Journal of Machine Learning Research (2007) Submitted ; Published

Shift Invariant Sparse Coding of Image and Music Data

Morten Mørup and **Mikkel N. Schmidt** and **Lars K. Hansen**
Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, Building 321
2800 Kgs Lyngby
email: {mm,mns,lkh}@imm.dtu.dk

Editor: n/a

Abstract

Sparse coding is a well established principle for unsupervised learning. Traditionally, features are extracted in sparse coding in specific locations, however, often we would prefer a shift invariant representation. This paper introduces the shift invariant sparse coding (SISC) model. The model decomposes an image into shift invariant feature images as well as a sparse coding matrix indicating where and to what degree in the original image these features are present. The model is not only useful, for analyzing shift invariant structures in image data, but also for analyzing the amplitude spectrogram of audio signals since a change in pitch relates to a shift in a logarithmic frequency axis. The SISC model is extended to handle data from several channels under the assumption that each feature is linearly mixed into the channels. For image analysis this implies that each feature has a fixed color coding for all locations. While for analysis of audio signals it means that features have fixed spatial position. The model is overcomplete and we therefore invoke sparse coding. The optimal degree of sparseness is estimated by an 'L-curve'-like argument. We propose to 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 in the sparse coding matrix. With this choice of regularization, the algorithm can correctly identify components of non-trivial artificial as well as real image and audio data. For image data, the algorithm identify relevant patterns and the sparse coding matrix indicates where and to what degree these patterns are present. When applied to music, the model can identify the harmonic structures of instruments, while the sparse coding matrix accounts for the notes played.

Keywords: Sparse coding, shift invariance, 2D deconvolution, multiplicative updates, NMF, L-curve.

1. Introduction

Sparse coding and the closely related independent component analysis (ICA) are well established principles for feature extraction (Olshausen and Field, 2004; Olshausen, 1996; Hoyer, 2002; Eggert and Korner, 2004; Olshausen and Field, 1997; Hyvärinen and Hoyer, 2001; Lee and Lewicki, 2002; Hyvarinen et al., 2001; Hoyer and Hyvärinen, 2000). Olshausen and Field (2004) 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 (Tanaka, 1996). Based on Tanaka's observation Hashimoto and Kurata (2000) formulated a model that estimates such shift invariant image features. The resulting features are complex patterns rather than the Gabor-like features often ob-

tained by sparse coding or ICA decomposition (Olshausen, 1996; Hyvärinen and Oja, 2000). These shift invariant features can potentially constitute an icon alphabet.

It has also been demonstrated that sparse over-complete linear representations solve hard acoustic signal processing problems (Asari et al., 2006). 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 (Trehub, 2003), 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 $\Psi_d(x,y)$ (Olshausen, 1996; Lee and Lewicki, 2002; Hoyer and Hyvärinen, 2000; Hyvärinen and Hoyer, 2001; Olshausen and Field, 1997; Hyvärinen and Oja, 2000)

$$I(x,y) \approx \sum_d \alpha_d \Psi_d(x,y). \tag{1}$$

A drawback of this approach is that the extracted features depend on how the image is subdivided. To overcome this problem, Hashimoto and Kurata (2000) propose a model which incorporates shift invariance. Here, each image patch, $I(x,y)$, is modelled by a linear combination of feature images, Ψ_d, which can be translated based on the model

$$I(x,y) \approx \sum_d \alpha_d \Psi_d(x-u_d, y-v_d). \tag{2}$$

Direct estimation of u_d and v_d by exhaustive search is time consuming, but by estimating these parameters independently, the algorithm is computationally feasible (Hashimoto and Kurata, 2000). Since the model only allows for one fixed translation, u_d, v_d, of each feature in each image patch, it will not lead to a compact representation if a specific feature is present more than once within the same patch. This is for example relevant when the image contains a repeated pattern which is not aligned with the patches.

Transformation invariance is a generalization of shift invariance (Eggert et al., 2004; Wersing et al., 2003). Here, the features are invariant to a pre-specified set of linear operators, T_m

$$I(x,y) \approx \sum_{d,m} \alpha_{d,m} (T_m \Psi_d)(x,y). \tag{3}$$

These operators can account for more involved transformations within each patch such as scaling, rotation, etc. The model we present in this paper incorporates only shift invariance, but it can be generalized to rotational invariance.

The paper is structured as follows: First, we state our shift invariant sparse coding model and give an algorithm for estimating its parameters. Then, 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 coding matrix. Next, we demonstrate how the model can identify the components of synthetic data as well as capture important features of real images and music. Finally, we discuss the properties of the model and the features which it extracts. A Matlab implementation of the algorithm is available online (Mørup and Schmidt, 2007).

2. Shift Invariant Sparse Coding

The model for shift invariant sparse coding is based on the following main ideas and observations

- Analysis is performed on the entire image without subdividing the image into patches.
- The estimation of the positions of the features in the image is handled by sparse coding rather than exhaustive search.
- Shift invariance can be stated in terms of 2D-convolution of the feature and a code matrix, which enables efficient computation by the fast Fourier transform (FFT).
- Non-negativity constraints lead to a parts-based representation.
- The model can be generalized to multi-channel analysis such as color images and stereo/multi-channel audio.

Formally, the SISC model can be stated as

$$X(x,y) \approx \sum_{d,u,v} \alpha_d(u,v)\Psi_d(x-u,y-v), \tag{4}$$

where $X(x,y)$ is the entire image of size $I \times J$, 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 $M_1 \times M_2$ and codes, $\alpha_d(u,v)$ of size $K_1 \times K_2$. If $K_1 = I$ and $K_2 = J$ the model allows for each feature to be present at all possible positions in the image; however, because of 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 $K_1 < I$.

Image and music data often contains several channels. In images channels can for example code for color, i.e., the RGB or CMYK channels; for music, channels can for example represent stereo or multiple channels recorded with an array of microphones. We extend the model to handle data of more than one channel by assuming that the features have the same structure in each channel, varying only in amplitude. For color image data, it means that the features have a specific color; for audio data, it means that the sources are mixed linearly and instantaneously into the channels. With this extension, the SISC model reads

$$X_c(x,y) \approx \sum_d s_{c,d} \sum_{u,v} \alpha_d(u,v)\Psi_d(x-u,y-v). \tag{5}$$

Without shift invariance, i.e. $K_1 = 1$, $K_2 = J$ and with features of size $M_1 = I$, $M_2 = 1$, this model corresponds to the PARAFAC model (Harshman, 1970; Carroll and Chang, 1970). Note, the model has not previously been used for image analysis, however, we have previously used a model based on equation (4) to separate music signals (Schmidt and Mørup, 2006). Also, similar models have been used by (FitzGerald and Coyle, 2006; Smaragdis, 2004). However, none of the previous work has been based on sparse coding.

2.1 Non-negativity and Sparseness

We 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 the features cannot cancel each other, the "whole" is modeled as the sum of its "parts", which often results in easily interpretable features (Lee and Seung, 1999). Non-negativity is a natural constraint for image data (Lee and Seung, 1999; Hoyer, 2002, 2004). For audio analysis based on the spectrogram non-negativity is also useful (Smaragdis and Brown, 2003; Smaragdis, 2004; Wang and Plumbley, 2005; FitzGerald et al., 2005; Schmidt and Mørup, 2006).

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 if the data does not adequately span the positive orthant (Donoho and Stodden, 2003). Third, the SISC model suffers from a structural ambiguity, as image features can be arbitrarily represented in $\alpha_d(u,v)$ and $\Psi_d(x,y)$ (see for example Figure 1). By imposing sparseness, the over-complete representation can be resolved (Olshausen, 1996, 2003; Olshausen and Field, 1997), and uniqueness is improved (Eggert and Korner, 2004; Hoyer, 2002, 2004).

2.2 Parameter Estimation

We derive an algorithm for estimating the parameters of the SISC model, based on a generalization of the multiplicative updates for non-negative matrix factorization (NMF) (Lee and Seung, 1999, 2000; Lee et al., 2002). We base our derivation on a quadratic distance measure, but it can be generalized using other distance measures such as Bregman and Csiszár's divergence (Lee and Seung, 2000; Cichocki et al., 2006; Dhillon and Sra, 2005).

We enforce sparsity using an L_1-norm penalty on the code, similar to the approach of Eggert and Korner (2004). The L_1-norm is a good approximation to the L_0-norm; i.e., it minimizes the number of non-zero elements (Donoho, 2006), and it does so while preserving the convexity properties of the cost-function. Note, in SISC model the optimization problem is convex in each of the variables, $s_{c,d}$, $\alpha_d(u,v)$, and $\Psi_d(x,y)$, when the other two parameter sets are fixed, however, the joint estimation problem is not convex.

The cost function can be written as

$$C(\theta) = \frac{1}{2}\sum_{c,x,y}\big(X_c(x,y) - L_c(x,y)\big)^2 + \beta\sum_{d,u,v}\alpha_d(u,v), \tag{6}$$

where θ denotes all the parameters of the model, and

$$L_c(x,y) = \sum_d \widetilde{s}_{c,d}\sum_{u,v}\alpha_d(u,v)\widetilde{\Psi}_d(x-u,y-v), \tag{7}$$

$$\widetilde{s}_{c,d} = \frac{s_{c,d}}{\sqrt{\sum_{c',d'} s_{c',d'}^2}}, \quad \text{and} \quad \widetilde{\Psi}_d(x,y) = \frac{\Psi_d(x,y)}{\sqrt{\sum_{x',y'}\Psi_d^2(x',y')}}. \tag{8}$$

The normalization of $s_{c,d}$ and $\Psi_d(x,y)$ is necessary to avoid trivially minimizing the L_1-norm penalty by letting the elements of $\alpha_d(u,v)$ go to zero while the elements of $s_{c,d}$ and $\Psi_d(x,y)$ grow accordingly. The channel mixing, $s_{c,d}$, we normalize across both features and channels, such that the relative importance of the features is captured. This enables $s_{c,d}$ to "turn off" excess components, which results in a form of automatic model selection by pruning unimportant features.

To minimize the cost function, we derive a set of multiplicative update rules which provide a simple yet efficient way to estimate the model parameters. Alternatively, one could estimate the parameters using another optimization method such as projected gradient (Lin, 2007). An attractive property of multiplicative updates is that non-negativity is automatically ensured. When $\partial_{\theta_i}C(\theta)^+$ and $\partial_{\theta_i}C(\theta)^-$ denote the positive and negative terms in the partial derivative of the cost function with respect to θ_i, the multiplicative updates have the following form

$$\theta_i \leftarrow \theta_i\left(\frac{\partial_{\theta_i}C(\theta)^-}{\partial_{\theta_i}C(\theta)^+}\right)^{\gamma}, \tag{9}$$

A small constant ε is added to the denominator to avoid dividing by zero. By adding the same constant to the numerator the overall gradient is unchanged. γ is an over-relaxation learning rate which can be adaptively tuned (Salakhutdinov et al., 2003). Based on this, an algorithm for estimating the parameters in the SISC model can be stated as follows

1. Initialize $s_{c,d}$, $\alpha_d(u,v)$, and $\Psi_d(x,y)$ with random uniform distributed numbers.
2. Update channel mixing parameters.
   - $\diamond\ A_{c,d} = \sum_{x,y} X_c(x,y) \sum_{u,v} \alpha_d(u,v)\Psi_d(x-u,y-v)$,
   - $\diamond\ B_{c,d} = \sum_{x,y} L_c(x,y) \sum_{u,v} \alpha_d(u,v)\Psi_d(x-u,y-v)$,
   - $\diamond\ s_{c,d} \leftarrow s_{c,d} \dfrac{A_{c,d} + s_{c,d} \sum_{c',d'} s_{c',d'} B_{c',d'}}{B_{c,d} + s_{c,d} \sum_{c',d'} s_{c',d'} A_{c',d'}}$,
   - $\diamond\ s_{c,d} \leftarrow \dfrac{s_{c,d}}{\sqrt{\sum_{c',d'} s_{c',d'}^2}}$.
3. Update feature images.
   - $\diamond\ A_d(x,y) = \sum_c s_{c,d} \sum_{u,v} X_c(u,v)\alpha_d(u-x,v-y)$,
   - $\diamond\ B_d(x,y) = \sum_c s_{c,d} \sum_{u,v} L_c(u,v)\alpha_d(u-x,v-y)$,
   - $\diamond\ \Psi_d(x,y) \leftarrow \Psi_d(x,y) \dfrac{A_d(x,y) + \Psi_d(x,y) \sum_{x'y'} \Psi_d(x',y') B_d(x',y')}{B_d(x,y) + \Psi_d(x,y) \sum_{x'y'} \Psi_d(x',y') A_d(x',y')}$,
   - $\diamond\ \Psi_d(x,y) \leftarrow \dfrac{\Psi_d(x,y)}{\sqrt{\sum_{x',y'} \Psi_d^2(x',y')}}$.
4. Update sparse code.
   - $\diamond\ A_d(u,v) = \sum_c s_{c,d} \sum_{x,y} X_c(x,y)\Psi_d(x-u,y-v)$,
   - $\diamond\ B_d(u,v) = \sum_c s_{c,d} \sum_{x,y} L_c(x,y)\Psi_d(x-u,y-v)$,
   - $\diamond\ \alpha_d(u,v) \leftarrow \alpha_d(u,v) \dfrac{A_d(u,v)}{B_d(u,v)+\beta}$.
5. Repeat from step 2 until convergence.

2.3 Estimation of the Sparsity Parameter

The sparsity parameter, β, is important to obtain good solutions to the sparse coding problem. 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., 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 (Hansen, 1992; Lawson and Hanson, 1974), generalized cross-validation (Golub et al., 1979), a Bayesian approach (Hansen et al., 2006), etc. 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 (Hansen, 1992). The L-curve was originally developed in connection with Tikhonov regularization, but the idea generalizes well to 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 matrix and choose the solution as the point of maximum curvature. Notice, we regularize the problem by the L_1-norm only because it mimics the behavior of the L_0-norm (Donoho, 2006) 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. Experimental 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 cost function was less than 10^{-6} or at a maximum of 1000 iterations.

3.1 Colored Letters Image

To illustrate the SISC algorithm, we created an image which conforms perfectly with the model. The image contains six features; the letters A, E, I, O, U, and Y in different colors and with a maximum height and width of 12 pixels. The letters were placed at 400 randomly selected positions. The size of the image is $224 \times 200 \times 3$ (height×width×color channel) and the range of the data is $[0, 255]$.

We then analyzed the image with the SISC algorithm. We used eight features of size 25×25 in the analysis to ensure that the generating features could be captured in the estimated features. The L-curve method suggested that a value of $\beta = 15$ 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 was crucial in order to identify the features correctly. The result of the analysis is illustrated in Figure 1.

3.2 Oriental Straw Cloth Image

Next, we evaluated the SISC algorithm on a black and white photograph of an oriental straw cloth (Brodatz, 1966). The image displays a repeated weave pattern; its size is 201×201 and the range is $[0, 255]$.

We analyzed the image with the SISC algorithm using four features of size 25×25. The L-curve method suggested using $\beta = 250$ (the values around $\beta = 250$ gave similar results), and based on this, the analysis resulted in only one component, which corresponds well to what we would believe to be the main pattern of the cloth. The result of the analysis is illustrated in Figure 2.

3.3 Brick House Image

Next, we performed a SISC analysis of a color photograph of a brick house. The image data was of size $432 \times 576 \times 3$ with range $[0, 255]$.

The analysis captures components primarily corresponding to the brick wall, vertical lines in window and fence, the sky, horizontal lines and the window grille, see Figure 3.

3.4 Single Channel Music

For the analysis of the amplitudes of the log-spectrogram of music signals the SISC-model should theoretically display the u-th note at time v played by the d-th instrument in $\alpha_d(u, v)$ while the harmonic of the d-th instruments at relative frequency x at time lag (echo) y is captured by $\mathbf{\Psi}_d(x, y)$. Ideally, $s_{c,d}$ captures the strength in which the d-th instrument is present in the c-th audio channel.

Presently, we will analyze the single channel music described in (Zhang and Zhang, 2005). The analysis is based on the amplitude of the log-spectrogram of the music signal consisting of an organ and a piccolo flute mixed together. This data has previously been analyzed by Zhang and Zhang (2005) 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 algorithm unsupervised on the mixed signal of the two instruments to

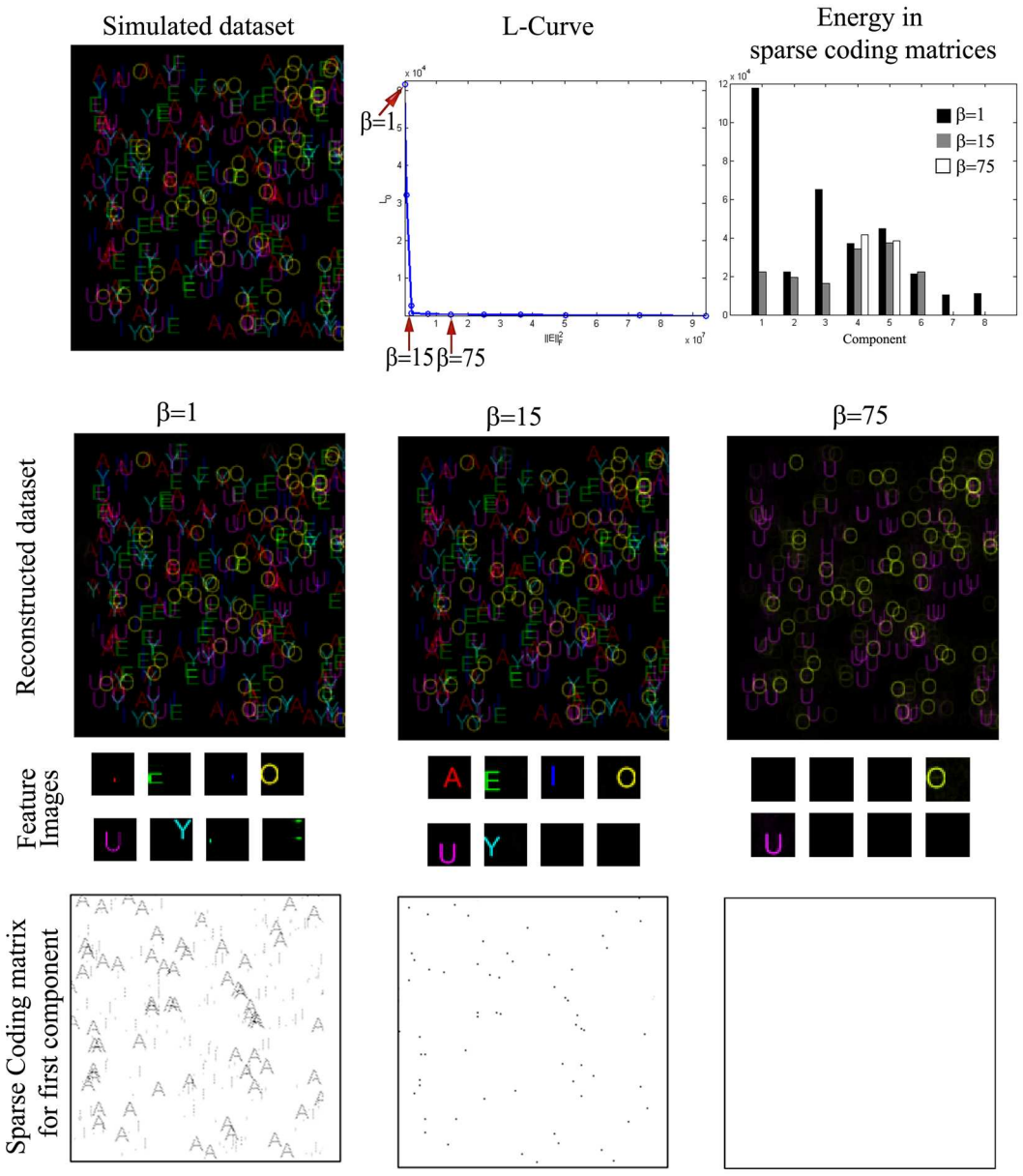

Figure 1: An eight component SISC analysis of an image of colored letters. **Top:** The original image, the L-curve, and the energy in the sparse coding matrix of each feature for three choices of β. **Center:** Result of the analysis for $\beta = \{1, 15, 75\}$. With too low sparsity, $\beta = 1$, the image is perfectly reconstructed, but the features are not found correctly. For example, the "A"-feature is simply a dot, and the "E"-feature corresponds to the upper or lower half of the letter. With properly selected sparsity, $\beta = 15$, the data is perfectly reconstructed and the features correspond to the generating features. With too high sparsity, $\beta = 75$, only two of the letters are captured. **Bottom:** First component of the code corresponding to the letter "A". With too low sparsity, the structure of A is given in the code matrix rather than in the feature. With properly selected sparsity, the code indicates where each "A" is present while the structure of the "A" is captured in the feature. When imposing too much sparsity the code matrix is forced to zero, and the "A" is pruned from the model.

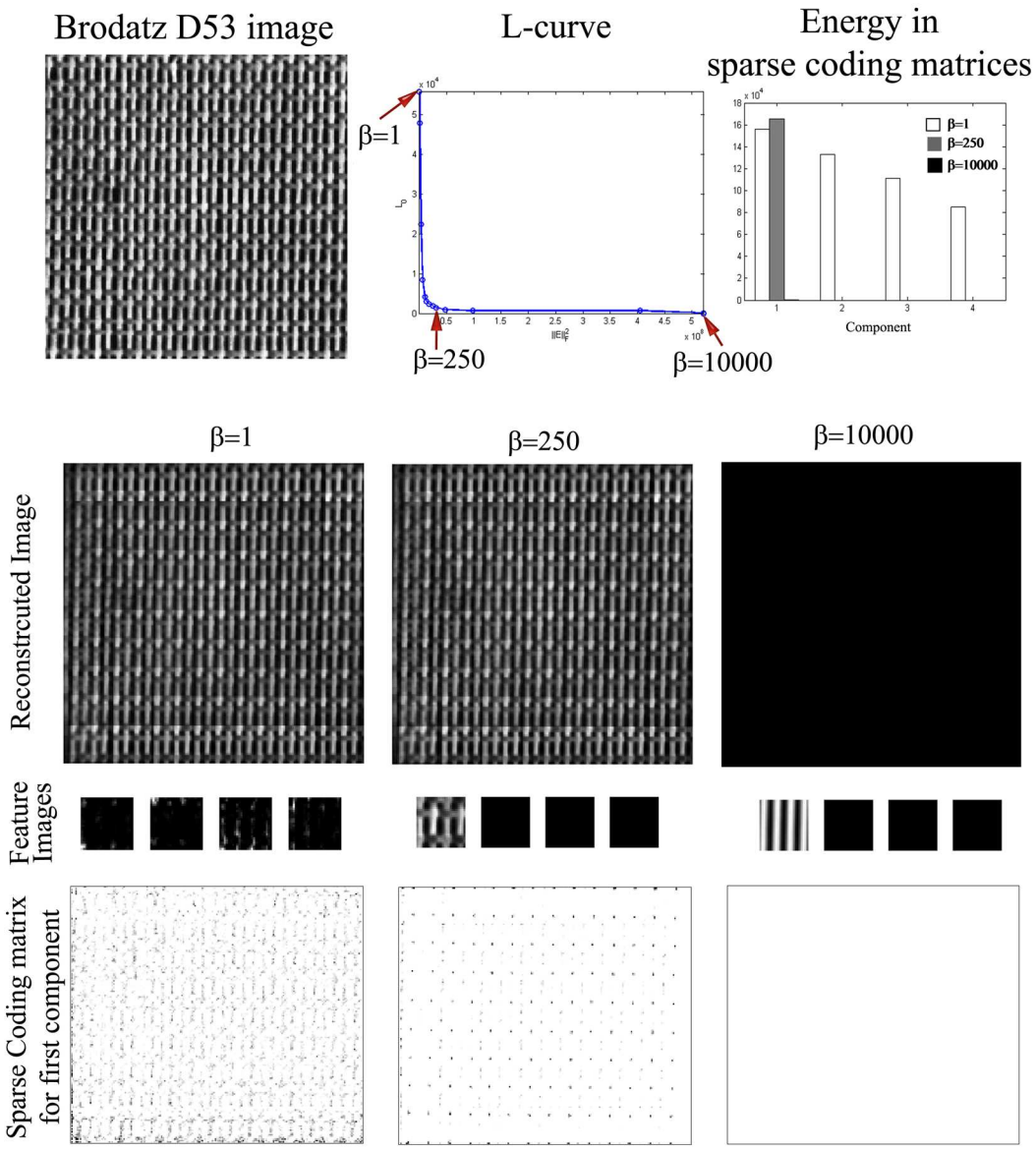

Figure 2: A four component SISC analysis of Brodatz D53, black and white photograph of oriental straw cloth. **Top:** The Brodatz D53 photograph, the L-curve, and the energy of each component in the sparse coding matrix for three choices of β. **Center:** Result of the analysis for $\beta = \{1, 250, 10000\}$. With too low sparsity, $\beta = 1$, the image is perfectly reconstructed, but the features are hard to interpret. With a properly selected sparsity, $\beta = 250$, only one feature image is found. With too high sparsity, $\beta = 10000$, the code is set to zero. **Bottom:** First component of the code. When the sparsity is selected properly, the code is simply a grid of dots.

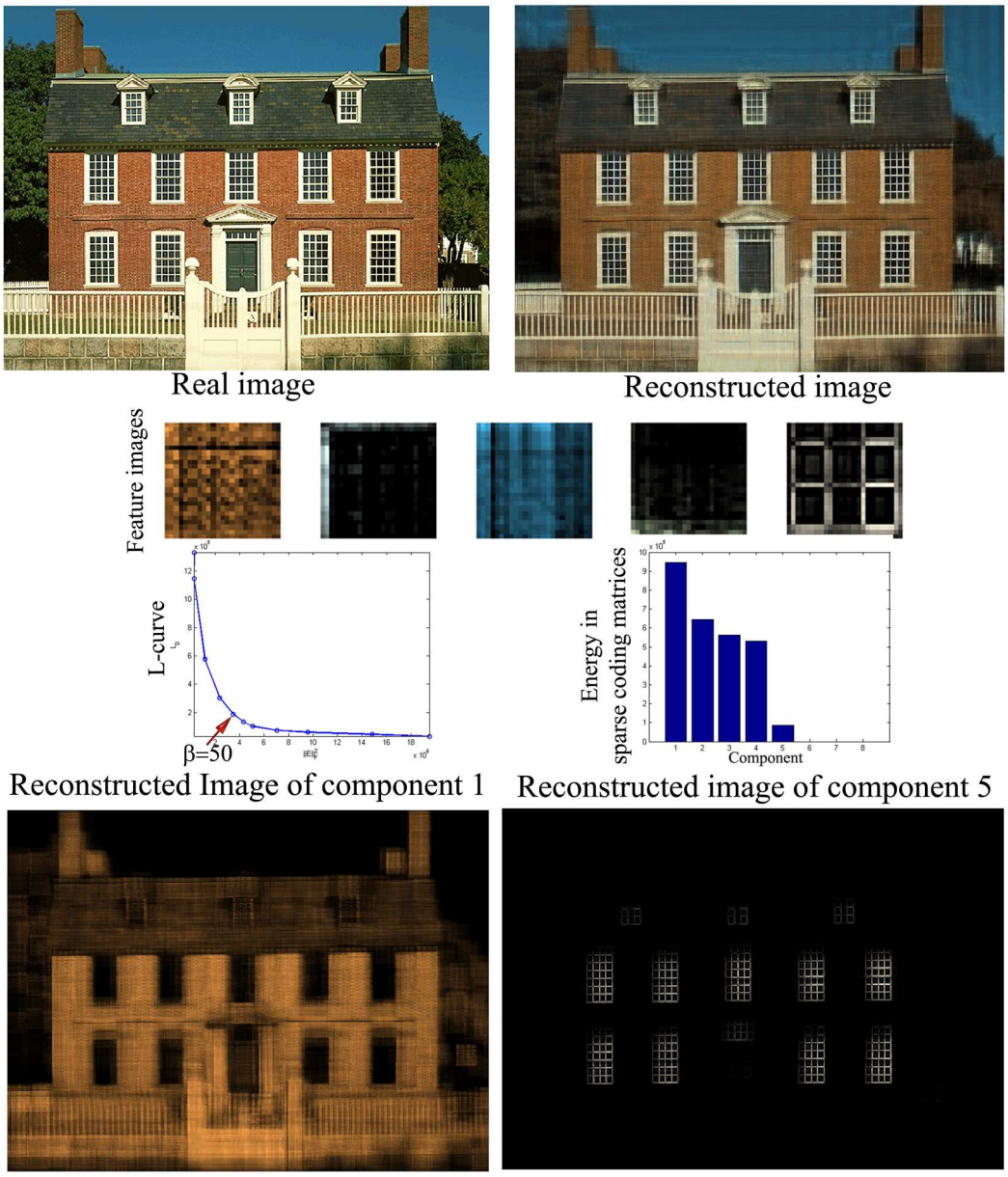

Figure 3: An eight component SISC analysis of a color photograph of a brick house. **Top:** The photograph of the house and the reconstructed image for $\beta = 50$. The model captures well the main features of the original image. **Center:** The analysis results in five components, which mainly correspond to the brick wall, vertical lines in window frames and fence, the sky, horizontal lines, and the window grille. Below is the L-curve and the energy of the sparse coding matrix of each feature. **Bottom:** Example of what each of the components correspond to in the full image. Two of the components are shown; component one mainly captures the brick wall while component five captures the window grille.

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.

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 $M_1 = 373$ and $M_2 = 4$ while $K_1 = 97$ covering 2 octaves, i.e. slightly more than the range of the notes played ($K_2 = 146$). As we were interested in identifying two components, a four component model was fitted. The decomposition found extracted well the two instruments into separate components and turned of the excess components, see Figure 4.

4. Discussion

From the simulated letter data set, see Figure 1, it was seen that the model identified the correct features of the data, namely the six letters and their respective positions in the sparse coding matrices. The model also captured the prominent feature forming the pattern of the oriental straw cloth , see Figure 2, as well as important features of the image of the house as seen in Figure 3. In the analysis of audio data, the model correctly separated the music into features corresponding to each instrument of the music as previously demonstrated in (Schmidt and Mørup, 2006). However, by imposing sparseness we obtained the extra benefit that all the harmonic structure is forced onto $\mathbf{\Psi}$ such that the harmonics of each instruments can be directly read from $\mathbf{\Psi}$ whereas information on what notes are played by the instruments, i.e., the scores, is captured in the sparse coding matrices α_d. A non-sparse model would have confounded the position of the harmonics both in $\mathbf{\Psi}$ and α_d as was the case in (Schmidt and Mørup, 2006), and consequently made the representation less interpretable.

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 (ARD). 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 coding matrices is very useful for the present SISC model. This approach should also be used for other types of L_1 constrained models such as sparse NMF (Eggert and Korner, 2004; Hoyer, 2004) which corresponds to $C = 1$, $K_1 = 1$, $M_2 = 1$ as well as a wider range of sparse models. This is the topic of current work.

In the analysis of the mucic data, the SISC model assumes a constant timbre, i.e., no change in the structure of the harmonics over pitch. Although, this is stated as reasonable in (Zhang and Zhang, 2005) and is valid for the present data set in general the timbre changes considerably over pitch (Nielsen et al., 2007). Thus, in general, each component is likely to work only within limited changes of pitch.

The algorithm we derived was for non-negative decompositions. However, the derived gradients can also be used for unconstrained optimization. Furthermore, the SISC algorithms can be considered an extension of the PARAFAC model to include 2D convolutive mixtures. Consequently, the algorithm devised here gives both a single convolutive mixture, i.e. if $M_1 = I$, $K_2 = J$ and either $M_2 = 1$ or $K_1 = 1$ as proposed by (Smaragdis, 2004; FitzGerald and Coyle, 2006) and a 2D convolutive mixture. Notice, that if both M_2 and K_1 equal one the SISC algorithm becomes an algorithm for sparse non-negative PARAFAC estimation. Furthermore, the developed model can easily be extended to include more modalities and also to incorporate convolutive mixtures in these extra

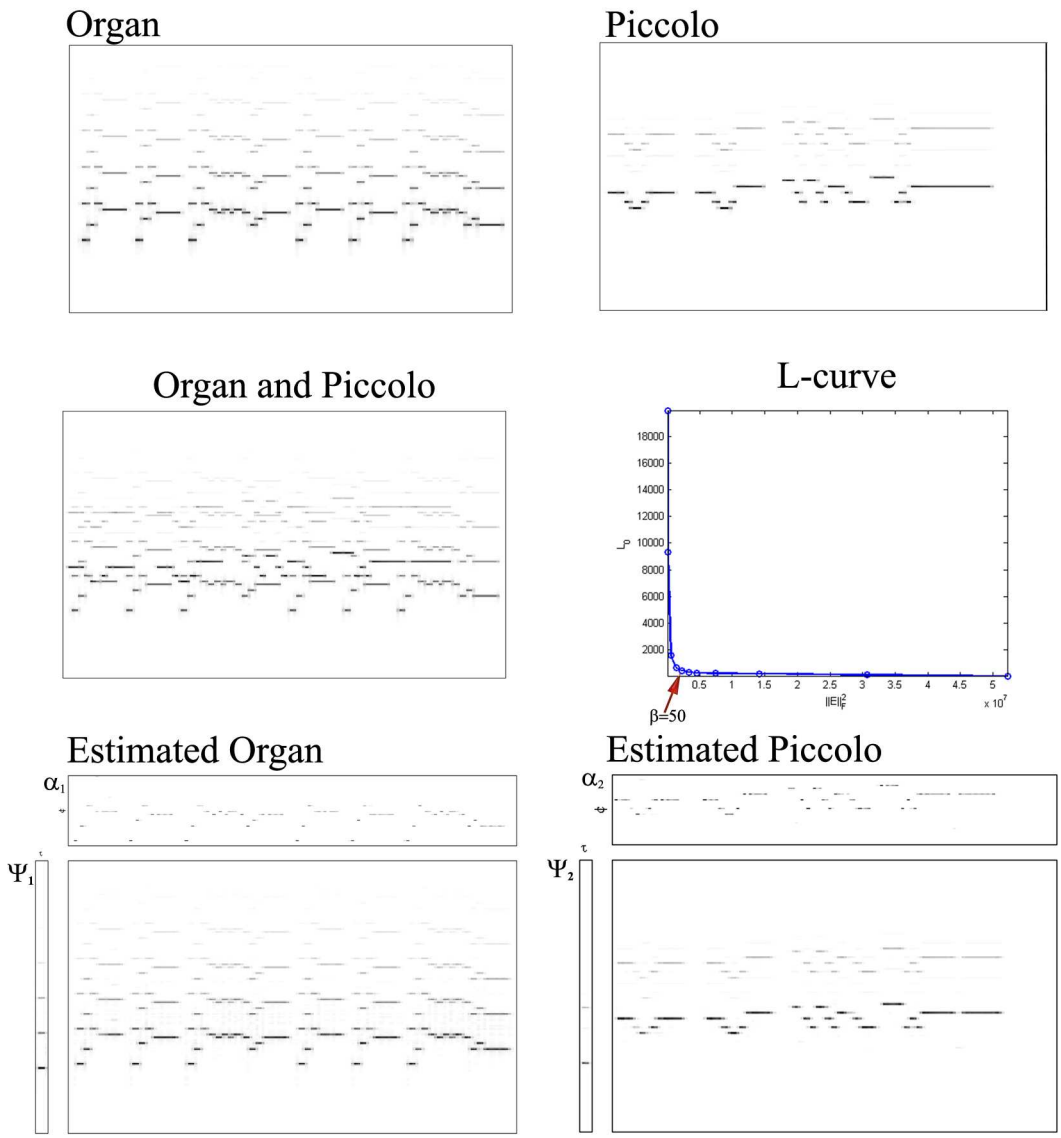

Figure 4: Analysis of the amplitude of the log-spectrogram of a music signal. **Top** The spectrogram of an organ and piccolo respectively. **Center:** spectrogram of the mixed signal of the organ and piccolo. **Bottom:** result obtained when analyzing the mixed spectrogram using a 4-component single 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 $\mathbf{\Psi}_d$ to the left of the reconstructed spectrograms while the scores played is given in α_d shown above the reconstructed spectrogram.

modalities, i.e., a model that is 3-D convolutive, 4-D convolutive etc. Consequently, the framework used here is generalizable to a wide range of higher order data analysis.

Presently, the model was set in the context of image and music analysis. However, the 2D deconvolution represents the data as shift invariant 2-D structures. Consequently, the algorithms devised are more generally useful when data indeed can be represented as such structures. Future work will focus on bridging the proposed model and the results obtained more closely to visual and auditory information processing of the brain. Finally, the SISC model can be expanded to incorporate other types of invariance and constraints than pure shifts. This will also be a focus in future work.

5. conclusion

This paper introduced the Shift Invariant Sparse Coding (SISC) model. The SISC circumvents the need to patch data, handles shift invariance by sparse coding rather than resorting to exhaustive search, is efficiently calculated through the fast Fourier transform (FFT), and generalizes to multi-channel analysis. We demonstrated how the model is useful in estimation of shift invariant features of image and music signals and proposed a method to estimate the optimal degree of sparseness based on the L-curve approach. The algorithm can be downloaded from (Mørup and Schmidt, 2007).

Appendix A. Derivation of the algorithms

In the following derivation of the algorithm for SISC the derivative of a given element of L_c with respect to a given element of $\alpha_d(u,v)$ and $\mathbf{\Psi}_d(x,y)$ and $s_{c,d}$ is needed:

$$\begin{aligned}
\frac{\partial L_c(x,y)}{\partial \mathbf{\Psi}_{d'}(x',y')} &= \frac{\partial \sum_d s_{c,d} \sum_{u,v} \alpha_d(u,v)\mathbf{\Psi}_d(x-u,y-v)}{\partial \mathbf{\Psi}_{d'}(x',y')} = \mathbf{s}_{c,d'}\alpha_{d'}(x-x',y-y'),\\
\frac{\partial L_c(x,y)}{\partial \alpha_{d'}(u',v')} &= \frac{\partial \sum_d s_{c,d} \sum_{u,v} \alpha_d(u,v)\mathbf{\Psi}_d(x-u,y-v)}{\partial \alpha_{d'}(u',v')} = s_{c,d'}\mathbf{\Psi}_{d'}(x-u',y-v'),\\
\frac{\partial L_c(x,y)}{\partial s_{c',d'}} &= \frac{\partial \sum_d s_{c,d} \sum_{u,v} \alpha_d(u,v)\mathbf{\Psi}_d(x-u,y-v)}{\partial s_{c',d'}} = \sum_{u,v} \alpha_{d'}(u,v)\mathbf{\Psi}_{d'}(x-u,y-v).
\end{aligned}$$

Furthermore, the derivatives of $\widetilde{s}_{c,d}$ and $\widetilde{\mathbf{\Psi}}(x,y)$ is needed for the normalization when imposing sparseness on α:

$$\begin{aligned}
\frac{\partial \widetilde{s}_{c',d'}}{\partial s_{c',d'}} &= \frac{\partial \frac{s_{c',d'}}{\|\mathbf{S}\|_F}}{\partial s_{c',d'}} = \frac{1}{\|\mathbf{S}\|_F} - s_{c',d'} \sum_{c,d} \frac{s_{c,d}}{\|\mathbf{S}\|_F^3},\\
\frac{\partial \widetilde{\mathbf{\Psi}}_{d'}(x',y')}{\partial \mathbf{\Psi}_{d'}(x',y')} &= \frac{\partial \frac{\mathbf{\Psi}_{d'}(x',y')}{\|\mathbf{\Psi}_{d'}\|_F}}{\partial \mathbf{\Psi}_{d'}(x',y')} = \frac{1}{\|\mathbf{\Psi}_{d'}\|_F} - \mathbf{\Psi}_{d'}(x',y') \sum_{x,y} \frac{\mathbf{\Psi}_{d'}(x,y)}{\|\mathbf{\Psi}_{d'}\|_F^3}
\end{aligned}$$

Where $\|\mathbf{S}\|_F = \sqrt{\sum_{c,d} s_{c,d}^2}$ and $\|\mathbf{\Psi}_d\|_F = \sqrt{\sum_{x,y} \mathbf{\Psi}_d(x,y)^2}$. The gradient of the cost functions can be derived by differentiation by parts, for instance we find when differentiating the least squares cost function with respect to $\mathbf{\Psi}_{d'}(x',y')$

$$\frac{\partial C_{LS}}{\partial \mathbf{\Psi}_{d'}(x',y')} = -\sum_{x,y,c} (X_c(x,y) - L_c(x,y)) \frac{\partial L_c(x,y)}{\partial \widetilde{\mathbf{\Psi}}_{d'}(x',y')} \frac{\partial \widetilde{\mathbf{\Psi}}_{d'}(x',y')}{\partial \mathbf{\Psi}_{d'}(x',y')}. \tag{10}$$

The updates are finally found using the approach of multiplicative updates, i.e. splitting the gradient into positive and negative terms and setting positive terms in the denominator and negative in the nominator, see section 2.2.

Appendix B. Convergence

In the following the convergence of the algorithm for $\gamma = 1$ without normalization of $\mathbf{\Psi}_d$ and $\mathbf{S}$ will be given, thus for the algorithm without sparseness. Although no proof is given for the convergence including normalization we never experienced divergence of the algorithm proposed for $\gamma = 1$. Had the algorithm diverged the step size parameter in the multiplicative update, γ, could have been tuned such that the algorithm would keep decreasing the cost-function, see also section 2.2.

The proof is based on the use of an auxiliary function and follows closely the proof for the convergence of the regular NMF algorithm by Lee and Seung (2000). Briefly stated, an auxiliary function G to the function F is defined by: $G(\alpha, \alpha^t) \geq F(\alpha)$ and $G(\alpha, \alpha) = F(\alpha)$. If G is an auxiliary function then F is non-increasing under the update $\alpha = \arg\min_\alpha G(\alpha, \alpha^t)$.

Essentially following the proof of the least squares NMF updates of Lee and Seung (2000), we start by defining:

$$F(\alpha) = \frac{1}{2} \sum_{x,y,c} \left(X_c(x,y) - L_c(x,y)\right)^2$$

Notice that F is just the regular least square cost function C_{LS}. Define the vector α_a as $\alpha_a = \alpha_d(u,v)$. This vector is simply a vectorization of α where a indexes all combinations of d, u and v. The gradient vector ∇F_a and Hessian matrix $\mathbf{Q}_{a,b}$ found by differentiating F with respect to the element $\alpha_d(u,v)$ and $\alpha_{d'}(u',v')$ denoted by a and b, gives:

$$\begin{aligned} \nabla F_a = \frac{\partial C_{LS}}{\partial \alpha_{d'}(u',v')} &= -\sum_{x,y,c} \left(X_c(x,y) - L_c(x,y)\right) s_{c,d'} \mathbf{\Psi}_{d'}(x-u', y-v') \\ \mathbf{Q}_{a,b} = \frac{\partial F(\alpha)^2}{\partial \alpha_{d'}(u',v') \partial \alpha_d(u,v)} &= \sum_{x,y,c} s_{c,d} \mathbf{\Psi}_d(x-u, y-u) \mathbf{\Psi}_{d'}(x-u', y-v') s_{c,d'} \end{aligned}$$

Since $F(\alpha)$ is a quadratic function it is completely described by a second order Taylor expansion here expressed in terms of α as:

$$F(\alpha) = F(\alpha^t) + (\alpha - \alpha^t)^{\mathrm{T}} \nabla F(\alpha^t) + \frac{1}{2}(\alpha - \alpha^t)^{\mathrm{T}} \mathbf{Q} (\alpha - \alpha^t)$$

Now let $K(\alpha^t)$ be a diagonal matrix defined by

$$K(\alpha^t)_{ab} = \delta_{ab} (\mathbf{Q}\alpha^t)_a / (\alpha^t)_a.$$

Further, define the auxiliary function

$$G(\alpha, \alpha^t) = F(\alpha^t) + (\alpha - \alpha^t) \nabla F(\alpha^t) + \frac{1}{2}(\alpha - \alpha^t)^{\mathrm{T}} K(\alpha^t)(\alpha - \alpha^t).$$

Clearly $G(\alpha, \alpha) = F(\alpha)$. Finding $G(\alpha, \alpha^t) \geq F(\alpha^t)$ corresponds to

$$(\alpha - \alpha^t)^{\mathrm{T}} (K(\alpha^t) - \mathbf{Q})(\alpha - \alpha^t) \geq 0$$

This requires the matrix $(K(\alpha^t) - \mathbf{Q})$ to be positive semidefinite (Lee and Seung, 2000).

The rest of the proof follows closely the convergence proof of the regular NMF (Lee and Seung, 2000). Define the matrix $\mathbf{M}_{a,b}(\alpha^t) = \alpha_a^t (K(\alpha^t) - \mathbf{Q})_{a,b} \alpha_b^t$. This is just a re-scaling of the elements

in $(K(\alpha^t)-\mathbf{Q})$. Then $(K(\alpha^t)-\mathbf{Q})$ is semi-positive definite if and only if $\mathbf{M}$ is

$$\begin{aligned}
\nu^t\mathbf{M}\nu &= \sum_{ab}\nu_a^t\mathbf{M}_{a,b}\nu_b \\
&= \sum_{ab}\nu_a^t(\alpha_a^t(\delta_{ab}(\mathbf{Q}\alpha^t)_a/(\alpha^t)_a-\mathbf{Q})_{a,b}\alpha_b^t)\nu_b \\
&= \sum_{ab}\alpha_a^t\mathbf{Q}_{a,b}\alpha_b^t\nu_a^2-\nu_a\alpha_a^t\mathbf{Q}\alpha_b^t\nu_b \\
&= \sum_{ab}\mathbf{Q}_{a,b}\alpha_a^t\alpha_b^t(\frac{1}{2}\nu_a^2+\frac{1}{2}\nu_b^t-\nu_a\nu_b) \\
&= \frac{1}{2}\sum_{ab}\mathbf{Q}_{a,b}\alpha_a^t\alpha_b^t(\nu_a-\nu_b)^2\geq 0
\end{aligned}$$

all that is left to prove is that minimizing G yield the least square updates

$$\frac{\partial G(\alpha,\alpha^t)}{\partial\alpha}=0\Leftrightarrow\alpha=\alpha^t-K(\alpha^t)^{-1}\nabla F(\alpha^t)\Leftrightarrow\alpha_a=\alpha_a^t-\frac{(\alpha^t)_a}{(\mathbf{Q}\alpha^t)_a}\nabla F(\alpha^t)_a. \tag{11}$$

Changing the indexing a to be of the parameters d, u, and v, we get

$$(\mathbf{Q}\alpha^t)_a=\sum_{x,y,c}s_{c,d}\mathbf{\Psi}_d(x-u,y-v)\sum_{u',v',d'}\mathbf{\Psi}_{d'}(x-u',y-v')s_{c,d'}\alpha_{d'}^t(u',v')=\sum_{x,y,c}s_{c,d}\mathbf{\Psi}_d(x-u,y-v)L_c^t(x,y).$$

Where $L_c^t(x,y)=\sum_d s_{c,d}\sum_{u,v}\alpha_d^t(u,v)\mathbf{\Psi}_d(x-u,y-v)$. Consequently

$$\begin{aligned}
\alpha_d(u,v) &= \alpha_d^t(u,v)+\frac{\alpha_d^t(u,v)\sum_{x,y,c}s_{c,d}\mathbf{\Psi}_d(x-u,y-v)(X_c(x,y)-L_c^t(x,y))}{\sum_{x,y,c}s_{c,d}\mathbf{\Psi}_d(x-u,y-v)L_c^t(x,y)} \\
&= \alpha_d^t(u,v)\frac{\sum_{x,y,c}s_{c,d}\mathbf{\Psi}_d(x-u,y-v)X_c(x,y)}{\sum_{x,y,c}s_{c,d}\mathbf{\Psi}_d(x-u,y-v)L_c^t(x,y)},
\end{aligned}$$

which concludes the proof. When imposing sparseness the convergence of the α update is easily proven for L_1 defining $K(\alpha^t)_{a,b}=\delta_{a,b}\frac{(\mathbf{Q}\alpha^t)_a+\beta}{\alpha_a^t}$ as proposed by Hoyer (2002) for regular NMF in the above. The convergence of the $\mathbf{\Psi}$ update can be similarly derived interchanging the roles of $\mathbf{\Psi}$ and α in the above. The convergence of the $\mathbf{S}$ update follows by restating the problem as regular NMF by vectorizing the images indexed by pixel row and column x and y into the new index $q(x,y)$, i.e. $X_{c,q(x,y)}=\sum_d s_{c,d}\mathbf{Z}_{q(x,y),d}$ where $\mathbf{Z}_{q(x,y),d}=\sum_{u,v}\mathbf{\Psi}_d(x-u,y-v)\alpha_d(u,v)$.

References

H. Asari, B. A. Pearlmutter, and A. M. Zador. Sparse representations for the cocktail party problem. *Journal of Neuroscience*, 26(28):7477–7490, 2006.

P. Brodatz. *Textures: A Photographic Album for Artists and Designers. (Brodatz mosaic D53).* Dover Publications, 1966.

J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35:283–319, 1970.

A Cichocki, R Zdunek, and 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*, pages 32–39, 2006.

I. S. Dhillon and S. Sra. Generalized nonnegative matrix approximations with bregman divergences. *NIPS*, pages 283–290, 2005.

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):797–829, 2006.

D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? *NIPS*, 2003.

J. Eggert and E. Korner. Sparse coding and nmf. In *Neural Networks*, volume 4, pages 2529–2533, 2004.

J. Eggert, H. Wersing, and E. Korner. Transformation-invariant representation and nmf. In *Neural Networks*, volume 4, pages 2535– 2539, 2004.

D. FitzGerald and E. Coyle. Sound source separation using shifted non-negative tensor factorisation. In *ICASSP2006*, 2006.

D. FitzGerald, M. Cranitch, and E. Coyle. Non-negative tensor factorisation for sound source separation. In *proceedings of Irish Signals and Systems Conference*, pages 8–12, 2005.

G.H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

L. K. Hansen, K. H. Madsen, and T. Lehn-Schiøler. Adaptive regularization of noisy linear inverse problems. In *Eusipco*, 2006. URL `http://www2.imm.dtu.dk/pubdb/p.php?4417`.

P. C. Hansen. Analysis of discrete ill-posed problems by means of the l-curve. *SIAM Review*, 34(4): 561–580, 1992.

R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

W. Hashimoto and K. Kurata. Properties of basis functions generated by shift invariant sparse representations of natural images. *Biol. Cybern.*, 83:111–118, 2000.

P. O. Hoyer and A. Hyvärinen. Independent component analysis applied to feature extraction colour and stereo images. *Network: Computation in Neural Systems*, 11(3):191–210, 2000.

P.O. Hoyer. Non-negative sparse coding. *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 557–565, 2002.

P.O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 2004.

A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley and Sons., 2001.

A. Hyvärinen and P. O. Hoyer. A two-layer coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, 21(18):2413–2423, 2001.

A. Hyvärinen and E. Oja. Independent component anlayhsis: Algorithms and application. *Neural Networks*, 13:411–430, 2000.

C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, 1974.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–91, 1999.

D.D. Lee, H.S. Seung, and L.K. Saul. Multiplicative updates for unsupervised and contrastive learning in vision. *Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies. KES 2002*, 1:387–91, 2002.

T.-W. Lee and M. S. Lewicki. Unsupervised image classification, segmentation, and enhancement using ica mixture models. *IEEE tansactions on Image Processing*, 11(3):270–279, 2002.

C.-J. Lin. Projected gradient methods for non-negative matrix factorization. *To appear in Neural Computation*, 2007.

M. Mørup and M. N. Schmidt. www2.imm.dtu.dk/pubdb/views/edoc_download.php/4652/zip/imm4652.zip., 2007.

A. B. Nielsen, S. Sigurdsson, L. K. Hansen, and J. Arenas-Garcia. On the relevance of spectral features for instrument classification. *ICASSP*, pages 485–488, 2007.

Bť. A. Olshausen. Learning sparse, overcomplete representations of time-varying natural images. *Image Processing, ICIP 2003. Proceedings. 2003 International Conference*, 1:41–44, 2003.

B. A. Olshausen and David J. Field. Sparse coding of sensorty inputs. *Current Opinion in Neurobiology*, 14:481–487, 2004.

B. A. Olshausen and J. Field, Dacvid. Sparse coding with an overcomplete basis set: A strategy employed by v1. *Vision Research*, 37(23):3311–3325, 1997.

F. D.J. Olshausen, B. A. Emergence of simple-cell receptiove field propertises by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

R. Salakhutdinov, S. Roweis, and Z. Ghahramani. On the convergence of bound optimization algorithms. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 509–516, San Francisco, CA, 2003. Morgan Kaufmann Publishers.

M. N. Schmidt and M. Mørup. Nonnegative matrix factor 2-D deconvolution for blind single channel source separation. In *ICA2006*, pages 700–707, 2006.

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)*, 3195:494, sep 2004.

P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 177–180, October 2003.

K. Tanaka. Representation of visual features of objects in the inferotemporal cortex. *Neural Networks*, 9(8):1459–1475, 1996.

S. E. Trehub. The development origins of musicality. *Nature Neuroscience*, 6(7):669–673, 2003.

B. Wang and M. D. Plumbley. Musical audio stream separation by non-negative matrix factorization. In *Proceedings of the DMRN Summer Conference*, july 2005.

H. Wersing, J. Eggert, and E. Körner. Sparse coding with invariance constraints. *Proc. Int. Conf. Artičcial Neural Networks ICANN*, pages 385–392, 2003.

Y.-G. Zhang and C.-S. Zhang. Separtation of music signals by harmonic structure modeling. *Proceedings of Neural Information Processing Systems (NIPS)*, pages 184–191, 2005.