphonic music transcription using multiple cause models and independent component analysis) and EU-FP6-IST-507142 project SIMAC (Semantic Interaction with Music Audio Contents).

9. REFERENCES

[1] Samer A. Abdallah. *Towards Music Perception by Redundancy Reduction and Unsupervised Learning in Probabilistic Models*. PhD thesis, Department of Electronic Engineering, King's College London, 2002.

[2] Samer A. Abdallah and Mark D. Plumbley. Unsupervised analysis of polyphonic music using sparse coding in a probabilistic framework. *IEEE Trans. on Neural Networks*, 2003. Submitted for review.

[3] Samer A. Abdallah and Mark D. Plumbley. Unsupervised onset detection: a probabilistic approach using ICA and a hidden Markov classifier. In *Cambridge Music Processing Colloquium*, Cambridge, UK, 2003.

[4] Horace B. Barlow. Sensory mechanisms, the reduction of redundancy, and intelligence. In *Proceedings of a Symposium on the Mechanisation of Thought Processes*, volume 2, pages 537–559, National Physical Laboratory, Teddington, 1959. Her Majesty's Stationery Office, London.

[5] William Feller. *An Introduction to Probability Theory and its Applications*, volume II. John Wiley and Sons, New York, 1971.

[6] David J. Field and Bruno A. Olshausen. Emergence of simple-cell receptive field properties by learing a sparse code for natural images. *Nature*, 381:607–609, 1996.

[7] Patrik O. Hoyer. Non-negative sparse coding. In *Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing)*, Martigny, Switzerland, 2002.

[8] Aapo Hyvärinen. Survey on Independent Component Analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[9] Kenneth Kreutz-Delgado, Joseph F. Murray, and Bhaskar D. Rao. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15:349–396, 2003.

[10] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[11] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 556–562, Cambridge, MA, 2001. MIT Press.

[12] Philippe Lepain. Polyphonic pitch extraction from musical signals. *Journal of New Music Research*, 28(4):296–309, 1999.

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

[14] Mark Plumbley. Algorithms for nonnegative independent component analysis. *IEEE Transactions on Neural Networks*, 14(3):534–543, 2003.

[15] L. Rossi, G. Girolami, and L. Leca. Identification of polyphonic piano signals. *Acustica*, 83(6):1077–1084, 1997.

[16] Paris Smaragdis. Non-negative matrix factorization for polyphonic music transcription. In *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 177–180, New Paltz, NY, 2003.