Ensemble Learning for Blind Image Separation and Deconvolution

James Miskin and David J. C. MacKay

Summary. In this chapter Ensemble Learning is applied to the problem of Blind Source Separation and Deconvolution of images. It is assumed that the observed images were constructed by mixing a set of images (consisting of independent identically distributed pixels), convolving the mixtures with unknown blurring filters and then adding Gaussian noise.

Ensemble Learning is used to approximate the intractable posterior distribution over the unknown images and unknown filters by a simpler separable distribution. The mixture of Laplacians used for the source prior respects the positivity of the image and favours sparse images. The model is trained by minimising the Kullback-Leibler divergence between the true posterior distribution and the approximating ensemble.

Unlike Maximum-Likelihood methods, increasing the number of hidden images does not lead to overfitting the data and so the number of hidden images in the observed data can be inferred.

The results show that the algorithm is able to deconvolve and separate the images and correctly identify the number of hidden images.

1 Introduction

Previous work on Blind Source Deconvolution has focused mainly on the problem of deconvolving sound samples. It is assumed that the observed sound samples are temporally convolved versions of the true source samples. Blind Deconvolution algorithms have fallen into two types, those where the inverse of the convolution filter is learnt [1],[3] and those where the aim is to learn the filter itself [1].

When applying these ideas to the problem of deconvolving images two problems become apparent. Firstly in many real data sets (for instance the images generated by telescopes observing the sky or the power spectrum from a Nuclear Magnetic Resonance (NMR) spectrometer) the pixel values correspond to intensities. So the pixel values must be positive. The standard blind separation approaches of assuming that the sources are distributed as $\frac{1}{\cosh}$ [3] or mixtures of Gaussians [2] lose this positivity of the source images. Deconvolution without a positivity constraint leads to reconstructed images that have areas of negative intensity corresponding to energy being sucked out of the detector. Lack of a positivity constraint explains why an optimal linear filter is suboptimal for deconvolution. The derivation of the optimal linear filter assumes that the source image consists of independent identically distributed Gaussian pixels and so does not force positivity [4],[8].

Another problem is that we know that the convolution filters must also be positive (in the case of observing the sky the convolution filter corresponds to blurring due to an imperfect telescope). Most algorithms that learn the inverses of the filters do not take this into account, particularly since positivity is only true of the convolution filter and not of the deconvolution filter. Those algorithms that learn the convolution filter also have problems if they assume that the filter can be inverted exactly, which is not necessarily the case since there may be zeros in the power spectrum of the convolution filter which lead to poles in the power spectrum of the deconvolution filter. Poles in the power spectrum of the deconvolution filter would not be a problem if the observed image were noise free and there were no numerical errors in the calculation but in real problems the inverse is ill conditioned.

It is possible to use MCMC sampling to solve blind inverse problems by sampling from the true posterior density for the latent variables, [7]. Sampling methods have the disadvantage that predictions can only be made by storing a set of samples or by repeating the sampling process whenever samples are needed.

In this chapter we apply Ensemble Learning (as introduced in chapter 6) to the problem of Blind Source Separation and Deconvolution. We apply Ensemble Learning so as to fit the ideal posterior density by an approximation satisfying the positivity constraints. The result of the training is a distribution (not a set of samples from the distribution) and so further inferences can be made by evaluating expectations using this approximation. Ensemble Learning has previously been applied to the Blind Source Separation problem where it was used to separate a sample of speech into independent components [5].

We apply the image separation algorithm to the problem of finding a representation for a sub-set of the MNIST handwritten digit data base. We show that the digit images can be represented by a smaller set of localised images representing parts of the digits. Factorisation of data using a positive constraint on latent variables has previously been used to find a parts based representation for images of faces and for text passages,[6].

2 Separation of Images

We will consider separating a linear mixture of hidden source images. The set of N observed images (each of which are I by J pixels) are assumed to be given by

$$\begin{aligned} y_{ij,n} &= \sum_{m=1}^{M} w_{nm} x_{ij,m} + \nu_{ij,n} \\ &= \hat{y}_{ij,n} + \nu_{ij,n} \end{aligned} \tag{1}$$

where w is an N by M matrix, x is the set of M hidden images (each I by J pixels) and ν is some zero mean Gaussian noise.

The priors over the latent variables (w and x) must respect the positivity constraint. The prior for the matrix elements is a Laplacian

$$p(w_{nm}) = \begin{cases} \beta_w \exp(-\beta_w w_{nm}) & w_{nm} \geq 0 \\ 0 & w_{nm} < 0 \end{cases} \tag{2}$$

where β_w is a scale parameter with the scale invariant prior

$$p\left(\ln \beta_w\right) = 1. \tag{3}$$

The prior for the source pixels is a mixture of Laplacians

$$p\left(x_{ij,n}\right) = \begin{cases} \sum_{\alpha=1}^{N_\alpha} \pi_\alpha \frac{1}{b_\alpha} \exp\left(-\frac{x_{ij,m}}{b_\alpha}\right) & x_{ij,m} \geq 0 \\ 0 & x_{ij,m} < 0 \end{cases} \tag{4}$$

where the hyper-priors are

$$p\left(\ln b_\alpha\right) = 1 \tag{5}$$

$$p\left(\{\pi_\alpha\}\right) \propto \delta\left(\sum_{\alpha=1}^{N_\alpha} \pi_\alpha - 1\right) \prod_{\alpha=1}^{N_\alpha} \pi_\alpha \,. \tag{6}$$

Figure 1 shows how a prior of this form favours sparse images, it should be noted that the prior does not include any prior knowledge we may have about spatial structure.

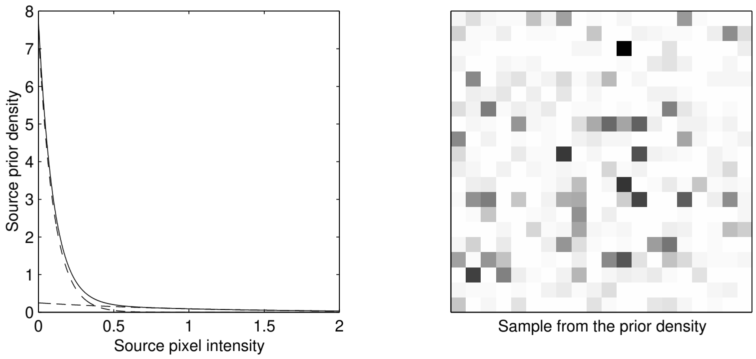

Fig. 1. Example of the form of the source pixel prior density. The sum of Laplacians can have a sharp peak at zero intensity and a long tail. Consequently the prior favours sparse source images

If we assume that we observe the mixtures with additive Gaussian noise, then the likelihood for the observed pixels is

$$p\left(\{y\} \left| \{x\}, \{w\}, \beta_\sigma\right.\right) = \prod_{ij,n} \mathcal{G}\left(y_{ij,n}; \hat{y}_{ij,n}, \beta_\sigma^{-1}\right) \tag{7}$$

where $\mathcal{G}(a; b, c)$ is a Gaussian distribution over a with mean b and variance c. β_σ is the inverse variance of the Gaussian noise and is assigned the hyper-prior

$$p(\ln \beta_\sigma) = 1. \tag{8}$$

We now define the model $\mathcal{H}$ to be the variables M, $\{b_\alpha\}$ and $\{\pi_\alpha\}$ and Θ to be the latent variables $\{x\}$, $\{w\}$, β_σ and β_w. Using Bayes theorem, the posterior density over the latent variables is

$$p(\Theta | \{y\}, \mathcal{H}) = \frac{p(\{y\} | \Theta, \mathcal{H}) \, p(\Theta)}{p(\{y\}, \mathcal{H})}. \tag{9}$$

The process of making inferences involves finding expectations under this probability density (typically expectations of the latent variables themselves), which is analytically intractable. As shown in chapter 6, we can approximate the true posterior by a more tractable distribution, $q(\{x\}, \{w\}, \beta_\sigma, \beta_w)$, for which the expectations are tractable. We can do this by minimising the cost function

$$\begin{aligned} C_{\mathrm{KL}} &= D\left(q(\Theta) \,||\, p(\Theta | \{y\}, \mathcal{H})\right) - \ln p(\{y\} | \mathcal{H}) - \ln p(\{b_\alpha\}) - \ln p(\{\pi_\alpha\}) \\ &= \int q(\Theta) \ln \frac{q(\Theta)}{p(\Theta, \{y\} | \mathcal{H})} d\Theta - \ln p(\{b_\alpha\}) - \ln p(\{\pi_\alpha\}) \\ &= \int \left[\ln \frac{q(\{w\})}{p(\{w\})} + \ln \frac{q(\{x\})}{p(\{x\})} + \ln \frac{q(\beta_\sigma)}{p(\beta_\sigma)} \right. \\ &\quad \left. + \ln \frac{q(\beta_w)}{p(\beta_w)} - \ln p(\{y\} | \Theta, \mathcal{H}) \right] d\Theta - \ln p(\{b_\alpha\}) - \ln p(\{\pi_\alpha\}) \ . \end{aligned} \tag{10}$$

It should be noted that because of the product form of the true posterior density, the cost function can be written as a sum of simpler terms.

2.1 Learning the Ensemble

In order to simplify the posterior density, we choose to use a separable distribution of the form

$$q(\Theta) = \prod_{ijm} q(x_{ij,m}) \times \prod_{nm} q(w_{nm}) \times q(\beta_\sigma) \, q(\beta_w) \,. \tag{11}$$

We will not assume a specific form for these distributions, instead we will find the set of functions that optimises the cost function (subject to the separable form and the constraint that each distribution is normalised).

We can update each distribution in turn, using current estimates for all of the other distributions. To illustrate this, we can consider performing all of the integrations in the cost function with the exception of the integration over β_w

$$C_{\mathrm{KL}} = \int q(\beta_w) \left[\ln q(\beta_w) - \sum_{nm} \left(\ln \beta_w - \beta_w \langle w_{nm} \rangle \right) + \ln \beta_w \right] d\beta_w \tag{12}$$

where we have dropped all terms that are independent of β_w and $\langle . \rangle$ denotes the expectation under the approximating ensemble. We now need to minimise this cost function with respect to the distribution $q(\beta_w)$, subject to the constraint that $q(\beta_w)$ is normalised.

$$\frac{\partial C_{\mathrm{KL}}}{\partial q(\beta_w)} = \ln q(\beta_w) - \sum_{nm} \left(\ln \beta_w - \beta_w \langle w_{nm} \rangle \right) + \ln \beta_w + 1 + \lambda_w \tag{13}$$

where λ_w is a Lagrange multiplier. Setting this derivative to zero, we find that the optimum distribution for β_w is

$$\ln q(\beta_w) = \sum_{nm} \left[\ln \beta_w - \beta_w \langle w_{nm} \rangle \right] - \ln \beta_w - 1 - \lambda_w . \tag{14}$$

Therefore the optimal distribution is

$$q(\beta_w) = \Gamma\left(\beta_w ; \sum_{nm} \langle w_{nm} \rangle , NM \right) \tag{15}$$

where the Γ distribution is

$$\Gamma(a; b, c) = \frac{1}{\Gamma(c)} b^c a^{(c-1)} \exp(-ab) . \tag{16}$$

Similarly we find that the optimal distribution for β_σ is

$$q(\beta_\sigma) = \Gamma\left(\beta_\sigma ; \frac{1}{2} \sum_{ijn} \left\langle \left(y_{ij,n} - \hat{y}_{ij,n} \right)^2 \right\rangle , \frac{IJN}{2} \right) . \tag{17}$$

For the remaining parameters the optimal distributions are

$$q(w_{nm}) = \frac{1}{Z_{nm}^{(w)}} p(w_{nm}) \exp\left(-\frac{1}{2} w_{nm}^{(2)} \left(w_{nm} - w_{nm}^{(1)} \right)^2 \right) \tag{18}$$

$$q(x_{ij,m}) = \frac{1}{Z_{ij,m}^{(x)}} p(x_{ij,m}) \exp\left(-\frac{1}{2} x_{ij,m}^{(2)} \left(x_{ij,m} - x_{ij,m}^{(1)} \right)^2 \right) \tag{19}$$

where $\left\{Z_{nm}^{(w)}\right\}$ and $\left\{Z_{ij,m}^{(x)}\right\}$ are the sets of normalising constants and we have defined

$$
\begin{aligned}
w_{nm}^{(2)} &= \frac{1}{\langle\beta_\sigma\rangle} \sum_{ij} \left\langle x_{ij,m}^2 \right\rangle && (20)\\
w_{nm}^{(1)} w_{nm}^{(2)} &= \langle\beta_\sigma\rangle \sum_{ij} \langle x_{ij,m}\rangle \left(y_{ij,n} - \sum_{m' \neq m} \langle w_{nm'}\rangle \langle x_{ij,m'}\rangle \right) && (21)\\
x_{ij,m}^{(2)} &= \frac{1}{\langle\beta_\sigma\rangle} \sum_{n} \left\langle w_{nm}^2 \right\rangle && (22)\\
x_{ij,m}^{(1)} x_{ij,m}^{(2)} &= \langle\beta_\sigma\rangle \sum_{n} \langle w_{nm}\rangle \left(y_{ij,n} - \sum_{m' \neq m} \langle w_{nm'}\rangle \langle x_{ij,m'}\rangle \right) . && (23)
\end{aligned}
$$

The optimal distributions for w are products of Laplacians and Gaussians, so the optimal distributions are rectified Gaussians (i.e. $q(w_{nm})$ is Gaussian for $w_{nm} \geq 0$ and zero otherwise). Similarly the optimal distributions for x are mixtures of rectified Gaussians. When evaluating the updates for the distributions, it is necessary to evaluate the expectations of the form $\langle w_{nm}\rangle$, $\left\langle w_{nm}^2\right\rangle$, etc. These can be evaluated using error functions.

The distributions can be trained by repeatedly updating each one in turn. But it is important to note that while we have chosen the approximate ensemble such that samples from the distributions are independent, the parameters of the distributions are correlated and so optimisation by successive update of each distribution can be slow to converge.

We can update all of the distributions in parallel by noting that the ensemble can be parametrised by the vector

$$
\boldsymbol{\theta} = \left(\left\{x^{(1)}\right\}, \left\{\log x^{(2)}\right\}, \left\{w^{(1)}\right\}, \left\{\log w^{(2)}\right\}, \log a_w, \log a_\sigma \right) \tag{24}
$$

where

$$
\begin{aligned}
a_w &= \sum_{nm} \langle w_{nm}\rangle \ , && (25)\\
a_\sigma &= \frac{1}{2} \sum_{ijn} \left\langle \left(y_{ij,n} - \hat{y}_{ij,n}\right)^2 \right\rangle \ . && (26)
\end{aligned}
$$

The current estimate of the ensemble can be parametrised by $\boldsymbol{\theta}^{(\tau)}$. We can then define the vector $\boldsymbol{\theta}^{(\mathrm{opt})}$ to be the ensemble formed from the optimal distributions according to Eqns. 15, 17 and 20– 23. A small step along the vector from $\boldsymbol{\theta}^{(\tau)}$ to $\boldsymbol{\theta}^{(\mathrm{opt})}$ must reduce the cost function. Therefore the new ensemble can be defined to be the minimum along the vector from $\boldsymbol{\theta}^{(\tau)}$ to $\boldsymbol{\theta}^{(\mathrm{opt})}$.

If the distributions are independent, a single line minimisation will result in convergence to the optimum distribution (since in this case $\boldsymbol{\theta}^{(\tau+1)} = \boldsymbol{\theta}^{(\text{opt})}$). Alternatively if the distributions are not independent, successive line minimisations will result in convergence to the optimum ensemble.

2.2 Learning the Model

We would also like to be able to infer the parameters of the prior on the source pixels. We can do this by noting that the terms in the cost function relating to the prior on the source pixels are

$$\begin{aligned} C_{\mathrm{KL}} &= -\ln p\left(\{b_\alpha\}\right) - \ln p\left(\{\pi_\alpha\}\right) - \sum_{ij,m} \left\langle \ln p\left(x_{ij,m}\right)\right\rangle \\ &= -\ln p\left(\{b_\alpha\}\right) - \ln p\left(\{\pi_\alpha\}\right) \\ &\quad - \sum_{ij,m} \int q\left(x_{ij,m}\right) \ln \sum_\alpha \pi_\alpha \frac{1}{b_\alpha} \exp\left(-\frac{x_{ij,m}}{b_\alpha}\right) dx_{ij,m} \ . \end{aligned} \tag{27}$$

If the current parameters are $\left\{b_\alpha^{(\tau)}\right\}$ and $\left\{\pi_\alpha^{(\tau)}\right\}$ and the updated parameter values are $\left\{b_\alpha^{(\tau+1)}\right\}$ and $\left\{\pi_\alpha^{(\tau+1)}\right\}$, then by application of Jensen's inequality and discarding constants, a bound on the cost function can be obtained

$$\begin{aligned} C_{\mathrm{KL}} \leq &- \sum_{ij,m,\alpha} \int f_{\alpha,i,j,m}\left(x_{ij,m}\right) \ln \left[\pi_\alpha^{(\tau+1)} \frac{1}{b_\alpha^{(\tau+1)}} \exp\left(-\frac{x_{ij,m}}{b_\alpha^{(\tau+1)}}\right)\right] dx_{ij,m} \\ &- \ln p\left(\left\{b_\alpha^{(\tau+1)}\right\}\right) - \ln p\left(\left\{\pi_\alpha^{(\tau+1)}\right\}\right) \end{aligned} \tag{28}$$

where

$$\begin{aligned} f_{\alpha,i,j,m}\left(x_{ij,m}\right) = &\frac{1}{Z_{ij,m}^{(x)}} \pi_\alpha^{(\tau)} \frac{1}{b_\alpha^{(\tau)}} \exp\left(-\frac{x_{ij,m}}{b_\alpha^{(\tau)}}\right) \\ &\times \exp\left(-\frac{1}{2} x_{ij,m}^{(2)} \left(x_{ij,m} - x_{ij,m}^{(1)}\right)^2\right) \ . \end{aligned} \tag{29}$$

The bound on the cost function can be optimised by setting the new parameters for the prior to

$$\pi_\alpha^{(\tau+1)} = \frac{1 + \sum_{ijm} \left[\int f_{\alpha,i,j,m}\left(x_{ij,m}\right) dx_{ij,m}\right]}{IJM + N_\alpha} \tag{30}$$

$$b_\alpha^{(\tau+1)} = \frac{\sum_{ijm} \left[\int f_{\alpha,i,j,m}\left(x_{ij,m}\right) x_{ij,m} dx_{ij,m}\right]}{1 + \sum_{ijm} \left[\int f_{\alpha,i,j,m}\left(x_{ij,m}\right) dx_{ij,m}\right]} \ . \tag{31}$$

2.3 Example

Figure 2 shows the results of separating a mixture of three grey-scale Dilbert images [Dilbert image Copyright©1997 United Feature Syndicate, Inc., used with permission.]. The images were mixed with a random positive matrix and Gaussian noise was added. The three columns of the figure show the true hidden images, the noisy observations and the ensemble average for the reconstructed images. Three of the reconstructed images match the hidden images. The other two images do not contribute to the mixture. The elements in the w matrix corresponding to those images are set to approximately zero. If we look at (22), the $x^{(2)}$ parameters tend to zero as the elements of w tend to zero and so the posterior density for all of the pixels in the blank images matches the prior density.

We can see that the posterior tends to the prior by looking at Fig. 3 where the KL divergence between the posterior and prior source pixel densities for each reconstructed image is plotted as a function of iteration. It can be seen that the divergence tends to zero for two of the images which means that the posterior and the prior are the same densities.

It might be useful to infer the number of source images that contribute to the observation. In Maximum Likelihood methods, increasing the number of sources cannot decrease the likelihood since the extra source images will model the noise in the observations. Therefore the number of source images will be inferred to be at least as large as the number of observed images.

The correct way to perform the inference of the number of sources is to perform model selection, where each model corresponds to a different number of hidden images. The cost function gives us a bound on the evidence for a model,

$$\ln p\left(\{y\}\left|\mathcal{H}\right.\right) \geq -C_{\mathrm{KL}}. \tag{32}$$

Therefore we can use Bayes theorem to evaluate the posterior probability of a given model using

$$p\left(\mathcal{H}\left|\{y\}\right.\right) = \frac{p\left(\{y\}\left|\mathcal{H}\right.\right) p\left(\mathcal{H}\right)}{p\left(\{y\}\right)}. \tag{33}$$

If we choose a flat prior over the number of hidden images, the model that maximises the posterior distribution is the model that maximises $p\left(\{y\}\left|\mathcal{H}\right.\right)$. We could assume that this is the same as the model that maximises the bound on $p\left(\{y\}\left|\mathcal{H}\right.\right)$ and so the model to choose is the model that minimises C_{KL}.

It may be too time consuming to train multiple models, one for each possible number of images. A simpler method would be to remove source images from the model that do not contribute to the observations, this will reduce the cost function since we know that the KL divergence between the posterior and prior densities for the source pixels and for the mixing matrix elements must be greater than zero. Practically, we remove a source image if the KL divergence between the posterior and prior densities drops to less than 10^{-3} per pixel.

Figure 4 shows how the histograms of intensity compare for the hidden images, the observed images and the recovered images. It should be noted that the recovered images are much more sparse than the observed images.

Fig. 2. Demonstration of the separation of a mixture of three images from a set of five observed images. The left hand column shows the true hidden images. The centre column shows the noisy mixtures of the hidden images. The right hand column shows the reconstructed source images. Three of the reconstructed images match the true images, the remaining two images are uniform as their approximate posterior density, $q(x_{ij,m})$, is equivalent to the prior density. [Dilbert image Copyright©1997 United Feature Syndicate, Inc., used with permission.]

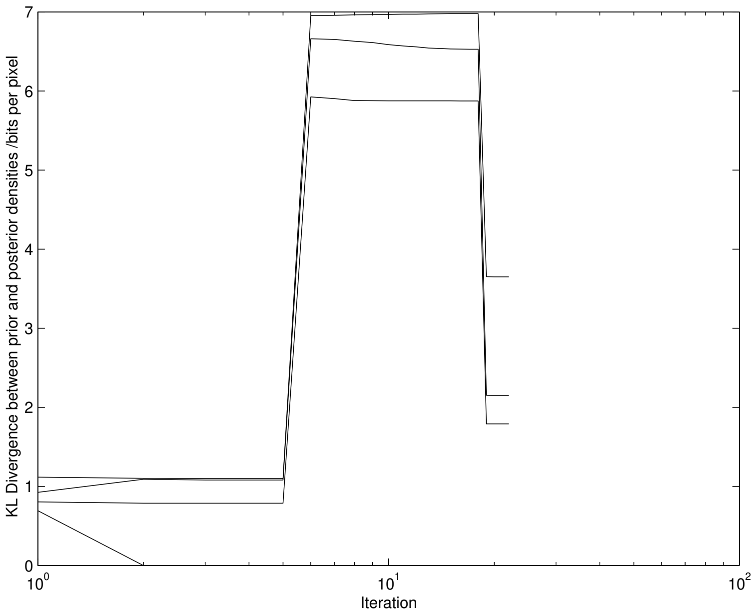

Fig. 3. Variation of the KL divergence between the approximate posterior density and the prior density for each of the reconstructed images. There are three stages to training. During the first stage the source prior and $q(\beta_\sigma)$ are not trained. During the second stage $q(\beta_\sigma)$ is trained so the approximate posterior distributions become sharper and the KL divergence increases. During the final stage the source prior is updated so that it better fits the approximate posterior and the KL divergence drops. For two of the images, the KL divergence tends to zero (that is the posterior density tends to the prior density), these images are not required to be able to reconstruct the observations

2.4 Parts-Based Image Decomposition

Positive constraints on latent pixels have previously been used to find a non-negative factorisation of a set of face data, [6]. In that case it was found that the reconstructed images corresponded to a parts-based decomposition of the face data into localised features.

We can consider trying to find images in a set of natural images by using the EL blind separation algorithm. Figure 5 shows the first 16 examples of handwritten "3"s in the MNIST data set. Figure 6 shows the first 16 PCA components generated from the first 256 "3"s in the MNIST data set. These components represent the highest variance components of the data set, but it is not obvious visually what the set of components represents.

The PCA components do not respect the known positivity of the images (the digits range from white to black or zero ink to lots of ink). Therefore when the PCA components are added together there is an interaction between positive and negative regions in different components to give the positive digit images.

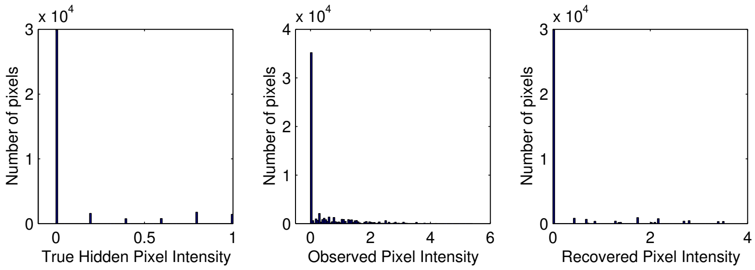

Fig. 4. Histograms of the intensities of the pixels in the images. The first plot shows the true images, here we can see that the images are quantised grey-scale images. The second plot shows the plot for the observed images, as there is a mixing of the quantised levels, the observations are no longer quantised. The third plot shows the histogram for the reconstructed images. The reconstructed images do not match the true images exactly because the ICA model has an invariance with respect to rescaling each source image, but the reconstructed images are more sparse than the observed images

Fig. 5. The first 16 "3"s in the MNIST handwritten data set. The digits are stored as 28x28 pixel grey scale images. Each digit has been preprocessed to center it in the image and to deskew it

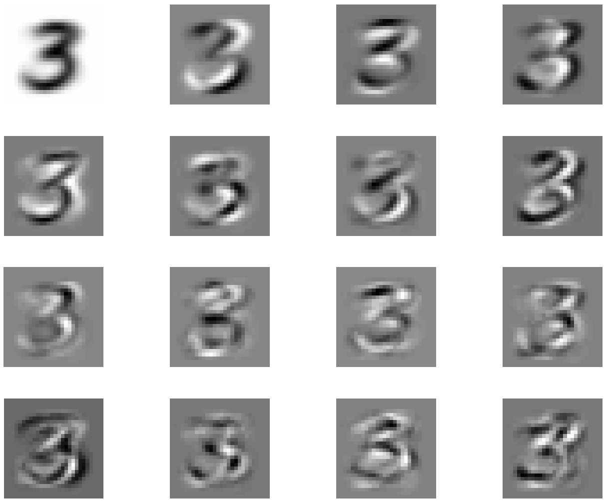

Fig. 6. The first 16 PCA components from the first 256 "3"s in the MNIST data set. Statistically these components correspond to the highest variance components in the data set, but visually it is not obvious what these components represent

Instead we can consider enforcing positivity of the latent images. Applying the Ensemble Learning algorithm to the set of 256 "3"s (assuming that there are 64 hidden images) leads to the decomposition shown in Fig. 7. Instead of the images being based on corrections to a prototype "3" (as in the PCA case) the reconstructed images are all localised and take the form of different shapes of curves, tails, etc. Figure 8 shows the reconstructions of the digits in Fig. 5 using the learnt hidden images. Therefore a parts based decomposition can give a good representation of the data set.

The parts based representation could be used for image compression (by storing images in terms of the parts required to construct them) or as a method of image recognition (by training a set of models of different digits, "1"s, "2"s, etc, a classifier could be made by evaluating the posterior probability of each model for each trial digit).

3 Deconvolution of Images

We can extend the model to include localised blurring of the images. The model for blurring could be used to model the point spread function for a telescope, the line width in NMR experiments or motion blur. The observed images are now defined by

$$\begin{aligned} y_{ij,n} &= \sum_{m=1}^{M} \sum_{k=-K}^{K} \sum_{l=-K}^{K} w_{nm} e_{kl,n} x_{i-k,j-l,m} + \nu_{ij,n} \\ &= \hat{y}_{ij,n} + \nu_{ij,n} \end{aligned} \tag{34}$$

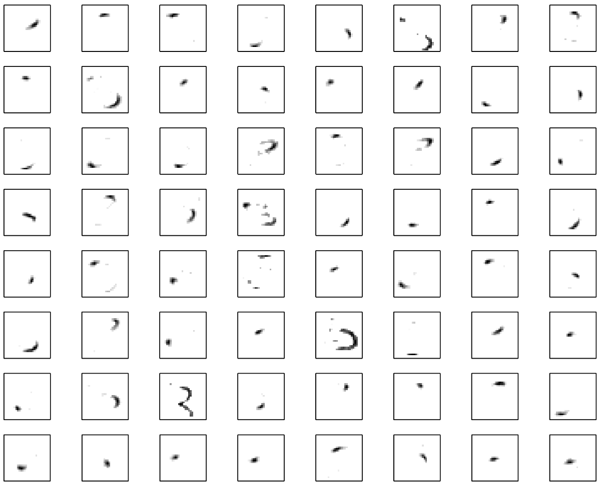

Fig. 7. The hidden images learnt when the 256 "3"s are assumed to be made from a linear combination of 64 non-negative latent images. The learnt images are localised (unlike the PCA components) and each represents a part of a "3" with different images representing different shapes of curves in different positions

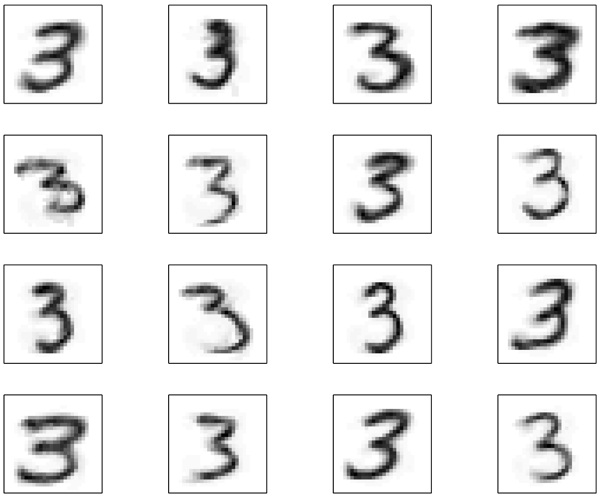

Fig. 8. Reconstructions of the true digits using the 64 non-negative images. Here we can see that the parts based representation is able to model a variety of shapes of "3" and we can see that a set of 64 latent images is able to represent the data set of 256 handwritten "3"s

where w, x, ν and y have the same definitions as the previous model and e is a set of localised convolution filters (one for each image) which extend from $-K$ to K in each dimension. In evaluating this sum, it is assumed that $x_{ij,m}$ is zero outside the defined extent of the image.

The priors for this model are the same as for the previous model with the addition of a prior for the convolution filters. The prior is similar to the prior for the mixing parameters and respects the positivity of the filter.

$$p\left(e_{kl,n}\right) = \begin{cases} \beta_e \exp\left(-\beta_e e_{kl,n}\right) & e_{kl,n} \geq 0 \\ 0 & e_{kl,n} < 0 \end{cases} \tag{35}$$

where β_e is a scale parameter with the scale invariant prior

$$p\left(\ln \beta_e\right) = 1. \tag{36}$$

We can now approximate the true posterior by the separable distribution

$$q\left(\Theta\right) = \prod_{ijm} q\left(x_{ij,m}\right) \times \prod_{nm} q\left(w_{nm}\right) \times \prod_{kln} q\left(e_{kl,n}\right) \times q\left(\beta_\sigma\right) q\left(\beta_w\right) q\left(\beta_e\right). \tag{37}$$

Again we do not assume a specific form for the distributions in $q\left(\Theta\right)$. If we find the optimal distributions we obtain

$$q\left(\beta_w\right) = \Gamma\left(\beta_w; \sum_{nm} \left\langle w_{nm}\right\rangle, NM\right) \tag{38}$$

$$q\left(\beta_e\right) = \Gamma\left(\beta_e; \sum_{kln} \left\langle e_{kl,n}\right\rangle, (2K+1)^2 N\right) \tag{39}$$

$$q\left(\beta_\sigma\right) = \Gamma\left(\beta_\sigma; \frac{1}{2}\sum_{ijn} \left\langle \left(y_{ij,n} - \hat{y}_{ij,n}\right)^2\right\rangle, \frac{IJN}{2}\right) \tag{40}$$

$$q\left(w_{nm}\right) = \frac{1}{Z^{(w)}_{nm}} p\left(w_{nm}\right) \exp\left(-\frac{1}{2} w^{(2)}_{nm} \left(w_{nm} - w^{(1)}_{nm}\right)^2\right) \tag{41}$$

$$q\left(e_{kl,n}\right) = \frac{1}{Z^{(e)}_{kl,n}} p\left(e_{kl,n}\right) \exp\left(-\frac{1}{2} e^{(2)}_{kl,n} \left(e_{kl,n} - e^{(1)}_{kl,n}\right)^2\right) \tag{42}$$

$$q\left(x_{ij,m}\right) = \frac{1}{Z^{(x)}_{ij,m}} p\left(x_{ij,m}\right) \exp\left(-\frac{1}{2} x^{(2)}_{ij,m} \left(x_{ij,m} - x^{(1)}_{ij,m}\right)^2\right) \tag{43}$$

where we have defined

$$w^{(2)}_{nm} = \frac{1}{\left\langle \beta_\sigma \right\rangle} \sum_{ij} \left[\left(\left\langle e_{kl,n}\right\rangle \left\langle x_{i-k,j-l,m}\right\rangle\right)^2\right.$$

$$+\sum_{kl}\left(\left\langle e_{kl,n}^2\right\rangle\left\langle x_{i-k,j-l,m}^2\right\rangle-\left\langle e_{kl,n}\right\rangle^2\left\langle x_{i-k,j-l,m}\right\rangle^2\right)\Bigg] \tag{44}$$

$$\begin{aligned} w_{nm}^{(1)}w_{nm}^{(2)} &= \left\langle\beta_\sigma\right\rangle\sum_{ij}\Bigg[\sum_{kl}\left\langle e_{kl,n}\right\rangle\left\langle x_{i-k,j-l,m}\right\rangle y_{ij,n} \\ &\quad -\sum_{k_1l_1k_2l_2}\sum_{m'\neq m}\left\langle w_{nm'}e_{k_1l_1,n}e_{k_2l_2,n}x_{i-k_1,j-l_1,m}x_{i-k_2,j-l_2,m'}\right\rangle\Bigg] \end{aligned} \tag{45}$$

$$\begin{aligned} e_{kl,n}^{(2)} &= \frac{1}{\left\langle\beta_\sigma\right\rangle}\sum_{ij}\Bigg[\left(\sum_m\left\langle w_{nm}\right\rangle\left\langle x_{i-k,j-l,m}\right\rangle\right)^2 \\ &\quad +\sum_m\left(\left\langle w_{nm}^2\right\rangle\left\langle x_{i-k,j-l,m}^2\right\rangle-\left\langle w_{nm}\right\rangle^2\left\langle x_{i-k,j-l,m}\right\rangle^2\right)\Bigg] \end{aligned} \tag{46}$$

$$\begin{aligned} e_{kl,n}^{(1)}e_{kl,n}^{(2)} &= \frac{1}{\left\langle\beta_\sigma\right\rangle}\sum_{ij}\Bigg[\sum_m\left\langle w_{nm}\right\rangle\left\langle x_{i-k,j-l,m}\right\rangle y_{ij,n} \\ &\quad -\sum_{m_1m_2}\sum_{k_2l_2\neq kl}\left\langle w_{nm_1}w_{nm_2}e_{k_2l_2,n}x_{i-k,j-l,m_1}x_{i-k_2,j-l_2,m_2}\right\rangle\Bigg] \end{aligned} \tag{47}$$

$$x_{ij,m}^{(2)} = \frac{1}{\left\langle\beta_\sigma\right\rangle}\sum_{i'j'n}\left\langle w_{nm}^2\right\rangle\left\langle e_{i'-i,j'-j,n}^2\right\rangle \tag{48}$$

$$\begin{aligned} x_{ij,m}^{(1)}x_{ij,m}^{(2)} &= \left\langle\beta_\sigma\right\rangle\sum_{i_2j_2n}\Big[\left\langle w_{nm}\right\rangle\left\langle e_{i_2-i,j_2-j,n}\right\rangle y_{i_2j_2,n} \\ &\quad -\sum_{m_2k_2l_2\neq mi_2-ij_2-j}\left\langle w_{nm}w_{nm_2}e_{i_2-ij_2-j,n}e_{k_2l_2,n}x_{i_2-k,j_2-l,m_2}\right\rangle\Big]\ . \end{aligned} \tag{49}$$

The posterior distributions can be trained iteratively by performing repeated line minimisations as for the Separation of Images model.

3.1 Examples

Figure 9 shows the results of using the Ensemble Learning algorithm to reconstruct the hidden image and the blurring filter from a single observed image. In each case the reconstructed filter matches the true filter. The reconstructed images match the true hidden images.

Figure 10 shows the results of using the Ensemble Learning algorithm to reconstruct the hidden images and the blurring filters from a set of blurred images (the images are the same as those used in Fig. 2, but with added blurring). In each case the reconstructed filter matches the true filter. The reconstructed images also match the true hidden images.

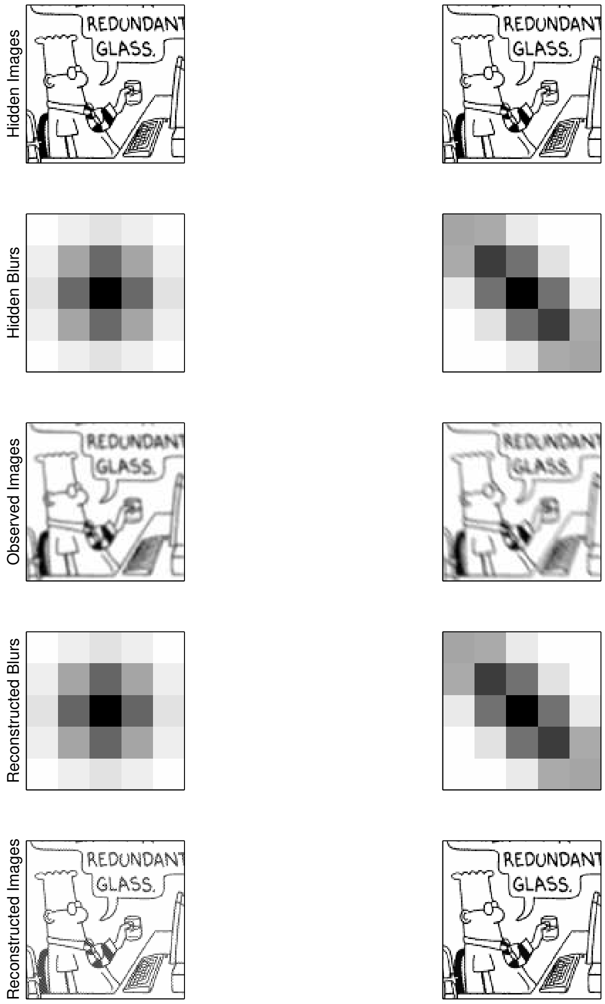

Fig. 9. Demonstration of the deconvolution of two blurred images. In each test the same image was blurred by a different filter. The reconstructed filters match the true filters. The reconstructed images are close to the hidden images. [Dilbert image Copyright©1997 United Feature Syndicate, Inc., used with permission.]

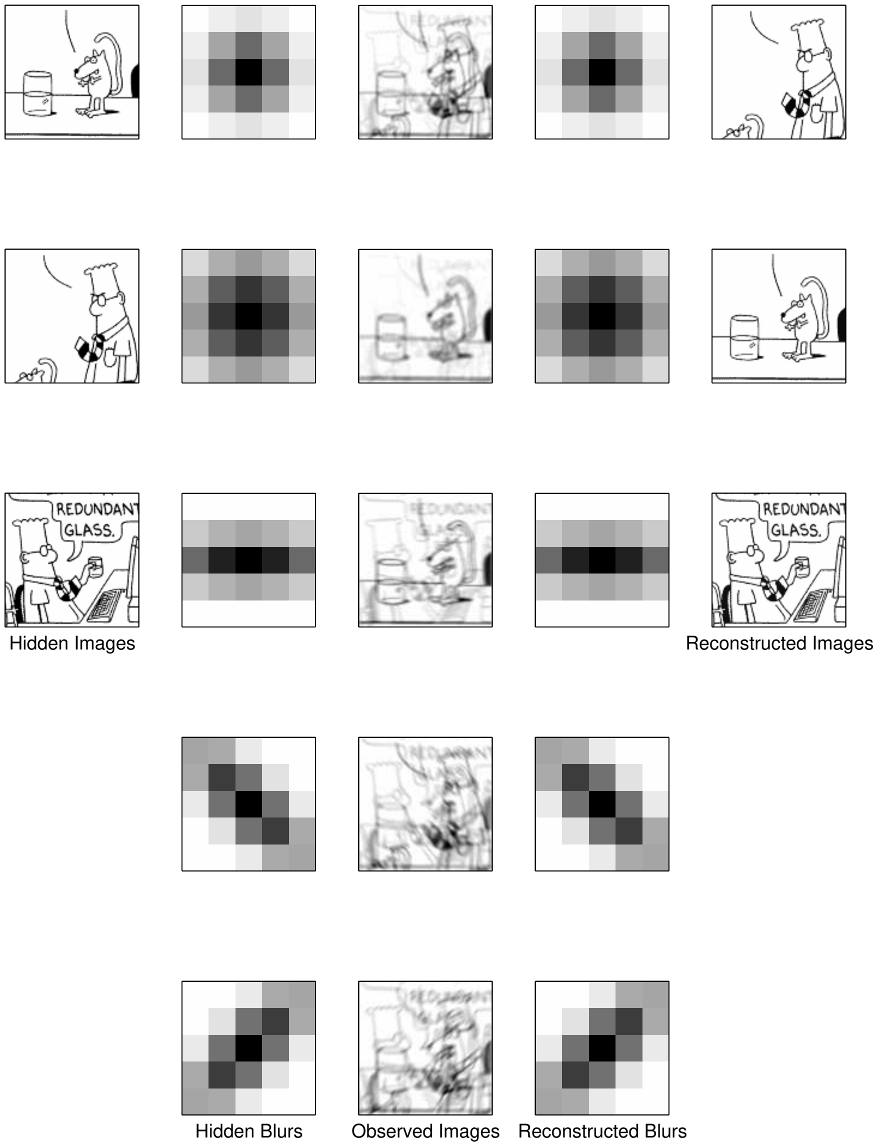

Fig. 10. Demonstration of the deconvolution of multiple images. The source images were mixed and then blurred by a set of localised blur filters. The reconstructed images match the source images showing that the correct mixing matrix and blurring filters were learnt. [Dilbert image Copyright©1997 United Feature Syndicate, Inc., used with permission.]

Figure 11 shows how the histograms of intensity compare for the hidden images, the observed images and the recovered images. As with the pure mixing case, the observed images are much less sparse than the true hidden images. The choice of a sparse prior for the images helps to force a set of sparse reconstructed source images to be found.

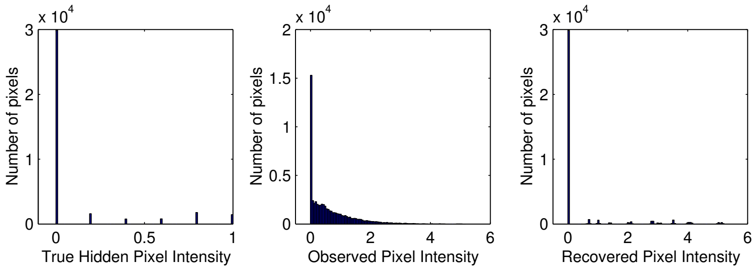

Fig. 11. Histograms of the intensities of the pixels in the multiple blurred images. The first plot shows the true images, here we can see that the images are quantised grey-scale images. The second plot shows the plot for the observed images, as there is a mixing of the quantised levels, the observations are no longer quantised. The third plot shows the histogram for the reconstructed source images

4 Conclusion

Freeform Ensemble Learning allows for tractable solutions to blind inverse probems. Approximating the true posterior by a more tractable separable distribution means that the blind inverse problem can be reduced to a function minimisation problem. Consequently the inverse need not be performed by resorting to an MCMC sampler.

A side effect of using a separable approximating posterior distribution is that correlations between the latent variables in the true posterior distribution are lost. On the other hand the fitting process uses distributions over possible values for all of the parameters, unlike Maximum Likelihood methods which find a point estimate for the latent variables and consequently can suffer from over-fitting to the data.

Use of Ensemble Learning allows the number of hidden images to be inferred by minimising the cost function (or equivalently maximising the bound on the evidence) with respect to the number of hidden images.

The results show that the algorithm is able to deconvolve and separate noisy mixtures of images. The results also show that the algorithm can be used to obtain a parts based representation of images.

For hidden images that have an intrinsic correlation, the images could be modelled by a set of independent pixels (as in the model described above) convolved with another unknown blurring filter. The extra filter could be learnt in a similar way to the filter in this model and may improve separation of images that have intrinsic correlations.

5 Acknowledgements

The authors would like to thank Harri Lappalainen for useful discussions and Yann LeCun for the use of the MNIST data set.

References

1. Haggai Attias: 'Blind source separation and deconvolution: The dynamic component analysis algorithm'. Neural Computation **10**, pp. 1373–1424 (1998)
2. Haggai Attias: 'Independent factor analysis'. Neural Computation **11**, pp. 803–851 (1998)
3. Anthony J. Bell and Terrence J. Sejnowski: 'An information maximisation approach to blind separation and blind deconvolution'. Neural Computation **7**, pp. 1129–1159 (1995)
4. S. F. Gull and G. J. Daniell: 'Image reconstruction from incomplete and noisy data'. Nature **272**, pp. 686–690 (1978)
5. Harri Lappalainen: 'Ensemble learning for independent component analysis'. In: *Proceedings of the First International Workshop on Independent Component Analysis and Blind Signal Separation* (1998)
6. Daniel D. Lee and H. Sebastian Seung: 'Learning the parts of objects by non-negative matrix factorization'. Nature **401**, pp. 788–791 (1999)
7. M. F. Ochs, R. S. Stoyanova, F. Arias-Mendoza, and T. R. Brown: 'A new method for spectral decomposition using a bilinear bayesian approach'. Journal of Magnetic Resonance **137**, pp. 161–176 (1999)
8. John E. Tansley, Martin J. Oldfield, and David J. C. Mackay: 'Neural network image deconvolution'. In: *Maximum Entropy and Bayesian Methods*. ed. by G. R. Heidbreder (Kluwer Academic Publishers 1996) pp. 319–325