
A Mixture Model for Learning Sparse Representations

Greg Brunet

Department of Computer Science
University of Toronto
Toronto, Ontario
greg.brunet@utoronto.ca

Abstract

In a latent variable model, an overcomplete representation is one in which the number of latent variables is at least as large as the dimension of the data observations. Overcomplete representations have been advocated due to robustness in the presence of noise, the ability to be sparse, and an inherent flexibility in modeling the structure of data [9]. In this report, we modify factor analysis to obtain a method for learning overcomplete sparse representations by replacing the Gaussian prior on the factors with a prior that encourages sparseness. This is achieved by using the factorable Laplacian, which implicitly adds a lasso-type penalty term on the latent variables. In order to approximate the intractable integrals introduced into this model, a variational technique is used to lower bound the posterior distributions. Using this lower bound, it is possible to develop an Expectation-Maximization (EM) learning algorithm for estimating the model parameters. We use this technique to extend the sparse factor analysis model to a mixture of sparse factor analyzers and develop an EM algorithm. The new EM algorithm for the mixture model is applied to a handwritten digit recognition problem and is compared to existing methods.

1 Introduction

In many learning models, the observations are reconstructed as a weighted sum of model vectors. In vector quantization, only one of the model vectors is active (non-zero) at any given time. This is referred to as local coding. In factor analysis (FA), the factors are assumed to have a Gaussian prior distribution, and so many of the latent variables can be active at any given time. This is referred to as global coding. In between these two extremes is sparse coding, in which observations are reconstructed using only a few of the model vectors. Sparse coding has a variety of applications, including medical image processing (such as EEG classification) [1], and sound source separation [6]. There is also biological evidence that suggests that the human brain uses sparse coding [4]. For example, humans are able to distinguish between different voices in the presence of several active conversations (noise) with ease. Thus, perhaps understanding sparse coding will shed light on how the human brain performs tasks that are so difficult to simulate algorithmically.

The problem of blind source separation led to the development of Independent Component Analysis (ICA) [8], which is closely related to factor analysis. Unlike factor analysis, ICA has a non-Gaussian and factorable prior on the latent variables. The simplest

form of ICA has as many outputs as sources (the number of latent variables is equal to the dimension of data), which is called a complete representation. In [9], a factorable Laplacian prior is used for learning overcomplete representations. They develop an algorithm that is a generalization of traditional ICA, in which they employ the use of the standard Laplace approximation of the data likelihood about the MAP. However, using the Laplace approximation gives no guarantee of an increase in the data likelihood per iteration, therefore using the same approximation for factor analysis is undesirable. This problem is solved in [6] by exploiting a variational approximation to the Laplacian. The variational approximation provides a strict lower bound on the prior density and the posterior distribution, which guarantees an increase in the data log likelihood. Note that approximating the intractable integrals in non-Gaussian factor analysis is widely studied. For example, see [2] for a survey of several methods. In our case, we can easily apply the variational approximation to obtain a basic sparse factor analyzer (SFA) without any additional theory, as this has been done in [6] with respect to a slightly more general linear model. However, the key benefit in a mixture of factor analyzers (MFA) over basic factor analysis is that each mixture component is able to model a different part of the covariance structure of the data. Therefore, it is natural to extend the basic sparse factor analysis model to a mixture of sparse factor analyzers (MSFA).

The report is structured as follows: Section 2 outlines traditional FA and MFA methods. Section 3 outlines the variational method discussed in [6] and applies it to FA. We then show how to extend this method to apply to a mixture of factor analyzers, just as in the traditional case. Section 4 describes the experiments and results obtained in an application to handwritten digits, and section 5 contains our conclusions.

2 Preliminaries

In this section, we briefly introduce traditional factor analysis and mixture of factor analyzers, as presented in [5]. For the rest of the paper, i , j , and k are used to index the i -th data case, j -th mixture model, and k -th vector component. Also, $N_{\mathbf{x}}(\mu, \Sigma)$ denotes a normal distribution in \mathbf{x} with mean μ and covariance Σ .

2.1 Factor Analysis

In factor analysis, the observed data $\mathbf{x} \in \mathfrak{R}^D$ is modeled using a K -dimensional vector of real-valued factors, \mathbf{z} , where in general K is much smaller than D . The generative model is then:

$$\mathbf{x} = \Lambda \mathbf{z} + \mu, \quad (1)$$

where Λ is a $D \times K$ factor loading matrix. The factors \mathbf{z} are assumed to be $N(0, \mathbf{I})$ distributed, and the D -dimensional random variable μ is distributed $N(0, \Psi)$, where Ψ is a diagonal matrix. The diagonality of Ψ means that the observed variables are independent given the factors. It follows that the data variable \mathbf{x} is distributed with zero mean and covariance $\Lambda \Lambda^T + \Psi$, as we can subtract off the mean μ from the data before hand:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} = N_{\mathbf{x}}(0, \Lambda \Lambda^T + \Psi). \quad (2)$$

The goal of factor analysis is to find Λ and Ψ that best model the covariance of \mathbf{x} . See [5] for the details of an EM algorithm for factor analysis.

2.2 Mixture of Factor Analyzers

Suppose now that we have a mixture of M factor analyzers indexed by $\omega_j, j = 1, \dots, M$, and that each factor is K -dimensional. The generative model has the following mixture

distribution:

$$p(\mathbf{x}) = \sum_{j=1}^M \pi_j \int p(\mathbf{x}|\mathbf{z}, \omega_j) p(\mathbf{z}|\omega_j) d\mathbf{z}, \quad (3)$$

where $\pi_j = p(\omega_j)$. The factors are again assumed to be $N(0, \mathbf{I})$ distributed, so that

$$p(\mathbf{z}|\omega_j) = p(\mathbf{z}) = N(0, \mathbf{I}). \quad (4)$$

In a mixture of factor analyzers, rather than subtracting off the mean, each factor analyzer has its own mean, μ_j , allowing each to model a different part of the observation space. Thus,

$$p(\mathbf{x}|\mathbf{z}, \omega_j) = N(\mu_j + \Lambda_j \mathbf{z}, \Psi). \quad (5)$$

The parameters for MFA are then $\{(\mu_j, \Lambda_j)_{j=1}^M, \pi, \Psi\}$, where π contains the mixing proportions $\pi_j = p(\omega_j)$. An EM algorithm for estimating the parameters is given [5].

3 Sparse Factor Analysis

This section first introduces a sparse version of factor analysis based on a variational technique given in [6], and extends it to a mixture of sparse factor analyzers. An EM algorithm is developed for the new mixture model.

3.1 A single sparse factor analyzer

We now assume that $K \geq D$ (an overcomplete representation). To introduce sparsity into factor analysis, we replace the Gaussian prior on the latent variables (analogous to a ridge regression penalty on the standard objective function) with a factorable Laplacian prior (analogous to a lasso-type penalty - i.e., the L_1 norm of the latent variables):

$$p(\mathbf{z}) = \prod_{k=1}^K \exp(-|\mathbf{z}_k|). \quad (6)$$

The new prior distribution gives more weight to values that are close to zero, thereby encouraging the model to set many latent variables to (or close to) zero. This makes it ideal for learning sparse representations. In particular, it generates concise models by performing variable selection, while still maintaining the stability of the ridge regression penalty [3]. Note that this is not the case for the Gaussian prior, because even if we increase the number of latent variables in the model to a complete or overcomplete basis, the underlying assumption is that the data is still Gaussian, and so many latent variables will be active at all times. In [6], it is shown that the Laplacian prior can be represented as an optimization over the variational parameter $\xi = (\xi_1, \dots, \xi_K)^T$, such that

$$\prod_{i=1}^K \exp(-|z_i|) = \sup_{\xi} \left\{ \prod_{k=1}^K \phi(\xi_k) N_{\mathbf{z}}(0, \mathbf{V}) \right\}, \quad (7)$$

where $\phi(\xi_k) = \sqrt{2\pi|\xi_k|} \exp(-\frac{1}{2}|\xi_k|)$, and $\mathbf{V} = \text{diag}[|\xi_1|, \dots, |\xi_K|]$ ¹. Using the fact that the log prior is a convex function, we have a strict lower bound on the prior density for any

¹For a vector \mathbf{v} , $\text{diag}[\mathbf{v}]$ is a square matrix with \mathbf{v} on the diagonal and zeros elsewhere.

value of the variational parameter ξ as follows:

$$\begin{aligned}
p(\mathbf{x}|\Lambda, \Psi) &\geq p(\mathbf{x}|\Lambda, \Psi, \xi) = \int p(\mathbf{x}|\mathbf{z}, \Lambda_j, \Psi, \xi)p(\mathbf{z}|\xi)d\mathbf{z} \\
&= \prod_{i=1}^K \phi(\xi_i) \int N_{\mathbf{x}}(\Lambda_j \mathbf{z}, \Psi) N_{\mathbf{z}}(0, \mathbf{V}) d\mathbf{z} \\
&= \prod_{i=1}^K \phi(\xi_i) N_{\mathbf{x}}(0, \Lambda \mathbf{V} \Lambda^T + \Psi). \tag{8}
\end{aligned}$$

This also allows us to lower bound the posterior as well:

$$\begin{aligned}
p(z|x, \Lambda, \Psi) &\geq p(z|x, \Lambda, \Psi, \xi) \\
&= \frac{N_x(\Lambda z, \Psi) N_z(0, \mathbf{V})}{\int N_x(\Lambda z, \Psi) N_z(0, \mathbf{V}) dz}, \tag{9}
\end{aligned}$$

which simplifies to a normal distribution with the following first and second moments, respectively:

$$E[\mathbf{z}|\mathbf{x}] = (\Lambda^T \Psi^{-1} \Lambda + \mathbf{V}^{-1})^{-1} \Lambda^T \Psi^{-1} \mathbf{x} \tag{10}$$

$$E[\mathbf{z}\mathbf{z}^T|\mathbf{x}] = (\Lambda^T \Psi^{-1} \Lambda + \mathbf{V}^{-1})^{-1} + E[\mathbf{z}|\mathbf{x}]E[\mathbf{z}|\mathbf{x}]^T. \tag{11}$$

We can now develop an EM algorithm for this model. For each data case \mathbf{x}_i , we associate a K-dimensional variational parameter ξ_i and let $V_i = \text{diag}[|\xi_{i1}|, \dots, |\xi_{iK}|]$. The E-step for this model consists of computing the first and second moments for each data case \mathbf{x}_i as given by equations (10) and (11). To derive the M-step, we note that the expected complete log likelihood for the sparse model has the same form as for the traditional model, and so the updates for Λ and Ψ have the familiar form. Additionally, we use the M-step updates derived in [6] for the variational parameters ξ_i . Putting these together, the M-step is:

$$\Lambda^{new} = \left(\sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T \beta_i \Lambda \mathbf{V}_i \right) \left(\sum_{i=1}^N (\mathbf{V}_i - \mathbf{V}_i \Lambda^T \beta_i (\mathbf{I} - \mathbf{x}_i \mathbf{x}_i^T \beta_i^T) \Lambda \mathbf{V}_i) \right)^{-1} \tag{12}$$

$$\Psi^{new} = \text{diag}_1 \left[S - \frac{1}{N} \Lambda^{new} \sum_{i=1}^N \mathbf{V}_i \Lambda^T \beta_i \mathbf{x}_i \mathbf{x}_i^T \right] \tag{13}$$

$$\begin{aligned}
\xi_i \circ \xi_i &= \text{diag}_2 [E[\mathbf{z}\mathbf{z}^T|\mathbf{x}_i]] \\
&= \text{diag}_2 \left[\mathbf{V}_i - \mathbf{V}_i \Lambda^T \beta_i (\mathbf{I} - \mathbf{x}_i \mathbf{x}_i^T \beta_i^T) \Lambda \mathbf{V}_i \right], \tag{14}
\end{aligned}$$

where \circ denotes component-wise multiplication, $\beta_i = (\Lambda \mathbf{V}_i \Lambda^T + \Psi)^{-1}$, S is the sample covariance, and N is the size of the data set. Additionally, $\text{diag}_1[\mathbf{M}]$ sets the off-diagonal elements of a matrix \mathbf{M} to zero, and $\text{diag}_2[\mathbf{M}]$ is a vector containing the diagonal elements of \mathbf{M} . Unlike [6], we have imposed the standard diagonality constraint on Ψ .

3.2 A mixture of sparse factor analyzers

Using the techniques of the previous section, we can extend the simple sparse factor analyzer to a mixture of sparse factor analyzers and develop an EM algorithm. We introduce additional K-dimensional variational parameters ξ_{ij} , $i = 1, \dots, N$, $j = 1, \dots, M$ for each data case \mathbf{x}_i and each factor analyzer j , and let $V_{ij} = \text{diag}[|\xi_{ij1}|, \dots, |\xi_{ijK}|]$. We follow closely the derivation of the EM algorithm for traditional MFA given in [5]. By equation (8) in section 3.1, for each data case \mathbf{x}_i and mixture index ω_j we have a lower bound on the conditional

distribution $p(\mathbf{x}_i|\omega_j)$ as follows:

$$p(\mathbf{x}_i|\Lambda_j, \Psi, \omega_j) \geq p(\mathbf{x}_i|\Lambda_j, \Psi, \omega_j, \xi_{ij}) = \prod_{k=1}^K \phi(\xi_{ijk}) N_{\mathbf{x}_i}(\mu_j, \Lambda_j \mathbf{V}_{ij} \Lambda_j^T + \Psi). \quad (15)$$

Thus, given the model parameters θ , by equations (3) and (15) we have the following lower bound for any given data case \mathbf{x}_i :

$$p(\mathbf{x}_i|\theta) \geq \sum_{j=1}^M \pi_j \prod_{k=1}^K \phi(\xi_{ijk}) N_{\mathbf{x}_i}(\mu_j, \Lambda_j \mathbf{V}_{ij} \Lambda_j^T + \Psi). \quad (16)$$

For the E-step, we need to compute $E[\omega_j \mathbf{z} | \mathbf{x}_i]$ and $E[\omega_j \mathbf{z} \mathbf{z}^T | \mathbf{x}_i]$ for each ω_j and \mathbf{x}_i . Note that we are now referring to the moments of the approximations, analogous to section 3.1. Fortunately, as pointed out in [5], the following two helpful identities hold:

$$E[\omega_j \mathbf{z} | \mathbf{x}_i] = E[\omega_j | \mathbf{x}_i] E[\mathbf{z} | \omega_j, \mathbf{x}_i] \quad (17)$$

$$E[\omega_j \mathbf{z} \mathbf{z}^T | \mathbf{x}_i] = E[\omega_j | \mathbf{x}_i] E[\mathbf{z} \mathbf{z}^T | \omega_j, \mathbf{x}_i]. \quad (18)$$

Similar to traditional MFA, define

$$h_{ij} = E[\omega_j | \mathbf{x}_i] \propto p(\mathbf{x}_i, \omega_j) = p(\omega_j) p(\mathbf{x}_i | \omega_j) = \pi_j \prod_{k=1}^K \phi(\xi_{ijk}) N(\mu_j, \Lambda_j \mathbf{V}_{ij} \Lambda_j^T + \Psi). \quad (19)$$

Then by equations (10), (11), (17) and (18), we have

$$E[\omega_j \mathbf{z} | \mathbf{x}_i] = h_{ij} \Sigma_{ij} \Lambda_j^T \Psi^{-1} (\mathbf{x}_i - \mu_j) \quad (20)$$

$$E[\omega_j \mathbf{z} \mathbf{z}^T | \mathbf{x}_i] = h_{ij} (\Sigma_{ij} + \Sigma_{ij} \Lambda_j^T \Psi^{-1} (\mathbf{x}_i - \mu_j) (\mathbf{x}_i - \mu_j)^T \Psi^{-1} \Lambda_j \Sigma_{ij}), \quad (21)$$

where $\Sigma_{ij} = (\Lambda_j^T \Psi^{-1} \Lambda_j + \mathbf{V}_{ij}^{-1})^{-1}$. Maximizing the expected complete log likelihood with respect to the traditional model parameters gives the usual form of the M-step update for MFA. We also include the updates derived from the relative log likelihood function in [6] for each ξ_{ij} . Putting these together gives the complete M-step update:

$$[\Lambda_j^{new} \mu_j^{new}] = \left(\sum_{i=1}^N h_{ij} E[\tilde{\mathbf{z}} | \mathbf{x}_i, \omega_j]^T \right) \left(\sum_{i=1}^N h_{ij} E[\tilde{\mathbf{z}} \tilde{\mathbf{z}}^T | \mathbf{x}_i, \omega_j] \right)^{-1} \quad (22)$$

$$\Psi^{new} = \frac{1}{N} \text{diag}_1 \left[\sum_{i=1}^N \sum_{j=1}^M h_{ij} \left(\mathbf{x}_i - \tilde{\Lambda}_j^{new} E[\tilde{\mathbf{z}} | \mathbf{x}_i, \omega_j] \right) \right] \quad (23)$$

$$\pi_j^{new} = \frac{1}{N} \sum_{i=1}^N h_{ij} \quad (24)$$

$$\xi_{ij} \circ \xi_{ij} = \text{diag}_2 [E[\mathbf{z} \mathbf{z}^T | \mathbf{x}_i, \omega_j]], \quad (25)$$

where

$$\tilde{\mathbf{z}} = \begin{pmatrix} \mathbf{z} \\ 1 \end{pmatrix}, \quad E[\tilde{\mathbf{z}} | \mathbf{x}_i, \omega_j] = \begin{pmatrix} E[\mathbf{z} | \mathbf{x}_i, \omega_j] \\ 1 \end{pmatrix}, \quad E[\tilde{\mathbf{z}} \tilde{\mathbf{z}}^T | \mathbf{x}_i, \omega_j] = \begin{pmatrix} E[\mathbf{z} \mathbf{z}^T | \mathbf{x}_i, \omega_j] & E[\mathbf{z} | \mathbf{x}_i, \omega_j] \\ E[\mathbf{z} | \mathbf{x}_i, \omega_j]^T & 1 \end{pmatrix},$$

and $\tilde{\Lambda}_j^{new} = [\Lambda_j^{new} \mu_j^{new}]$. The above M-step has the familiar form of a mixture of factor analyzers. In effect, we have used the lower bounds given in [6] and applied them to a mixture of factor analyzers to obtain approximations to the posterior moments. At each iteration of the EM algorithm, we must also update the variational parameters associated with each conditional distribution $p(\mathbf{x}|\Lambda_j, \Psi, \omega_j)$ to better approximate the exact value.

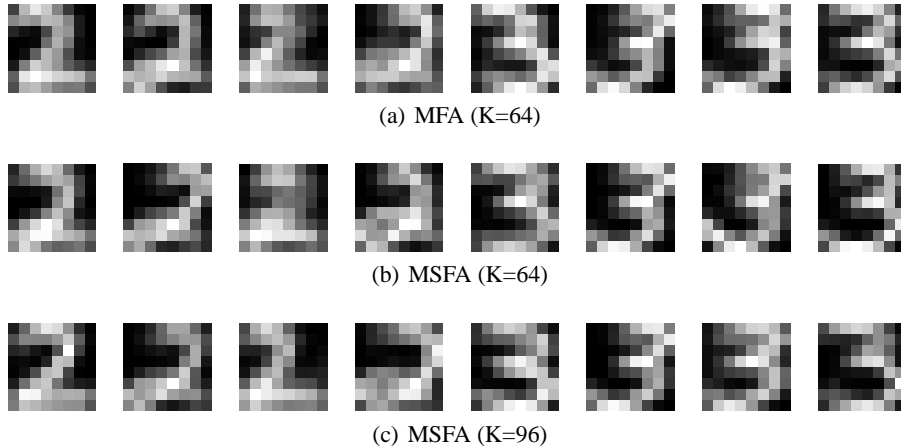


Figure 1: The inferred means for the mixture model components on both data sets: (a) MFA means using 64 latent variables (b) MSFA means using 64 latent variables (c) MSFA means using 96 latent variables.

Method	Data set log likelihood	Average log likelihood
FA (K=64)	2.6302×10^4	23.9113
SFA (K=64)	7.0463×10^4	64.0573
MFA (M=4, K=64)	4.5712×10^4	41.5566
MSFA (M=4, K=64)	7.9650×10^4	72.4094
MSFA (M=4, K=96)	8.8920×10^4	80.8362

Table 1: Various training log likelihoods on the data set for the digit "2"

4 Experiments and Discussion

We have tested the SFA and MSFA models on a collection of handwritten digits obtained from the MNIST database. The representation of the observations \mathbf{x} is a 64-dimensional (8×8) vector of grayscale pixel values for each digit. We trained both models on two separate training sets that correspond to the digits "2" and "3", with each training set consisting of 1100 unprocessed images. The SFA model was trained with 64 latent variables. The MSFA model was trained using a mixture of four factor analyzers, once with $K = 64$ (complete) and once with $K = 96$ (1.5-overcomplete). We also trained a traditional MFA on the same two training sets with a mixture of four analyzers, each with 64 latent variables. As mentioned, traditional factor analysis models are not designed to handle complete or overcomplete representations, but this provides us with a means of comparison on the inferred parameters and quantitative aspects of the training. In essence, we have simply regularized the traditional factor analysis models with a lasso-type penalty term on the latent variables. Therefore, these experiments give us an idea of how well the regularizer performs with respect to overcomplete representations.

The inferred means for the mixture models on each training set are shown in Figure 1. The results for MFA are similar for $K = 96$ and are therefore omitted. We see that the MSFA model has modeled four clusters for each digit that resemble the clusters modeled by the MFA model. In general, the inferred means for MSFA contain less pixel noise around the edges of the digits, indicating a more concise model. For MSFA, when we used 96 latent variables instead of only 64, some of the images smoothed out (for example,

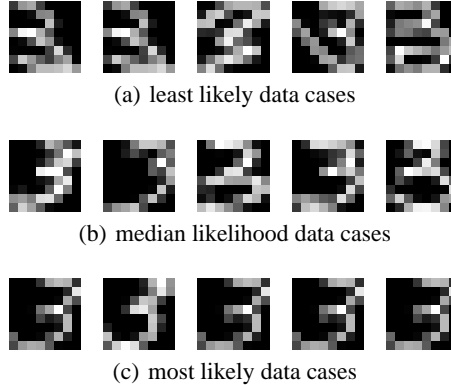


Figure 2: The data cases corresponding to the minimum, median and maximum log likelihoods for FA ($K=64$), SFA ($K=64$), MFA ($K=64$), MSFA ($K=64$), and MSFA ($K=96$), respectively from left to right.

the second last "3" in Figure 1 (b) and (c)). Table 1 contains the data set log likelihoods and average data case log likelihoods on the training set for the digit "2". The results are similar on the training set for the digit "3" and are therefore omitted. The SFA model achieved higher log likelihoods than FA and MFA with 64 latent variables. Also, the MSFA model with 64 latent variables achieved higher log likelihoods than MFA, SFA, and FA with 64 latent variables. However, SFA is not too far off, which is reasonable because we have only used a mixture of four sparse factor analyzers. Finally, training MSFA with 96 latent variables per mixture component yielded the highest log likelihoods. In general, the sparse factor analyzers achieved higher log likelihoods than the traditional factor analyzers in these experiments, and the mixture of sparse analyzer models achieved higher log likelihoods than the basic sparse factor analyzers. As we have calculated the log likelihood for each data case, it is informative to look at the the data cases that correspond to the minimum, median and maximum log likelihoods. These are shown in Figure 2 for each method using 64 latent variables, and for MSFA using 96 latent variables.

When running the experiments, we noticed that the MSFA-EM algorithm was very sensitive to the random initializations of the model parameters. To get around this problem we ran each experiment several times and selected the run with the highest average training log likelihood. This is a problem with EM algorithms in general due to a susceptibility to local minima. However, the MFA-EM algorithm was much less sensitive to the initialization and generally converged in 60-80 iterations, whereas MSFA-EM generally required 100-120 iterations. This may be due to the level of accuracy of the variational approximation to the sparse Laplacian priors at any given iteration. Furthermore, the additional M-step updates for the variational parameters adds to the computational complexity of the MSFA-EM algorithm. Indeed, there is an extra K -dimensional variational parameter for each mixture and data case.

5 Conclusions

In this report we have replaced the Gaussian priors on the latent variables in factor analysis with a factorable Laplacian given by equation (6). This prior encourages sparseness in overcomplete representations because it gives more weight to values close to zero. We then used the variational approximation to the Laplacian in [6] to obtain an EM-algorithm for a single sparse factor analyzer for learning overcomplete representations. The key feature of

this algorithm due to the nature of the approximation is that we have a lower bound on the log likelihood, which guarantees an increase in log likelihoods per EM iteration. We also extended the basic sparse factor analyzer to a mixture of sparse factor analyzers by simply following the FA to MFA extension in [5], and by introducing more variational parameters. The sparse models were applied to a handwritten digit recognition problem, and performed well against the traditional factor analyzer models. This is an indication that the Laplacian prior has the desired effect: avoiding overfitting by forcing parameter decay and doing variable selection, in the spirit of the lasso.

6 References

- [1] A.J. Bell, T-P. Jung, S. Makeig, and T. J. Sejnowski. Independent component analysis for electroencephalographic data. *Advances in Neural Information Processing Systems*, vol. 8, pp. 145-151, 1996.
- [2] K. Chiu, Z. Liu, and L. Xu. Investigations on non-Gaussian factor analysis. *IEEE Signal Processing Letters*, vol. 11, no. 7, pp. 597-600, 2004.
- [3] M. Ferris, B. Klein, R. Klein, et. al. Variable selection and model building via likelihood basis pursuit. University of Wisconsin Technical Report No. 1059r, 2003.
- [4] P. Földiak and M. P. Young. Sparse coding in the primate cortex. *The Handbook of Brain Theory and Neural networks*, pp. 895-898, 1995.
- [5] Z. Ghahramani and G. Hinton. The EM algorithm for mixtures of factor analyzers. University of Toronto Technical Report CRG-TR-96-1, 1996.
- [6] M. Girolami. A variational method for learning overcomplete representations. *Neural Computation*, vol. 13, pp. 2517-2532, 2001.
- [7] I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397-3415, 1993.
- [8] J. Herault and C. Jutten. Space or time adaptive signal processing by neural models. *Proceedings AIP Conference: Neural Networks for Computing*, vol. 151, pp. 206-211, 1986.
- [9] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, vol. 12, pp. 337-365, 2000.