Shift Invariant Sparse Coding of Image and Music Data

Morten Mørup, Mikkel N. Schmidt and Lars Kai Hansen

DTU Informatics
Technical University of Denmark
Richard Petersens Plads bld. 321, 2800 Kgs. Lyngby, Denmark
{mm,mns,lkh}@imm.dtu.dk

Abstract. When analyzing multi-media data such as image and music it is useful to extract higher-level features that constitute prominent signatures of the data. We demonstrate how a 2D shift invariant sparse coding model is capable of extracting such higher level features forming so-called icon alphabets for the data. For image data the model is able to find high-level prominent features while for music the model is able to extract both the harmonic structure of instruments as well as indicate the scores they play. We further demonstrate that non-negativity constraints are useful since they favor part based representation. The success of the model relies in finding a good value for the degree of sparsity. For this, we propose an 'L-curve'-like argument and use the sparsity parameter that maximizes the curvature in the graph of the residual sum of squares plotted against the number of non-zero elements of the sparse code. Matlab implementation of the algorithm is available for download.

1 Introduction

Sparse coding and the closely related independent component analysis (ICA) are well established principles for feature extraction [23, 22, 14, 7, 15]. [23] argue that the brain might employ sparse coding since it allows for increased storage capacity in associative memories; it makes the structure in natural signals explicit; it represents complex data in a way that is easier to read out at subsequent level of processing; and it is energy efficient. Thus, sparseness is a natural constraint for unsupervised learning and sparse coding often results in parsimonious features.

Neurons in the inferotemporal cortex respond to moderately complex features, icon alphabets, which are invariant to the position of the visual stimulus [31]. Based on this observation, [13] formulated a model that estimates such shift invariant image features. The resulting features are complex patterns rather than the Gabor-like features often obtained by sparse coding or ICA decomposition [22, 15]. These shift invariant features can potentially constitute such icon alphabet.

For audio, it has been demonstrated that sparse over-complete linear representations solve hard acoustic signal processing problems [1]. These results suggest that auditory cortex employs sparse coding. Receptive fields in auditory cortex often have broad and complex time-frequency structure, and the auditory

system uses a highly over-complete representation. The features in the sparse over-complete representation are complex structures that form an "acoustic icon alphabet". Furthermore, infants can distinguish melodies regardless of pitch [32], and since a change of pitch relates to a shift on a logarithmic frequency axis, shift invariance appears a natural constraint for audio signals modelling.

Thus, we find ample motivation for sparse coding with shift invariance as a starting point for analysis of image and audio signals. We present our ideas in the context of image processing, but we also briefly include an example of their application to audio processing.

In many existing image feature extraction methods, the image is subdivided into patches, $I(x, y)$, of the same size as the desired features. The image patches are modeled as a linear combination of feature images $\Psi_d(x, y)$ [23, 22, 14, 7, 15] $I(x, y) \approx \sum_d \alpha_d \Psi_d(x, y)$. A drawback of this approach is that the extracted features depend on how the image is subdivided while similar features at different locations have to be coded in separate components. To overcome this problem, [13] propose a model which allows each feature to be shifted a given amount within each image patch. In [8, 33] general invariance to transformation was considered. Here, the features within a given patch are invariant to a pre-specified set of linear operators. In [18, 2, 30] shift invariance was considered for time series signals $I(t)$ using sparse coding based on models that can be stated in convolutional form $I(t) \approx \sum_{d,m} \alpha_d(m)\Psi_d(t - m)$, such that $\alpha_d(m)$ codes the degree in which the d^{th} component time series shifted m samples, $\Psi_d(t - m)$, is present. A similar approach was used in [21] to code video images $I(x, y, t)$ as a sparse convolution of component video images $\Psi_d(x, y, t)$, i.e. $I(x, y, t) \approx \sum_{d,m} \alpha_d(m)\Psi_d(x, y, t - m)$. We presently generalize this convolutional form to form a 2D shift invariant sparse coding model that work on the complete image rather than relying on patching the image. We will exploit that convolution can efficiently be implemented through the fast fourier transform (FFT) and generalize the model to accommodate multi-channel image and sound data.

The paper is structured as follows: First, we state our shift invariant sparse coding model and give an algorithm for estimating its parameters. Next, we present a method to find the sparseness parameter in the model based on evaluating the tradeoff between quality of fit and number of non-zero elements in the sparse code. Finally, we demonstrate how the model can identify components that constitute high-level features, i.e. so-called icon alphabets, of both image and music data.

2 Shift Invariant Sparse Coding

The shift invariant sparse coding (SISC) model reads

$$X_c(x, y) \approx L_c(x, y) = \sum_d s_{c,d} \sum_{u,v} \alpha_d(u, v)\Psi_d(x - u, y - v). \tag{1}$$

where $X_c(x, y)$ is the entire image of size $X \times Y$ at channel c, and the code, $\alpha_d(u, v)$, is sparse, i.e., most of its elements are zero. The image is modelled as

a sum of 2-D convolutions of feature images, $\Psi_d(x, y)$, of size $Z \times W$ and codes, $\alpha_d(u, v)$ of size $U \times V$. Due to the sparseness of the code, only a small number of positions are active. For image data the sparse code will be the full image while for audio data U will be set to cover potential changes in pitch of the harmonic structure coded in $\boldsymbol{\Psi}_d$. We have previously used a model similar to equation (1) to separate music signals [26] and similar models have also been derived by [9, 28, 27]. The channel mixing matrix $\mathbf{S}$ encodes the color in images, i.e., the RGB or CMYK channels; for music, $\mathbf{S}$ codes the mixing in multi-channel recordings. The model handles data of more than one channel by assuming that the features are consistent across channels, varying only in strength. For color image data, this means that the features have a specific color; for audio data, it means that the sources are mixed linearly and instantaneously into the channels.

It is naturally to assume that the code, $\alpha_d(u, v)$; the features, $\Psi_d(x, y)$; and the channel mixing parameters, $s_{c,d}$, are non-negative. A non-negative representation is relevant when the data is non-negative since it tends to favor easily interpretable part based representations [16]. Furthermore, non-negativity is a natural constraint both for image data [7, 14] and audio amplitude spectrograms [29, 28, 10, 26]. Finally, if data and model parameters can be assumed non-negative, the component identification is improved by restricting the parameter space to the positive orthant. In the following, we will both derive an unconstrained and a non-negativity constrained algorithm for SISC to compare the results of the two approaches.

The sparseness of the code, $\alpha_d(u, v)$, is needed for several reasons. First of all, the SISC model is over-complete, i.e., the number of parameters is larger than the number of data points. Second, the model is ambiguous even when constrained non-negative if the data does not adequately span the positive orthant [6]. Third, the SISC model suffers from a structural ambiguity, as image features can be arbitrarily represented in $\alpha_d(u, v)$ and $\Psi_d(x, y)$. For instance, a gray scale image can be completely described by a component $\alpha_d(u, v)$ identical to the image with $\Psi_d(x, y)$ having one non-zero entry. By imposing sparseness, the over-complete representation can be resolved [22, 21, 24], and uniqueness improved [7, 14].

2.1 Parameter Estimation

We base our derivation on a Gaussian noise model (i.e. a quadratic distance measure), but it can readily be generalized to other measures of distortion such as Bregman and Csiszár's divergence [17, 4]. We formulate the model in a probabilistic framework, and focus on algorithmic issues for MAP estimation of $\mathbf{S}, \boldsymbol{\alpha}, \boldsymbol{\Psi}$. We assume normal i.i.d. noise, $P(X|\boldsymbol{\alpha}, \boldsymbol{\Psi}, \mathbf{S}, \sigma^2) \sim \prod_{x,y,c} N(X_c(x, y); L_c(x, y), \sigma^2)$ and we enforce sparsity on the sparse code $\boldsymbol{\alpha}$ by the i.i.d. Laplace prior $P(\boldsymbol{\alpha}) = \prod_{d,u,v} \frac{\beta}{2} e^{-\beta|\alpha_d(u,v)|}$ (if $\boldsymbol{\alpha}$ is non-negative the normalization of the prior is trivially changed to β instead of $\frac{\beta}{2}$). To alleviate the scale ambiguity inherent in the SISC model we assign improper uniform priors over the unit hyper-sphere for the feature images $\boldsymbol{\Psi}_d$, $P(\boldsymbol{\Psi}) \propto \prod_d \delta(\|\boldsymbol{\Psi}_d\|_F - 1)$. The channel mixing, we normalize across both features and channels, such that the relative importance

of the features is captured in $\mathbf{S}$, $P(\mathbf{S}) \propto \delta(\|\mathbf{S}\|_F - 1)$. This enables $\mathbf{S}$ to "turn off" excess components, which results in a form of automatic model selection by pruning unimportant features. Using Bayes theorem the joint posterior for $\boldsymbol{\alpha}$, $\boldsymbol{\Psi}$ and $\mathbf{S}$ can be written as

$$P(\boldsymbol{\alpha}, \boldsymbol{\Psi}, \mathbf{S}|X, \sigma^2, \beta) \propto P(X|\boldsymbol{\alpha}, \boldsymbol{\Psi}, \mathbf{S}, \sigma^2)P(\boldsymbol{\alpha}|\beta)P(\boldsymbol{\Psi})P(\mathbf{S}). \tag{2}$$

Ignoring constants and subjecting $\|\boldsymbol{\Psi}_d\|_F = 1$ and $\|\mathbf{S}\|_F = 1$, the negative log-posterior is given by

$$-\log P(\boldsymbol{\alpha}, \boldsymbol{\Psi}, \mathbf{S}|X, \beta') = \frac{1}{2}\sum_{c,x,y}\big(X_c(x,y) - L_c(x,y)\big)^2 + \beta'\sum_{d,u,v}|\alpha_d(u,v)|,$$

where $\beta' = \beta\sigma^2$.

The minimization of the log posterior will be based on gradient descent, for details on the derivation of the gradients see appendix A. For the optimization under non-negativity constraints we derive a set of multiplicative update rules [17] with exponentiated step sizes [25] which provide a simple yet efficient way to estimate the model parameters, see Appendix B. The algorithm for estimating the parameters in the SISC model is given in Algorithm 1. Non-negativity constrained updates are denoted by $\diamond$, unconstrained by $*$. By inspecting the updates it can be seen that both A and B used to form the gradients are derived by a series of 2D convolutions that can be efficiently calculated through the FFT. μ_s, μ_Ψ and μ_α are estimated by line-search, i.e.μ_s is chosen such that $\log P(\boldsymbol{\alpha}^t, \boldsymbol{\Psi}^t, \mathbf{S}^{t+1}|X, \beta') > \log P(\boldsymbol{\alpha}^t, \boldsymbol{\Psi}^t, \mathbf{S}^t|X, \beta')$. For more details of this efficient implementation of the algorithm see the Matlab script available from [20].

2.2 Estimation of the Sparsity Parameter

The sparsity parameter, β', is important to obtain good solutions of the sparse code. A good solution is one which is parsimonious in the sense that the data is well described by a small number of components, i.e., by a good trade-off between the residual error and the sparsity of the code.

There are many different approaches to making this trade-off such as the L-curve [12], generalized cross-validation or Bayesian approach [11]. Here, we base the selection of β' on the concept of the L-curve. The idea is to plot the norm of the regularization versus the residual norm, which gives a graphical display of the compromise between regularization and residual error. An ad-hoc method for finding a good solution is to choose the point of maximum curvature, which corresponds to the "corner" of the L-curve [12]. The L-curve was originally developed in connection with Tikhonov regularization, but the idea generalizes well to minimizing the number of non-zero elements in the sparse code, i.e. L_0-norm minimization. In the following, we plot the reconstruction error $\|E\|_F^2 = \sum_{x,y,c}(X_c(x,y) - L_c(X,y))^2$ against the L_0-norm of the sparse code $\boldsymbol{\alpha}$ and choose the solution as the point of maximum curvature. Notice, we regularize the problem by the Laplace prior corresponding to regularizing by the

Algorithm 1 Shift Invariant Sparse Coding (SISC)

Initialization.
  $t = 0$, $s^0_{c,d}$, $\alpha^0_d(u,v)$, and $\Psi^0_d(x,y)$ random uniform initialized.
repeat
  Update channel mixing parameters.
  $\widetilde{A}_{c,d} = \sum_{x,y} X_c(x,y) \sum_{u,v} \alpha^t_d(u,v) \Psi^t_d(x-u, y-v)$,
  $\widetilde{B}_{c,d} = \sum_{x,y} L_c(x,y) \sum_{u,v} \alpha^t_d(u,v) \Psi^t_d(x-u, y-v)$,
  $A_{c,d} = A_{c,d} + s^t_{c,d} \sum_{c',d'} s^t_{c',d'} \widetilde{B}_{c',d'}$,
  $B_{c,d} = B_{c,d} + s^t_{c,d} \sum_{c',d'} s^t_{c',d'} \widetilde{A}_{c',d'}$.
  $\diamond$ $s^{t+1}_{c,d} \leftarrow s^t_{c,d} \left(\frac{A_{c,d}}{B_{c,d}} \right)^{\mu_s}$
  $*$ $s^{t+1}_{c,d} \leftarrow s^t_{c,d} - \mu_s (B_{c,d} - A_{c,d})$
  $s^{t+1}_{c,d} \leftarrow \frac{s^{t+1}_{c,d}}{\|s^{t+1}\|_F}$.
  Update feature images.
  $\widetilde{A}_d(x,y) = \sum_c s^{t+1}_{c,d} \sum_{u,v} X_c(u,v) \alpha^t_d(u-x, v-y)$,
  $\widetilde{B}_d(x,y) = \sum_c s^{t+1}_{c,d} \sum_{u,v} L_c(u,v) \alpha^t_d(u-x, v-y)$,
  $A_d(x,y) = A_d(x,y) + \Psi^t_d(x,y) \sum_{x'y'} \Psi^t_d(x',y') \widetilde{B}_d(x',y')$,
  $B_d(x,y) = B_d(x,y) + \Psi^t_d(x,y) \sum_{x'y'} \Psi^t_d(x',y') \widetilde{A}_d(x',y')$.
  $\diamond$ $\Psi^{t+1}_d(x,y) \leftarrow \Psi^t_d(x,y) \left(\frac{A_d(x,y)}{B_d(x,y)} \right)^{\mu_\Psi}$
  $*$ $\Psi^{t+1}_d(x,y) \leftarrow \Psi_d(x,y)^t - \mu_\Psi (B_d(x,y) - A_d(x,y))$
  $\Psi^{t+1}_d(x,y) \leftarrow \frac{\Psi^{t+1}_d(x,y)}{\|\Psi^{t+1}_d\|_F}$.
  Update sparse code.
  $A_d(u,v) = \sum_c s^{t+1}_{c,d} \sum_{x,y} X_c(x,y) \Psi^{t+1}_d(x-u, y-v)$,
  $B_d(u,v) = \sum_c s^{t+1}_{c,d} \sum_{x,y} L_c(x,y) \Psi^{t+1}_d(x-u, y-v)$,
  $\diamond$ $\alpha^{t+1}_d(u,v) \leftarrow \alpha_d(u,v)^t \left(\frac{A_d(u,v)}{B_d(u,v)+\beta'} \right)^{\mu_\alpha}$
  $*$ $\alpha^{t+1}_d(u,v) \leftarrow \alpha^{t+1}_d(u,v) - \mu_\alpha (B_d(u,v) - A_d(u,v))$
    $\alpha^{t+1}_d(u,v) = \begin{cases} 0 & \text{if} \quad |\alpha^{t+1}_d(u,v)| < \mu_\alpha \beta' \\ \alpha^{t+1}_d(u,v) - \mu_\alpha \beta' \operatorname{sign}(\alpha^{t+1}_d(u,v)) & \text{otherwise} \end{cases}$
  $t = t + 1$
until convergence.

L_1-norm only because it mimics the behavior of the L_0-norm [5] without introducing additional minima. Thus, we evaluate the quality of regularization by the L_0-norm rather than the L_1-norm. This has the benefit that bias introduced by the L_1-norm regularization leaves the L_0-norm unaffected. Consequently, potential improvements in the tradeoff are only achieved when elements are turned off (set to zero).

3 Results

We evaluated the algorithm on synthetic data as well as real image and music data. The convergence criterion was to stop when the relative change in the log posterior was less than 10^{-6} or at a maximum of 1000 iterations.

Colored Letters Image: To illustrate the SISC algorithm, we created an image which conforms perfectly with the model. The image contains four features; the letters A, B, C, D in different colors. The letters were placed at randomly selected positions. The size of the image is $250 \times 250 \times 3$ (height×width×color channel) and the range of the data is $[0; 765]$. We ran the SISC algorithm both constrained non-negative and unconstrained. We used eight components with image features $\boldsymbol{\Psi}_d$ of size 32×32 in the analysis to ensure that the generating features could be captured by the estimated features. The L-curve method suggested that a value of $\beta' = 100$ was appropriate. The analysis correctly identified the generating image features when β' was chosen according to the L-curve method. The right choice of sparsity is crucial in order to identify the features correctly and turn off excess components. The result of the analysis is illustrated in figure 1. Notice, how both the unconstrained and non-negative constrained analysis give similar results.

Image of honey comb: We next analyzed a gray scale image of a honey comb. As the unconstrained analysis gave identical results we have for brevity only included the non-negativity constrained solution. The size of the image was 160×200 and the range of data $[0; 250]$. We set the feature images $\boldsymbol{\Psi}_d$ to have size 25×25. From figure 2 it can be seen that for an adequate value of $\beta' = 50$ the image is coded into two features. One coding for the hexagon shape of the cubes, the other coding whether the hexagons are filled or empty - hence constituting an efficient icon alphabet to code for the image. Notice, how the sparse code $\boldsymbol{\alpha}$ shown for the first component codes where in the image the hexagon shape is present.

Image of Brick house: Next, we performed a SISC analysis of a color photograph of a brick house, see figure 3. The image data was of size $432 \times 576 \times 3$ with range $[0; 255]$. The size of the feature images $\boldsymbol{\Psi}_d$ were 25×25. The non-negative SISC analysis captures components primarily corresponding to the brick wall, vertical lines in window and fence, the sky, horizontal lines and the window grille, i.e. forming an icon alphabet of various parts of the image. The unconstrained SISC model on the other hand form more complex patters since components are allowed to be subtractive and does as such not yield features pertaining to specific parts of the image. Hence, non-negativity favor part based representation as also reported in [16] thus help to form more interpretable icon alphabets constituting the image.

Single channel recording of mixed organ and piccolo: We analyzed the single channel music of mixed organ and piccolo described in [34]. The analysis is based on the amplitude of the log-spectrogram, and the data has previously been analyzed by [34] using a harmonic structure model, i.e. by supervised learning the harmonic structure of each instrument and then separate a mixed signal of the instruments using these learned structures. Presently, we use the SISC algo-

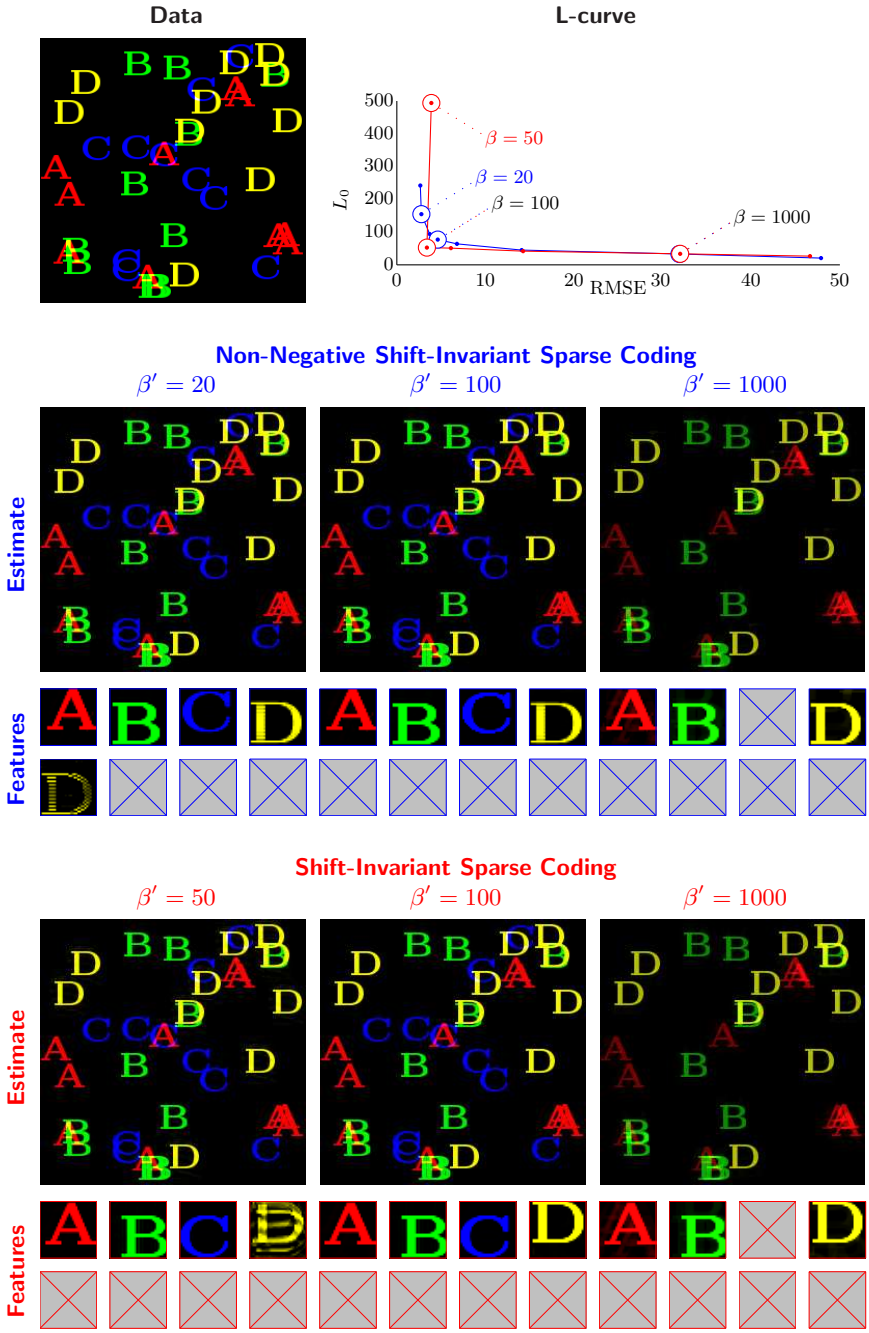

Fig. 1. SISC analysis of colored letters image. To the top the colored letter image is given as well as the L-curve showing the tradeoff between reconstruction and sparsity of the code given for the non-negative (blue) and unconstrained (red) estimation. Below is given the results for different values of β' including the value with optimal tradeoff in the L-curve ($\beta' = 100$). With this optimal value of β' the four letters constituting the image are identified and excess components turned off.

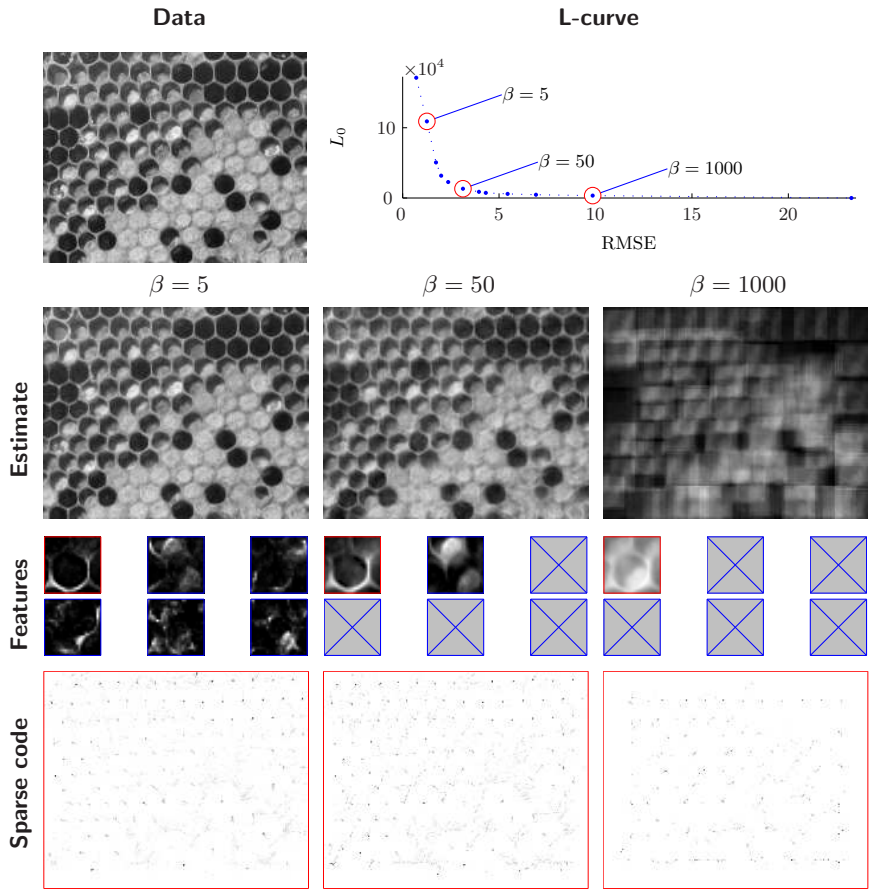

Fig. 2. Non-negative SISC analysis of an image of a honey cube (unconstrained analysis gave similar results). To the top is given the image as well as the L-curve of reconstruction error vs. sparsity of the code. From the curve a good tradeoff is found when $\beta' = 50$. The feature extracted correspond for this value to a feature coding the hexagon shape of the cubes and a feature coding weather the hexagon is filled or not forming an efficient icon alphabet for the image. At the bottom is given the sparse code $\boldsymbol{\alpha}$ for the first component coding where the hexagon feature is present in the image. Notice, again how sparsity turns off excess components.

rithm unsupervised on the mixed signal of the two instruments to both learn the harmonic structures of each instrument as well as which notes were played such that the mixed signal can be separated by identifying what parts of the log spectrogram originates from each instrument. Since both the extracted harmonic structures $\boldsymbol{\Psi}_d$ and scores coded in $\boldsymbol{\alpha}_d$ should be non-negative we fitted a four

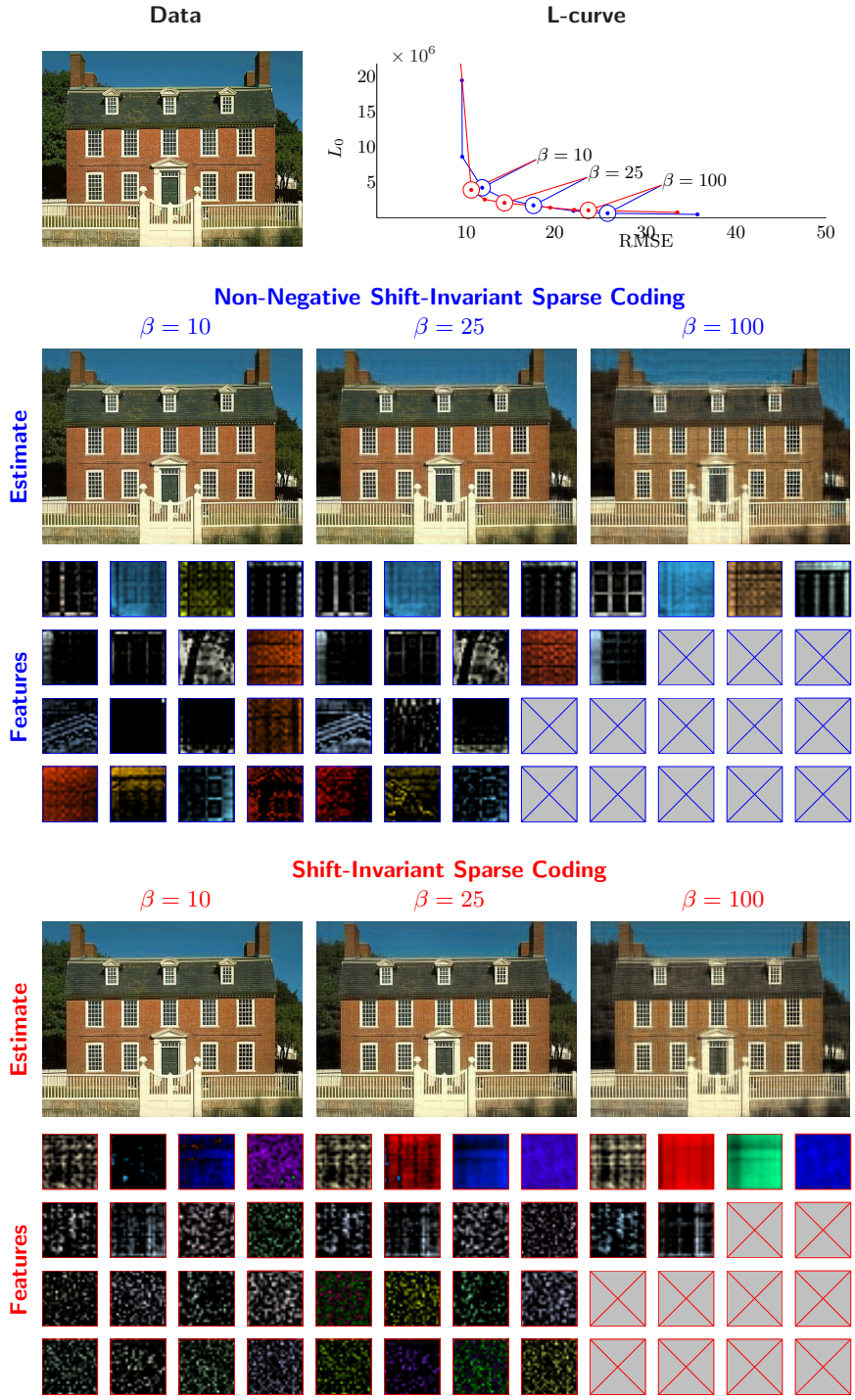

Fig. 3. A 16 component SISC analysis of a color photograph of a brick house. **Top:** The photograph of the house and the L-curves obtained plotting the reconstruction error versus number of non-zero elements in the sparse code $\boldsymbol{\alpha}$. **Center:** A 16 component non-negative SISC analysis given for $\beta' = \{10, 25, 100\}$. Clearly, features pertaining to the image correspond to the brick wall, vertical lines in window frames and fence, the sky, horizontal lines, and the window grilles have been extracted forming icon alphabets for the image. **Bottom:** A 16 component SISC analysis given for $\beta' = \{10, 25, 100\}$. Although features are extracted coding for colors the features constitute more complex patterns rather than pertaining to specific parts of the image.

component non-negative SISC model to the data. [1] From figure 4 it can be seen that the non-negative SISC analysis extracts the two instruments in two separate components while excess components are turned off at the optimal tradeoff between reconstruction and sparsity of the code ($\beta' = 50$). The spectrogram of the instruments are coded into their harmonic structure $\boldsymbol{\Psi}_d$ as well as when and at what pitch this structure is present (i.e. the scores of the instrument) coded in the sparse code $\boldsymbol{\alpha}_d$.

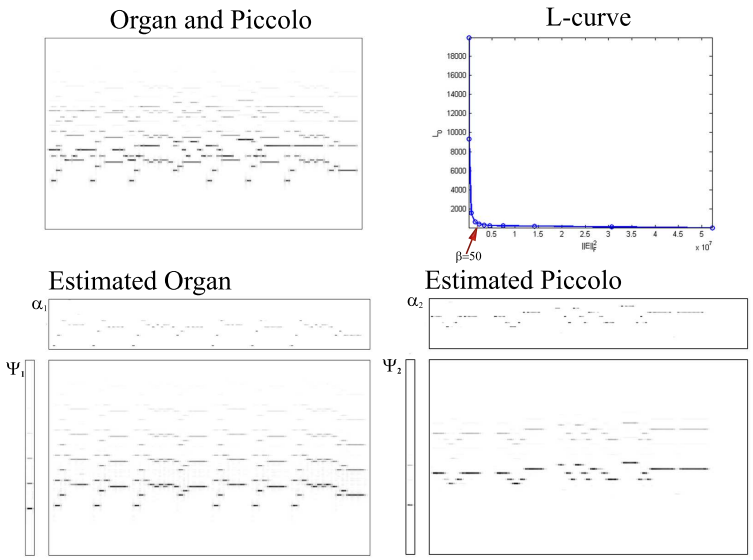

Fig. 4. Non-negative SISC analysis of the amplitude of the log-spectrogram of a music signal. **Top:** spectrogram of the mixed signal of the organ and piccolo as well as the L-curve obtained plotting the reconstruction error versus number of non-zero elements in the sparse code $\boldsymbol{\alpha}$. **Bottom:** result obtained when analyzing the mixed spectrogram using a 4-component single channel SISC model. From the L-curve, $\beta' = 50$ was used (the values of β' just around $\beta' = 50$ gave similar results). With this choice of β' two components were turned off. The reconstructed spectrograms of the two remaining components correspond well to the organ and piccolo respectively. Furthermore, the harmonics of each instruments is given by $\boldsymbol{\Psi}_d$ to the left of the reconstructed spectrograms while the scores played is indicated in the sparse code $\boldsymbol{\alpha}_d$ shown above the reconstructed spectrogram.

[1] The music was sampled at 22 kHz and analyzed by a short time Fourier transform based on a 8192 point Hanning window with 50% overlap providing a total of 146 FFT frames. We grouped the spectrogram into 373 logarithmically spaced frequency bins in the range of 50 Hz to 11 kHz with 48 bins per octave, which corresponds to four bins per half tone. We chose $Z = 373$ and $W = 4$ while $U = 97$ covering 2 octaves, i.e. slightly more than the range of the notes played while $V = 146$.

4 Discussion

Although, the SISC model is highly overcomplete, the L_1-norm regularization is able to resolve the ambiguity of the representation and to find the correct model order by turning off excess components. However, for identification of the important features the choice of the regularization parameter β' is important. Too low values lead to ambiguous results while too large regularization removed important features of the data. From the proposed L-curve approach a good value of β' could be found such that the important features of the data were identified while excess components turned off. Hence, the L_1-norm regularization worked as a method for automatic relevance detection. We conclude that the value of β' with the maximum curvature in the plot of the reconstruction error against the L_0-norm of the sparse code $\boldsymbol{\alpha}$ is very useful for the present SISC model. This approach should also be useful for other types of L_1 constrained models such as sparse coding and sparse NMF [22, 7, 14].

The sisc model is capable of identifying relevant features of both image and audio data that form icon alphabets for the data. While both the unconstrained and non-negative constrained model found relevant features of both the image data of letters and honey cubes non-negativity favor a more part based representation than the unconstrained optimization. Thus, for the analysis of the brick house features more closely constituting specific parts of the image was extracted when restricting the model to the positive orthant. In the analysis of the music data, the SISC model assumes a constant timbre, i.e., no change in the structure of the harmonics over pitch. In general, each component is likely to work only within limited changes of pitch. Despite this shortcoming, the SISC model seem promising in finding prominent higher-level features of multi-media data. Future work will focus on feature extraction with more general types of invariance such as invariance to scale and rotation. Presently, additional channels were modeled as linear mixtures of the feature images $\boldsymbol{\Psi}$. Alternatively, the channel information can be directly coded in the feature images $\boldsymbol{\Psi}$. This should be investigated in future work.

A Derivation of the sisc algorithms

To incorporate the constraints $\|\boldsymbol{\Psi}_d\|_F = 1$ and $\|\mathbf{S}\|_F = 1$ we recast the log likelihood in the normalization invariant variables $\widetilde{\Psi}_d(x,y) = \frac{\Psi_d(x,y)}{\|\boldsymbol{\Psi}_d\|_F}$ and $\widetilde{s}_{c,d} = \frac{s_{c,d}}{\|S\|_F}$ such that $\widetilde{L}_c(x,y) = \sum_d \widetilde{s}_{c,d} \sum_{u,v} \alpha_d(u,v)\widetilde{\Psi}_d(x-u, y-v)$. Hence, by formulating the log likelihood in these variables $\boldsymbol{\Psi}_d$ and $\mathbf{S}$ can be normalized at each iteration without impacting the log-likelihood, i.e.

$$-\log P(\boldsymbol{\alpha}, \widetilde{\boldsymbol{\Psi}}, \widetilde{\mathbf{S}}|X, \beta') = \frac{1}{2}\sum_{c,x,y}\left(X_c(x,y) - \widetilde{L}_c(x,y)\right)^2 + \beta' \sum_{d,u,v} |\alpha_d(u,v)|.$$

The gradients are derived by differentiation by parts. Differentiating a given element of $\widetilde{L}_c$ with respect to a given element of $\alpha_d(u,v)$ and $\widetilde{\Psi}_d(x,y)$ and $\widetilde{s}_{c,d}$

gives

$$\frac{\partial \widetilde{L}_c(x,y)}{\partial \widetilde{\Psi}_{d'}(x',y')} = \frac{\partial \sum_d \widetilde{s}_{c,d} \sum_{u,v} \alpha_d(u,v)\widetilde{\Psi}_d(x-u,y-v)}{\partial \widetilde{\Psi}_{d'}(x',y')} = \widetilde{s}_{c,d'}\alpha_{d'}(x-x',y-y'),$$

$$\frac{\partial \widetilde{L}_c(x,y)}{\partial \alpha_{d'}(u',v')} = \frac{\partial \sum_d \widetilde{s}_{c,d} \sum_{u,v} \alpha_d(u,v)\widetilde{\Psi}_d(x-u,y-v)}{\partial \alpha_{d'}(u',v')} = \widetilde{s}_{c,d'}\widetilde{\Psi}_{d'}(x-u',y-v'),$$

$$\frac{\partial \widetilde{L}_c(x,y)}{\partial \widetilde{s}_{c',d'}} = \frac{\partial \sum_d \widetilde{s}_{c,d} \sum_{u,v} \alpha_d(u,v)\widetilde{\Psi}_d(x-u,y-v)}{\partial \widetilde{s}_{c',d'}} = \sum_{u,v} \alpha_{d'}(u,v)\widetilde{\Psi}_{d'}(x-u,y-v).$$

Furthermore, the derivatives of $\widetilde{s}_{c,d}$ and $\widetilde{\Psi}_d(x,y)$ with respect to $s_{c,d}$ and $\Psi_d(x,y)$ is given by

$$\frac{\partial \widetilde{s}_{c',d'}}{\partial s_{c',d'}} = \frac{\partial \frac{s_{c',d'}}{\|\mathbf{S}\|_F}}{\partial s_{c',d'}} = \frac{1}{\|\mathbf{S}\|_F} - s_{c',d'} \sum_{c,d} \frac{s_{c,d}}{\|\mathbf{S}\|_F^3},$$

$$\frac{\partial \widetilde{\Psi}_{d'}(x',y')}{\partial \Psi_{d'}(x',y')} = \frac{\partial \frac{\Psi_{d'}(x',y')}{\|\boldsymbol{\Psi}_{d'}\|_F}}{\partial \Psi_{d'}(x',y')} = \frac{1}{\|\boldsymbol{\Psi}_{d'}\|_F} - \Psi_{d'}(x',y') \sum_{x,y} \frac{\Psi_{d'}(x,y)}{\|\boldsymbol{\Psi}_{d'}\|_F^3}.$$

Thus, by differentiation by parts we now find for instance when differentiating the negative log-likelihood with respect to $\Psi_{d'}(x',y')$

$$\frac{\partial -\log P}{\partial \Psi_{d'}(x',y')} = -\sum_{x,y,c} (X_c(x,y) - \widetilde{L}_c(x,y)) \frac{\partial \widetilde{L}_c(x,y)}{\partial \widetilde{\Psi}_{d'}(x',y')} \frac{\partial \widetilde{\Psi}_{d'}(x',y')}{\partial \Psi_{d'}(x',y')}. \tag{3}$$

The variables are then updated by gradient descent or for the non-negative SISC by the multiplicative updates described in the following section.

B Multiplicative updates

Multiplicative updates were introduced in [16, 17] for non-negative matrix factorization (NMF). Although, other types of updates exists for non-negativity constraint optimization such as projected gradient [19] and active sets [3], multiplicative updates are simple to implement and extend well to sparse coding [7]. Consider the objective function $C(\theta)$ of the non-negative variables θ. Let further $\frac{\partial C(\theta)_i^+}{\partial \theta_i}$ and $\frac{\partial C(\theta)_i^-}{\partial \theta_i}$ be the positive and negative part of the derivative with respect to θ_i. Then the multiplicative update has the following form:

$$\theta_i \leftarrow \theta_i \left(\frac{\frac{\partial C(\theta)^-}{\partial \theta_i}}{\frac{\partial C(\theta)^+}{\partial \theta_i}} \right)^{\mu}. \tag{4}$$

A small constant $\varepsilon = 10^{-9}$ can be added to the denominator to avoid potential division by zero. By also adding the constant to the numerator the corresponding gradient is unaltered. When the gradient is zero $\frac{\partial C(\theta)}{\partial \theta_i}^+ = \frac{\partial C(\theta)^-}{\partial \theta_i}$ such that θ is left unchanged. If the gradient is positive $\frac{\partial C(\theta)^+}{\partial \theta_i} > \frac{\partial C(\theta)^-}{\partial \theta_i}$ hence θ_i will decrease and vice versa if the gradient is negative. Thus, there is a one-to-one relation between fixed points of the multiplicative update rule and stationary points under gradient descend. One attractive property of multiplicative updates is that, since θ_i, $\frac{\partial C(\theta)^+}{\partial \theta_i}$ and $\frac{\partial C(\theta)^-}{\partial \theta_i}$ all are non-negative, non-negativity is naturally enforced as each update remains in the positive orthant. μ is a step size parameter that potentially can be tuned to assist convergence. When $\mu \to 0$ only very small steps in the negative gradient direction are taken.

References

1. H. Asari, B. A. Pearlmutter, A. M. Zador, Sparse representations for the cocktail party problem, Journal of Neuroscience 26 (28) (2006) 7477–7490.
2. T. Blumensath, M. Davies, On shift-invariant sparse coding, International Conference on Independent Component Analysis and Blind Source Separation 26 (2004) 1205–1212.
3. R. Bro, S. de Jong, A fast non-negativity-constrained least squares algorithm, J. of Chemometrics 11 (5) (1997) 393–401.
4. A. Cichocki, R. Zdunek, S. Amari, Csiszar's divergences for non-negative matrix factorization: Family of new algorithms, 6th International Conference on Independent Component Analysis and Blind Signal Separation (2006) 32–39.
5. D. Donoho, For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution, Communications on Pure and Applied Mathematics 59 (6) (2006) 797–829.
6. D. Donoho, V. Stodden, When does nonnegative matrix factorization give a correct decomposition into parts?, in: S. Thrun, L. Saul, B. Schölkopf (eds.), Advances in Neural Information Processing Systems 16, MIT Press, Cambridge, MA, 2004.
7. J. Eggert, E. Körner, Sparse coding and nmf, in: Neural Networks, vol. 4, 2004, pp. 2529–2533.
8. J. Eggert, H. Wersing, E. Körner, Transformation-invariant representation and nmf, in: Neural Networks, vol. 4, 2004, pp. 2535– 2539.
9. D. FitzGerald, E. Coyle, Sound source separation using shifted non-negative tensor factorisation, in: ICASSP2006, 2006.
10. D. FitzGerald, M. Cranitch, E. Coyle, Non-negative tensor factorisation for sound source separation, in: proceedings of Irish Signals and Systems Conference, 2005, pp. 8–12.
11. L. K. Hansen, K. H. Madsen, T. Lehn-Schiøler, Adaptive regularization of noisy linear inverse problems, in: Proceedings of Eusipco 2006, 2006.
URL http://www2.imm.dtu.dk/pubdb/p.php?4417
12. P. C. Hansen, Analysis of discrete ill-posed problems by means of the l-curve, SIAM Review 34 (4) (1992) 561–580.
13. W. Hashimoto, K. Kurata, Properties of basis functions generated by shift invariant sparse representations of natural images, Biol. Cybern. 83 (2000) 111–118.
14. P. O. Hoyer, Non-negative matrix factorization with sparseness constraints, Journal of Machine Learning Research.

15. A. Hyvrinen, E. Oja, Independent component anlayhsis: Algorithms and application, Neural Networks 13 (2000) 411–430.
16. D. Lee, H. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–91.
17. D. D. Lee, H. S. Seung, Algorithms for nonnegative matrix factorization, in: NIPS, 2000, pp. 556–562.
18. M. S. Lewicki, T. J. Sejnowski, Coding time-varying signals using sparse shift-invariant representations, Adv. Neural Inform. Process. Systems (NIPS'99) 11 (1999) 730–736.
19. C.-J. Lin, Projected gradient methods for non-negative matrix factorization, Neural Computation 19 (2007) 2756–2779.
20. M. Mørup, M. N. Schmidt, www2.imm.dtu.dk/pubdb/views/edoc_download.php/4652/zip/imm4652.zip. (2007).
21. B. A. Olshausen, Learning sparse, overcomplete representations of time-varying natural images, Image Processing, ICIP 2003. Proceedings. 2003 International Conference 1 (2003) 41–44.
22. B. A. Olshausen, D. J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (1996) 607–609.
23. B. A. Olshausen, D. J. Field, Sparse coding of sensory inputs, Current Opinion in Neurobiology 14 (2004) 481–487.
24. B. A. Olshausen, J. Field, David, Sparse coding with an overcomplete basis set: A strategy employed by v1, Vision Research 37 (23) (1997) 3311–3325.
25. R. Salakhutdinov, S. Roweis, Z. Ghahramani, On the convergence of bound optimization algorithms, in: Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03), Morgan Kaufmann Publishers, San Francisco, CA, 2003, pp. 509–516.
26. M. N. Schmidt, M. Mørup, Nonnegative matrix factor 2-d deconvolution for blind single channel source separation, Independent Component Analysis and Blind Signal Separation, pages 700-707, 2006 (2006) 700–707.
27. B. R. Smaragdis, P, M. Shashanka, Sparse and shift-invariant feature extraction from non-negative data, In proceedings IEEE International Conference on Audio and Speech Signal Processing.
28. P. Smaragdis, Non-negative matrix factor deconvolution; extraction of multiple sound sourses from monophonic inputs, International Symposium on Independent Component Analysis and Blind Source Separation (ICA) 3195 (2004) 494.
29. P. Smaragdis, J. C. Brown, Non-negative matrix factorization for polyphonic music transcription, IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) (2003) 177–180.
30. E. Smith, M. S. Lewicki, Efficient coding of time-relative structure using spikes., Neural Computation 17 (2005) 19–45.
31. K. Tanaka, Representation of visual features of objects in the inferotemporal cortex, Neural Networks 9 (8) (1996) 1459–1475.
32. S. E. Trehub, The development origins of musicality, Nature Neuroscience 6 (7) (2003) 669–673.
33. H. Wersing, J. Eggert, E. Korner, Sparse coding with invariance constraints, Proc. Int. Conf. Artificial Neural Networks ICANN (2003) 385–392.
34. Y.-G. Zhang, C.-S. Zhang, Separtation of music signals by harmonic structure modeling, Proceedings of Neural Information Processing Systems (NIPS) (2005) 184–191.