Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition Letters 26 (2005) 2491–2499

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Blind source separation of more sources than mixtures using sparse mixture models

Zhenwei Shi [a,*], Huanwen Tang [b], Yiyuan Tang [c,d,e]

[a] State Key Laboratory of Intelligent Technology and Systems, Department of Automation, Tsinghua University, Beijing 100084, PR China
[b] Institute of Computational Biology and Bioinformatics, Dalian University of Technology, Dalian 116023, PR China
[c] Institute of Neuroinformatics, Dalian University of Technology, Dalian 116023, PR China
[d] Laboratory of Visual Information Processing, The Chinese Academy of Sciences, Beijing 100101, PR China
[e] Key Lab for Mental Health, The Chinese Academy of Sciences, Beijing 100101, PR China

Received 21 December 2003; received in revised form 24 September 2004
Available online 12 July 2005

Communicated by E. Backer

Abstract

In this paper, blind source separation is discussed with more sources than mixtures. This blind separation technique assumes a linear mixing model and involves two steps: (1) learning the mixing matrix for the observed data using the sparse mixture model and (2) inferring the sources by solving a linear programming problem after the mixing matrix is estimated. Through the experiments of the speech signals, we demonstrate the efficacy of this proposed approach.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Blind source separation; Overcomplete representation; Sparse mixture model; Independent component analysis; Signal processing

1. Introduction

Independent component analysis (ICA) (Hyvärinen et al., 2001) is a technique that has received a great deal of attention due to various applications in blind source separation, blind deconvolution, feature extraction, and so on. The goal of ICA is to recover independent sources given only sensor observations that are unknown linear mixtures of the unobserved independent source signals (Amari et al., 1996; Bell and Sejnowski, 1995; Cardoso and Laheld, 1996; Comon, 1994; Hyvärinen, 1999; Lee et al., 1999a; Murata et al., 2001, 2002; Shi et al., 2004a).

[*] Corresponding author.
E-mail addresses: szw1977@yahoo.com.cn (Z. Shi), yy2100@163.net (Y. Tang).

0167-8655/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2005.05.006

The standard formulation of ICA requires at least as many sensors as sources. Several researchers proposed various methods for the noise model and the noise free model which generalized the standard ICA. For the noise model, Lewicki and Olshausen (1999) and Lewicki and Sejnowski (2000) derived a gradient-based method for learning overcomplete representations of the data that allowed for more basis vectors than dimensions in the inputs where there was a requirement for the assumption of a low level of noise. Lee et al. (1999b) demonstrated that three speech signals could be separated given only two mixtures of the three signals using overcomplete representations. An expectation-maximization (EM) algorithm for learning sparse and overcomplete data representations was presented by Girolami (2001). The proposed algorithm exploited a variational approximation to a range of heavy-tailed distributions whose limit was the Laplacian. Based on the EM algorithm, Zhong et al. (2004) presented a method for inferring the most probable basis coefficients and learning the overcomplete basis vectors. The conditional moments of the intractable posterior distribution were estimated by maximum a posteriori (MAP) estimation. Zibulevsky and Pearlmutter (2001) and Zibulevsky et al. (2001) suggested that the mixing matrix and the sources were estimated by using maximum a posteriori approach. The blind separation technique presented by Shi et al. (2004b) included two steps. The first step was to estimate the mixing matrix, and the second was to estimate the sources. If the sources were sparse, the mixing matrix could be estimated by using the generalized exponential mixture model. After estimating the mixing matrix, the sources could be obtained by using maximum a posteriori approach. For the noise free model, the blind separation technique proposed by Li et al. (2003) included two steps. The first step was to estimate the mixing matrix, the second was to estimate the sources. The mixing matrix was estimated using K-means clustering algorithm.

In this paper, we consider the noise free model. Motivated by these methods, we present a gradient learning algorithm for the sparse mixture model that is able to estimate the mixing matrix. After the mixing matrix is estimated, the sources are estimated by using a linear programming algorithm. Experiments with speech signals demonstrate good separation results.

2. Overcomplete representation and sparse mixture model

In blind source separation a sensor signal $x = (x_1, \ldots, x_M)^T \in R^M$ can be described using an overcomplete basis by the following noise free linear model:

$$x = As, \tag{1}$$

where the columns of the mixing matrix $A \in R^{M\times L}$ ($L > M$) define the overcomplete basis vectors, $s = (s_1, \ldots, s_L)^T \in R^L$ is the source signal (or the representation of the sensor signal). The elements of s are assumed mutually statistical independent. This means that the joint probability distribution of s is factorable, i.e., $p(s) = \prod_{l=1}^{L} p(s_l)$, where p represents the probability density function (p.d.f.). In addition, each prior $p(s_l)$ is assumed to be sparse typified by the Laplacian distribution (Lewicki and Sejnowski, 2000). Sparsity means that only a small number of the s_l differ significantly from zero. We aim to estimate the mixing matrix A and the source signal s given only the observed data x.

For a given mixing matrix A, the source signal can be found by maximizing the posterior distribution $p(s|x,A)$ (Lewicki and Sejnowski, 2000). This can be solved by a standard linear program when the prior is Laplacian (Chen et al., 1996; Lewicki and Sejnowski, 2000; Li et al., 2003). Thus, we can estimate the mixing matrix A first.

The phenomenon of data concentration along the directions of the mixing matrix columns can be used in clustering approaches to source separation (Zibulevsky et al., 2001). In a two-dimensional space, the observations x were generated by a linear mixture of three independent sparse sources (the same three sources of nature speech signals and mixing matrix as used in (Lee et al., 1999b)), as shown in Fig. 1 (Left) (scatter plot of two mixtures x_1 versus x_2). The three distinguished directions, which correspond to the columns of the mixing matrix A, are visible. In order to determine

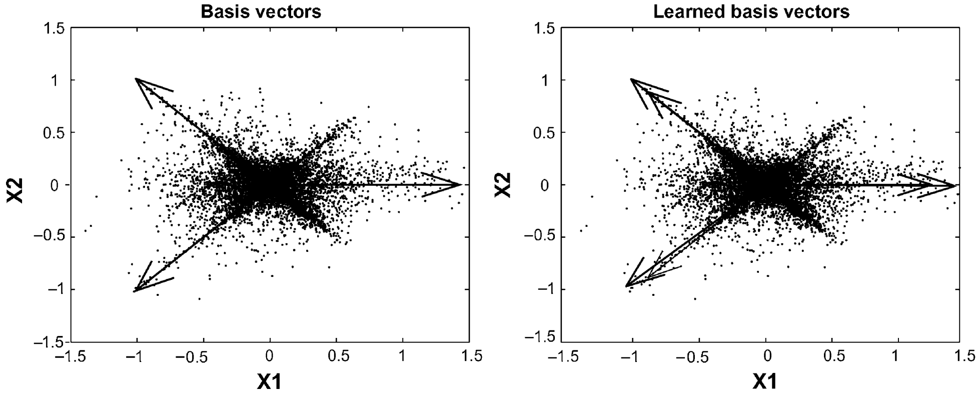

Fig. 1. Basis vectors (the columns of the mixing matrix): Left: In a two-dimensional space, the observations x are generated by a linear mixture of three independent sparse sources (scatter plot of two mixtures x_1 versus x_2). Right: After the learning algorithm converges, the learned basis vectors (the long arrows) are close to the true basis vectors (the short arrows).

orientations of data concentration, we project the data points onto the surface of a unit sphere by normalizing the sensor data vectors at every particular time index t: $x_t = x_t/\|x_t\|(x_t = (x_1(t),x_2(t))^T$, $t = 1,\ldots,T)$. Next, the data points were moved to a half-sphere, e.g., by forcing the sign of the first coordinate $x_1(t)$ to be positive (without this operation each 'line' of data concentration would yield two clusters on opposite sides of the sphere). For each point x_t, the data point $\alpha_t = \sin^{-1}(x_2(t))$ was computed by the second coordinate $x_2(t)$. This is a 1–1 mapping from Cartesian coordinates to polar coordinates, because the data vectors are normalized. Thus, the data $\alpha = \{\alpha_1,\ldots,\alpha_T\}$ also have the centers of the three clusters corresponding to the three distinguished directions for two mixtures. The histogram of the data α is presented in Fig. 2 (Left). The coordinates of the centers of the three clusters determine the columns of the estimated mixing matrix A.

We can see that the density function of the data α is formed from a linear combination of sparse functions. We therefore write this model for density as a linear combination of component densities $p(\alpha|k)$ (i.e., the kth cluster density) in the form

$$p(\alpha) = \sum_{k=1}^{K} p(\alpha|k)p(k), \tag{2}$$

where the coefficients $p(k)$ are called the mixing parameters. Such a model is called a mixture model. When the component densities $p(\alpha|k)$ are modelled as Gaussian, it is called a Gaussian mixture model. Here we consider that the component densities $p(\alpha|k)$ are modelled as sparse densities, and we call it the sparse mixture model. The sparse density typified by the Laplacian distribution is

$$p(\alpha|k) \propto \beta_k \exp(-\beta_k|\alpha - b_k|), \tag{3}$$

where b_k, β_k are the parameters for the Laplacian distribution. Fig. 2 (Right) shows a linear combination of three sparse densities typified by the Laplacian distributions. The sparse mixture distribution makes a good representation for the density function generating the data α.

Thus, we should determine cluster centers of the sparse mixture distribution using a specific algorithm (i.e. estimate the cluster centers b_k). Their coordinates will determine the columns of the estimated mixing matrix A.

3. A gradient learning algorithm for sparse mixture model

We consider the n-dimensional sparse mixture model in this section. Assume that the data $\alpha = \{\alpha_1,\ldots,\alpha_T\}$ are drawn independently and generated by a sparse mixture model. The likelihood of the data is given by the joint density

$$p(\alpha|\Theta) = \prod_{t=1}^{T} p(\alpha_t|\Theta).$$

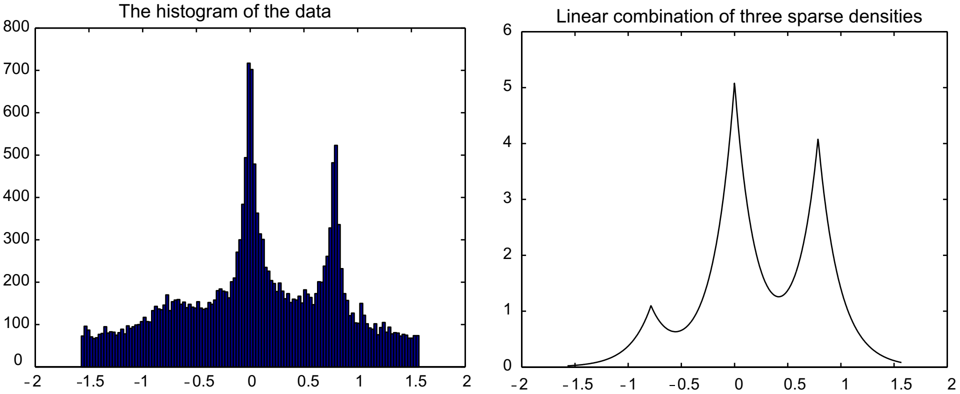

Fig. 2. Three sources and two mixtures: Left: The histogram of the data α. Right: A linear combination of three sparse densities typified by the Laplacian distributions. It makes a good representation for the density function generating the data α.

The mixture density is

$$p(\alpha_t|\Theta) = \sum_{k=1}^{K} p(\alpha_t|\theta_k, k)p(k),$$

where $\Theta = (\theta_1, \ldots, \theta_K)$ are the unknown parameters for each $p(\alpha_t|\theta_k,k)$. We assume that the component densities $p(\alpha_t|\theta_k,k)$ are modelled as sparse densities typified by the n-dimensional Laplacian distributions, i.e.

$$p(\alpha_t|\theta_k, k) \propto \beta_k^n \exp\left(-\beta_k \sum_{i=1}^{n} |\alpha_{t_i} - b_{k_i}|\right),$$

where $\theta_k = \{\beta_k, b_k\}$ are the parameters for the Laplacian distribution and $\beta_k \in R$, $b_k = (b_{k_1}, \ldots, b_{k_n})^T \in R^n$, $\alpha_t = (\alpha_{t_1}, \ldots, \alpha_{t_n})^T \in R^n$. Our goal is to infer the cluster centers b_k from the data $\alpha = \{\alpha_1, \ldots, \alpha_T\}$. Then, the coordinates of the centers of the clusters determine the columns of the estimated mixing matrix A. We derive an iterative learning algorithm which performs gradient ascent on the total likelihood of the data as follows (see Appendix A):

$$\Delta b_k \propto p(k|\alpha_t, \Theta)(\beta_k \tanh(\gamma(\alpha_t - b_k))),$$

$$\Delta\beta_k \propto p(k|\alpha_t, \Theta)\left(\frac{n}{\beta_k} - \sum_{i=1}^{n} |\alpha_{t_i} - b_{k_i}|\right),$$

where $\tanh(\gamma(\alpha_t - b_k)) = (\tanh(\gamma(\alpha_{t_1} - b_{k_1})), \ldots, \tanh(\gamma(\alpha_{t_n} - b_{k_n})))^T$, γ is a large positive constant.

Thus we obtain the learning algorithm as follows:

(i) Normalize the data vectors at every particular time index t: $x_t = x_t/\|x_t\|$.
(ii) The data points are moved to a half-sphere.
(iii) For each point x_t, the data point α_t is obtained by computing the polar coordinates of the data point x_t (i.e. from n-dimensional Cartesian coordinates to $n-1$-dimensional polar coordinates, because the data vectors are normalized).
(iv) The learning algorithm for the sparse mixture model is used for estimating the mixing matrix (using the $n-1$-dimensional sparse mixture model, i.e. estimating the cluster centers b_k for α, and the coordinates of b_k determine the columns of the estimated mixing matrix by transforming $n-1$-dimensional polar coordinates of b_k to n-dimensional Cartesian coordinates).
(v) After the mixing matrix is estimated, the linear programming algorithm (Lewicki and Sejnowski, 2000; Li et al., 2003) is performed for obtaining the sources.

4. Simulation examples

We considered separating three speech sources from two mixtures. The observations x were generated by a linear mixture of the three speech signals used in Section 2. Then we used the same method in Section 2 to compute the data $\alpha = \{\alpha_1, \ldots, \alpha_T\}$. The learning algorithm for the sparse mixture model in Appendix A was used for estimating the mixing matrix (using the one-dimensional sparse mixture model, i.e. estimating the three cluster centers b_k for α, $K = 3$ here). The parameters were randomly initialized and the learning rates were set to be 0.0005 (typically 40–60 iterations). Fig. 1 (Right) shows the learned basis vectors (the long arrows) and the true basis vectors (the short arrows). After the mixing matrix was estimated, we performed the linear programming algorithm for obtaining the sources. Three original signals, two mixtures, and three separated output signals are shown in Fig. 3. From a subjective listening point of view, the separation of the three nature speech example was remarkable for the high intelligibility of the recovered sentences, in spite of some background noise and crosstalk. In order to measure the accuracy of separation, we normalized the original sources with $\|s_j\|_2 = 1$, $j = 1, 2, 3$, and the estimated sources with $\|\tilde{s}_j\|_2 = 1, j = 1, 2, 3$. The error was computed as

$$\text{Error} = \|\tilde{s}_j - s_j\|_2.$$

For the noise free linear model, generally, if the mixing matrix was known, the sources were estimated by the linear programming algorithm (Zibulevsky et al., 2001; Li et al., 2003). If the mixing matrix was accuracy, we performed the linear programming algorithm for obtaining the estimated sources to aid comparison. The error of the three estimated sources was 0.5402, 0.3780 and 0.3240, respectively. And the error of the three estimated sources computed by our algorithm was 0.5407, 0.3788 and 0.3241, correspondingly.

Next, we considered the problem of separating five speech sources from three mixtures. The five speech signals (available at http://people.ac.upc.es/pau/shpica/instant.html and http://www.cnl.salk.edu/~tewon/Over) were mixed by a 3 × 5 matrix, such as

$$A = \begin{pmatrix} 0.4755 & -0.2939 & 0.7694 & -0.2939 & 0.4755 \\ 0.3455 & 0.9045 & 0.5590 & -0.9045 & -0.3455 \\ 0.8090 & 0.3090 & -0.3090 & 0.3090 & -0.8090 \end{pmatrix}.$$

The sensor data vectors were normalized at every particular time index t: $x_t = x_t/\|x_t\|$, ($x_t = (x_1(t), x_2(t), x_3(t))^T$, $t = 1, \ldots, T$). Next, the data points were moved to a half-sphere: IF $x_3(t) < 0$, THEN $x_t = -x_t$. For each point x_t, the data $\alpha_t = (\alpha_1(t), \alpha_2(t))^T$ was computed from:

$$x_3(t) = \sin(\alpha_1(t)), x_2(t) = \cos(\alpha_1(t))\sin(\alpha_2(t)),$$

$$x_1(t) = \cos(\alpha_1(t))\cos(\alpha_2(t)).$$

The learning algorithm for the sparse mixture model was used for estimating the mixing matrix (using the two-dimensional sparse mixture model, i.e. estimating the five cluster centers b_k for α, $K = 5$ here). The parameters were randomly initialized and the learning rates were set to be 0.0005 (typically 60–80 iterations). After the mixing matrix was estimated, we performed the linear programming algorithm for obtaining the sources. Fig. 4 shows the correlations between the estimated sources and the true sources (scatter plots of the estimated sources Si-esti versus the true sources Si-true, $i = 1, 2, 3, 4, 5$). We can see that the five recovered sources are nicely correlated with one of the true sources and uncorrelated with the remaining sources. For the goal of comparison, we performed the linear programming algorithm for obtaining the estimated sources if the mixing matrix was accuracy. The error of the five estimated sources was 0.2419, 0.5100, 0.5185, 0.6263 and 0.6928, respectively. And the error of the five estimated sources computed by our algorithm was 0.3021, 0.6394, 0.6185, 0.6614 and 0.8570, correspondingly.

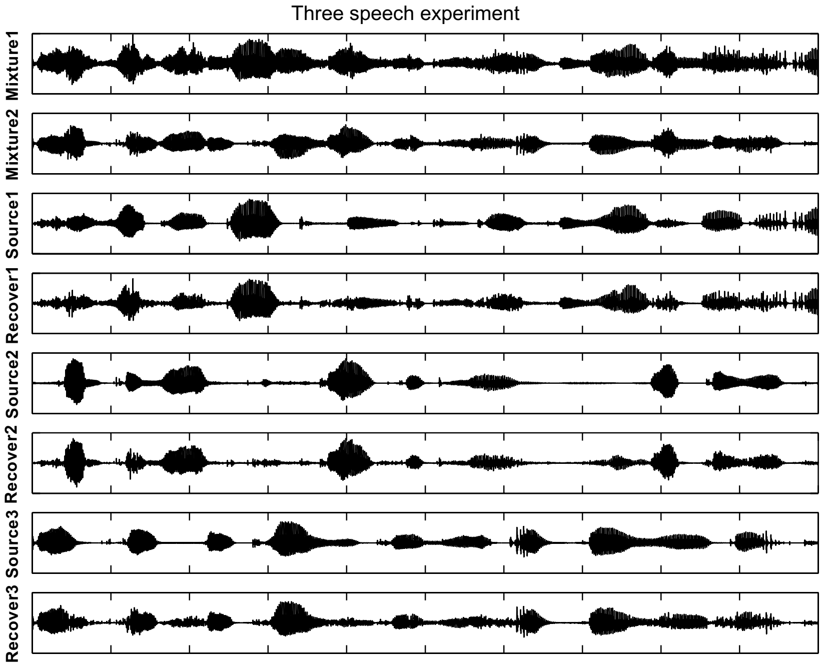

Fig. 3. Three speech experiment: Two mixtures, three original signals and three recovered signals.

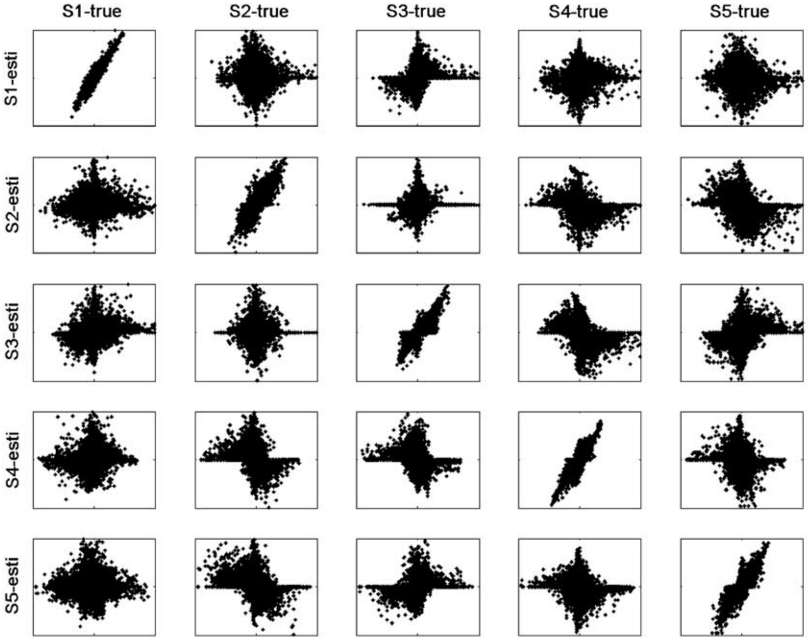

Fig. 4. Demonstration of the separation of five speech source signals from three mixtures. Scatter plots of the estimated sources S*i*-esti versus the true sources S*i*-true, $i = 1, 2, 3, 4, 5$.

5. Conclusions

In this paper we have presented a procedure for the blind separation with more sources than mixtures. If the sources are sparse, the mixing matrix can be estimated by using the sparse mixture model. The sparse mixture model is a powerful simple framework for modelling sparse distribution and provides a general method to learn the mixing matrix for sparse sources. We derive an iterative learning algorithm for the *n*-dimensional sparse mixture model. The coordinates of the cluster centers determine the columns of the estimated mixing matrix. After the mixing matrix is estimated, the sources are estimated by solving a linear programming problem. Several experiments have been presented involving speech signals, with good results, including the successful separation of three sources from two mixtures and five sources from three mixtures. Combining the sparse mixing model with the frequency information (Bofill and Zibulevsky, 2001; Li et al., 2003) and the exact choice of the sparseness measure will be considered in our future work.

Acknowledgements

We are grateful to all the anonymous reviewers who provided insightful and helpful comments. The work was supported by NSFC (60472017, 90103033), MOE (KP0302) and MOST (ICPDF2003).

Appendix A

Derivation of a learning algorithm for sparse mixture model

Assume that the data $\alpha = \{\alpha_1, \ldots, \alpha_T\}$ are drawn independently and generated by a *n*-dimensional sparse mixture model. We derive an iterative learning algorithm which performs gradient ascent on the total likelihood of the data. The likelihood of the data is given by the joint density

$$p(\alpha|\Theta) = \prod_{t=1}^{T} p(\alpha_t|\Theta). \tag{A.1}$$

The mixture density is

$$p(\alpha_t|\Theta) = \sum_{k=1}^{K} p(\alpha_t|\theta_k, k)p(k), \tag{A.2}$$

where $\Theta = (\theta_1, \ldots, \theta_K)$ are the unknown parameters for each $p(\alpha_t|\theta_k,k)$, and we aim to infer them from the data $\alpha = \{\alpha_1, \ldots, \alpha_T\}$ (the number K is known in advance). The log-likelihood L is then

$$L = \sum_{t=1}^{T} \log p(\alpha_t|\Theta) \tag{A.3}$$

and using (A.2), the gradient for the parameters θ_k is

$$\begin{aligned}\nabla_{\theta_k} L &= \sum_{t=1}^{T} \frac{1}{p(\alpha_t|\Theta)} \nabla_{\theta_k} p(\alpha_t|\Theta) \\ &= \sum_{t=1}^{T} \frac{\nabla_{\theta_k}\left[\sum_{k=1}^{K} p(\alpha_t|\theta_k, k)p(k)\right]}{p(\alpha_t|\Theta)} \\ &= \sum_{t=1}^{T} \frac{\nabla_{\theta_k} p(\alpha_t|\theta_k, k)p(k)}{p(\alpha_t|\Theta)}.\end{aligned} \tag{A.4}$$

Using the Bayes's rule, we obtain

$$\begin{aligned}p(k|\alpha_t, \Theta) &= \frac{p(\alpha_t|\theta_k, k)p(k)}{\sum_{k=1}^{K} p(\alpha_t|\theta_k, k)p(k)} \\ &= \frac{p(\alpha_t|\theta_k, k)p(k)}{p(\alpha_t|\Theta)}.\end{aligned} \tag{A.5}$$

Substituting (A.5) in (A.4) leads to

$$\begin{aligned}\nabla_{\theta_k} L &= \sum_{t=1}^{T} p(k|\alpha_t, \Theta) \frac{\nabla_{\theta_k} p(\alpha_t|\theta_k, k)p(k)}{p(\alpha_t|\theta_k, k)p(k)} \\ &= \sum_{t=1}^{T} p(k|\alpha_t, \Theta) \nabla_{\theta_k} \log p(\alpha_t|\theta_k, k).\end{aligned} \tag{A.6}$$

We assume that the component densities $p(\alpha_t|\theta_k, k)$ are modelled as sparse densities typified by the *n*-dimensional Laplacian distributions, i.e.,

$$p(\alpha_t|\theta_k, k) \propto \beta_k^n \exp\left(-\beta_k \sum_{i=1}^{n} |\alpha_{t_i} - b_{k_i}|\right), \tag{A.7}$$

where $\theta_k = \{\beta_k, b_k\}$, $\beta_k \in R$, $b_k = (b_{k_1}, \ldots, b_{k_n})^T \in R^n$, $\alpha_t = (\alpha_{t_1}, \ldots, \alpha_{t_n})^T \in R^n$. We adapt the cluster centers b_k with (A.6)

$$\nabla_{b_k} L = \sum_{t=1}^{T} p(k|\alpha_t, \Theta) \nabla_{b_k} \log p(\alpha_t|\theta_k, k). \tag{A.8}$$

We adapt the width parameter β_k with (A.6)

$$\nabla_{\beta_k} L = \sum_{t=1}^{T} p(k|\alpha_t, \Theta) \nabla_{\beta_k} \log p(\alpha_t|\theta_k, k). \tag{A.9}$$

The goal of the learning algorithm is to determine the cluster centers b_k. For the centers b_k we find, making use of (A.7), we have

$$\begin{aligned}\nabla_{b_k} \log p(\alpha_t|\theta_k, k) &= \nabla_{b_k}\left(-\beta_k \sum_{i=1}^{n} |\alpha_{t_i} - b_{k_i}|\right)\\ &\approx \beta_k \tanh(\gamma(\alpha_t - b_k)),\end{aligned} \tag{A.10}$$

where $\tanh(\gamma(\alpha_t - b_k)) = (\tanh(\gamma(\alpha_{t_1} - b_{k_1})), \ldots, \tanh(\gamma(\alpha_{t_n} - b_{k_n})))^T$, γ is a large positive constant (for details why one chooses the approximation, see (Lewicki and Sejnowski, 2000)). Similarly, for the width parameter β_k, we obtain

$$\begin{aligned}&\nabla_{\beta_k} \log p(\alpha_t|\theta_k, k)\\ &\quad = \nabla_{\beta_k}\left(n \log \beta_k - \beta_k \sum_{i=1}^{n} |\alpha_{t_i} - b_{k_i}|\right)\\ &\quad = \frac{n}{\beta_k} - \sum_{i=1}^{n} |\alpha_{t_i} - b_{k_i}|.\end{aligned} \tag{A.11}$$

From (A.8), (A.9), (A.10) and (A.11), we derive a gradient ascent learning algorithm as follows:

$$\Delta b_k \propto p(k|\alpha_t, \Theta)(\beta_k \tanh(\gamma(\alpha_t - b_k))), \tag{A.12}$$

$$\Delta \beta_k \propto p(k|\alpha_t, \Theta)\left(\frac{n}{\beta_k} - \sum_{i=1}^{n} |\alpha_{t_i} - b_{k_i}|\right). \tag{A.13}$$

Practically, the parameters $\theta_k = \{\beta_k, b_k\}$ are randomly initialized and the adaptation is stopped once the log-likelihood function stabilizes asymptotically with increasing number of iterations.

References

Amari, S.-I., Cichocki, A., Yang, H., 1996. A new learning algorithm for blind source separationAdvances in Neural Information Processing System, vol. 8. MIT Press, Cambridge, MA, pp. 757–763.

Bell, A., Sejnowski, T., 1995. An information-maximization approach to blind separation and blind deconvolution. Neural Comput. 7 (6), 1129–1159.

Bofill, P., Zibulevsky, M., 2001. Underdetermined blind source separation using sparse representations. Signal Process. 81 (11), 2353–2362.

Cardoso, J.-F., Laheld, B.H., 1996. Equivariant adaptive source separation. IEEE Trans. Signal Process. 44 (12), 3017–3030.

Chen, S., Donoho, D.L., Saunders, M.A., 1996. Atomic decomposition by basis pursuit (Tech. Rep.). Department of Statistics, Stanford University, Stanford, CA.

Comon, P., 1994. Independent component analysis—a new concept? Signal Process. 36, 287–314.

Girolami, M., 2001. A variational method for learning sparse and overcomplete representations. Neural Comput. 13 (11), 2517–2532.

Hyvärinen, A., 1999. Fast and robust fixed-point algorithm for independent component analysis. IEEE Trans. Neural Networks 10 (3), 626–634.

Hyvärinen, A., Karhunen, J., Oja, E., 2001. Independent Component Analysis. John Wiley, New York.

Lee, T.W., Girolami, M., Sejnowski, T., 1999a. Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources. Neural Comput. 11 (2), 417–441.

Lee, T.W., Lewicki, M.S., Girolami, M., Sejnowski, T.J., 1999b. Blind source separation of more sources than mixtures using overcomplete representations. IEEE Signal Process. Lett. 6 (4), 87–90.

Lewicki, M.S., Olshausen, B.A., 1999. Probabilistic framework for the adaptation and comparison of image codes. J. Opt. Soc. Amer.: Opt. Image Sci. Vision 16 (7), 1587–1601.

Lewicki, M.S., Sejnowski, T.J., 2000. Learning overcomplete representations. Neural Comput. 12 (2), 337–365.

Li, Y., Cichocki, A., Amari, S., 2003. Sparse component analysis for blind source separation with less sensors than sources. Fourth Internat. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003) Japan, pp. 89–94.

Murata, N., Ikeda, S., Ziehe, A., 2001. An approach to blind source separation based on temporal structure of speech signals. Neurocomputing 41, 1–24.

Murata, N., Kawanabe, M., Ziehe, A., Müller, K.-R., Amari, S.-I., 2002. On-line learning in changing environments with applications in supervised and unsupervised learning. Neural Networ. 15, 743–760.

Shi, Z., Tang, H., Tang, Y., 2004a. A new fixed-point algorithm for independent component analysis. Neurocomputing 56, 467–473.

Shi, Z., Tang, H., Liu, W., Tang, Y., 2004b. Blind source separation of more sources than mixtures using generalized exponential mixture models. Neurocomputing 61, 461–469.

Zhong, M., Tang, H., Chen, H., Tang, Y., 2004. An EM algorithm for learning sparse and overcomplete representations. Neurocomputing 57, 469–476.

Zibulevsky, M., Pearlmutter, B.A., 2001. Blind source separation by sparse decomposition in a signal dictionary. Neural Comput. 13 (4), 863–882.

Zibulevsky, M., Pearlmutter, B.A., Bofill, P., Kisilev, P., 2001. Blind source separation by sparse decomposition in a signal dictionary. In: Robers, S., Everson, R. (Eds.), Independent Component Analysis: Principles and Practice. Cambridge University Press.