Connection between multilayer perceptrons and regression using independent component analysis

Aapo Hyvärinen and Ella Bingham
Neural Networks Research Centre
Helsinki University of Technology
P.O. Box 5400, FIN-02015 HUT, Finland
aapo.hyvarinen@hut.fi
http://www.cis.hut.fi/aapo/

Abstract

The data model of independent component analysis (ICA) gives a multivariate probability density that describes many kinds of sensory data better than classical models like gaussian densities or gaussian mixtures. When only a subset of the random variables is observed, ICA can be used for regression, i.e. to predict the missing observations. In this paper, we show that the resulting regression is closely related to regression by a multi-layer perceptron (MLP). In fact, if linear dependencies are first removed from the data, regression by ICA is, as a first-order approximation, equivalent to regression by MLP. This theoretical result gives a new interpretation of the elements of the MLP: The outputs of the hidden layer neurons are related to estimates of the values of the independent components, and the sigmoid nonlinearities are obtained from the probability densities of the independent components.

1 Introduction

Independent component analysis (ICA) [2, 11, 6, 13] is a recently developed statistical model where we express observed random variables $x_1, x_2, ..., x_q$ as linear combinations of unknown component variables, denoted by $s_1, s_2, ..., s_n$. The components s_i are, by definition, mutually statistically independent, and zero-mean. Let us arrange the observed variables x_i into a vector $\mathbf{x} = (x_1, x_2, ..., x_q)^T$ and the independent components s_i into a vector $\mathbf{s}$, respectively; then the linear relationship is given by

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

Here, $\mathbf{A}$ is an unknown $q \times n$ matrix, called the mixing matrix. The basic problem of ICA estimation is then to estimate the mixing matrix $\mathbf{A}$, as well as the densities of the s_i, using only observations of the mixtures x_j. This means that we try to approximate the joint density of $\mathbf{x}$ as precisely as possible by the densities of sums of independent random variables. We assume here that $n \geq q$, in order to have a nonsingular joint density.

Regression, i.e. prediction, is one of the fundamental problems in supervised learning. In the general regression problem, the variables in $\mathbf{x}$ are divided into two parts, observed and missing, that is, the predicting variables and the variables to be predicted. For simplicity, we can arrange the variables in $\mathbf{x}$ so that the k first variables form the vector of the observed variables $\mathbf{x}_o = (x_1, ..., x_k)^T$, and the remaining variables form the vector of the missing variables $\mathbf{x}_m = (x_{k+1}, ..., x_q)^T$. Thus the model can be written as

$$\begin{pmatrix} \mathbf{x}_o \\ \mathbf{x}_m \end{pmatrix} = \begin{pmatrix} \mathbf{A}_o \\ \mathbf{A}_m \end{pmatrix} \mathbf{s}. \tag{2}$$

The problem is now to predict $\mathbf{x}_m$ for a given observation of $\mathbf{x}_o$. To be able to predict the $\mathbf{x}_m$, we must use (an estimate of) the joint probability distribution of $\mathbf{x}$. Of course, we must have some previous observations of $\mathbf{x}_m$ to be able to estimate the joint probability distribution, that is, to be able to measure how the predicted (missing) variables depend on the predicting (observed) variables. (This is the case for any regression method.) The regression $\hat{\mathbf{x}}_m$ is conventionally defined as the conditional expectation:

$$\hat{\mathbf{x}}_m = E\{\mathbf{x}_m | \mathbf{x}_o\}. \tag{3}$$

Since the data model of ICA describes well some aspects of many kinds of sensory data [15], it would be natural to attempt to use ICA for regression for such data sets. In fact, since the ICA data model gives (an approximation of) the joint probability density of $\mathbf{x}$, it is straightforward, at least in principle, to first model the joint density of $\mathbf{x}$ by ICA, and then, for a given sample of incomplete data, predict the missing values in $\mathbf{x}_m$ using the conditional expectation, which is well defined once the ICA model has been estimated. Thus we obtain

$$E\{\mathbf{x}_m|\mathbf{x}_o\} = \mathbf{A}_m \int_{\mathbf{A}_o\mathbf{s}=\mathbf{x}_o} \mathbf{s}p(\mathbf{s})d\mathbf{s}. \tag{4}$$

In the following, we shall call this generic idea "regression by ICA".

Regression by ICA was already used in [14] to predict missing pixels in images. In [5], the method was considered in a more general setting, and it was proposed that instead of the conditional expectation, i.e. the minimum mean-square error estimator, one could use the maximum a posteriori estimator, which is computationally much simpler. A similar method was considered in [16], though the connection to ICA was not mentioned.

Regression by ICA is parametric,[1] yet nonlinear. It is in fact a direct generalization of ordinary linear regression: if the independent components s_i were gaussian, Eq.(1) would simply give multivariate gaussian distributions, and the conditional expectation would be a linear function of $\mathbf{x}_o$. Regression by ICA is also closely connected to projection pursuit regression [4], because it concentrates on those projections that are the most nongaussian. It could therefore be expected to partially avoid the curse of dimensions.

Thus, ICA gives us one approach to nonlinear regression. A vast literature on regression exists, however, both in neural network and statistics literature, and it would be most useful to know what is the connection between this regression by ICA and classical regression methods. The purpose of this paper is to show that an intimate connection exists between regression by ICA, and regression by multi-layer perceptrons whose structure closely mimics the structure of the ICA model. A two-layer MLP which has the same number of hidden units as the ICA model, and whose nonlinearity is equal to the so-called score function of the independent components gives, as a first-order approximation, the same regression as ICA. It is assumed here that linear dependencies are removed as a preprocessing step. This result gives a new interpretation of MLP's. Moreover, it shows clearly some further relations between regression by ICA and other regression methods.

Some preliminary results were reported in [7].

2 Regression by ICA and by an MLP: The connection

Before announcing our main result, we must discuss the preprocessing of the data. We assume here that the data is first linearly preprocessed so that any linearly predictable part of $\mathbf{x}_m$ is removed. In other words, the $\mathbf{x}_m$ are replaced by the residuals of linear regression. The result of this preprocessing step is that the $\mathbf{x}_o$ and $\mathbf{x}_m$ are uncorrelated. Second, the vectors $\mathbf{x}_o$ and $\mathbf{x}_m$ are each separately whitened. Note that these preprocessing steps cannot be replaced using ordinary whitening methods used in ICA, because they confound the division to observed (predicting) and missing (predicted) variables. As is usual in ICA, this particular form of whitening implies that $\mathbf{A}$ is an orthogonal matrix.

Our result is based on first-order approximations whose accuracy depends on the validity of some assumptions. First, the independent components must have distributions that are not too far from the gaussian distribution; this critical assumption is discussed in Sections 4 and 5. Second, we assume that the dimension of $\mathbf{x}_o$ is large when compared to the dimension of $\mathbf{x}_m$; this assumption seems to be true in most practical cases where multivariate regression is applied.

Let us denote the probability densities of the s_i by p_i, and by $g_i(u) = p_i'(u)/p_i(u) + cu$ a function that equals the negative score function p_i'/p_i of the probability density of s_i, plus an arbitrary linear term, which is the same for all i. For example, the tanh function is the score function of a mildly supergaussian (sparse) distribution [1]. Denote further by g the multi-dimensional function that consists of applying g_i on the i-th component of its argument, for every i. After the above preprocessing and assumptions we have the following result (proven in the Appendix):

$$\boxed{E\{\mathbf{x}_m|\mathbf{x}_o\} \approx \mathbf{A}_m g(\mathbf{A}_o^T\mathbf{x}_o).} \tag{5}$$

[1] We assume here that the distributions of the independent components are either known or modelled by a density family of a limited number of parameters. In general, if the distributions of the independent components are not know, the regression would be semiparametric, though arguably weakly so.

In other words, the regression function for data modeled by ICA, is given by the output of an MLP with one hidden layer. The weight vectors of the MLP are simple functions of the mixing matrix, and the nonlinear activation functions of the MLP are functions of the probability densities of the s_i.

To get insight into this approximation, let us consider supergaussian densities, in which case we can take $g_i(u) = -\tanh(u) + u$ for all i. This is a shrinkage function [8] that approximately reduces the value of its argument by a given constant, resembling a soft-thresholding operation. Now, the vector $\mathbf{A}_o^T \mathbf{x}_o$ can be interpreted as an initial linear estimate of $\mathbf{s}$. (In fact, due to whitening, $\mathbf{A}$ is orthogonal and therefore $\mathbf{A}_o^T$ is equal to the pseudoinverse of $\mathbf{A}_o$.) Thus, the nonlinear aspect of (5) consists largely of *thresholding* the linear estimates of $\mathbf{s}$, to obtain $\hat{\mathbf{s}} = g(\mathbf{A}_o^T \mathbf{x}_o)$. The thresholding can be considered as a way of improving the linear estimate, in a manner similar to the denoising method in [8]. The final linear layer is basically a linear reconstruction of the form $\mathbf{x}_m = \mathbf{A}_m \hat{\mathbf{s}}$.

3 Relation to other methods

3.1 Projection pursuit regression

Our results make as well the connection of regression by ICA to projection pursuit regression quite explicit. Assume that the dimension of the data is very high, and that only certain projections of the data have non-gaussian distributions. One variation of projection pursuit regression [4] would then consist of finding the most nongaussian projections, and using only those projections to construct the regression function. This can be intuitively justified as follows. Since all linear dependecies were removed as a preprocessing step, and the optimal regression for gaussian data is linear, gaussian projections of the data cannot give any new information that would be useful for regression, and thus it is sensible to concentrate on the nongaussian projections.

In fact, if we assume that some of the independent components are gaussian (say, the last ones with indices $i = l+1, ..., r$), the regression function in (5) has the form:

$$E\{\mathbf{x}_m | \mathbf{x}_o\} \approx \sum_{i=1}^{l} \mathbf{v}_i g_i(\mathbf{w}_i^T \mathbf{x}_o) \tag{6}$$

where $\mathbf{w}_i$ is the i-th column of the matrix $\mathbf{A}_o$, and $\mathbf{v}_i$ is the i-th column of the matrix $\mathbf{A}_m$. In this sum, only the l first linear estimates $\mathbf{w}_i^T \mathbf{x}_o$ of the independent components are used, i.e. only those corresponding to the nongaussian components. This is because the linear score function of the gaussian independent components can be taken equal to zero because of the possibility of adding an arbitrary linear term to the nonlinearities g_i. On the other hand, it is a well-known fact in the theory of ICA estimation that the projections in the most nongaussian directions give estimates of the independent components [6, 11]. (This is not exactly true here, though, because we estimate the independent components using a smaller number of observed variables.) Thus we see that the regression given in (5) is closely related to projection pursuit regression, both consisting of using component-wise nonlinearities in the most nongaussian directions.

3.2 Wavelet shrinkage

Regression by ICA is also closely related to wavelet shrinkage [3]. In wavelet shrinkage, the data is first transformed into the wavelet domain. In the regression context, any missing data points are treated as zeros. A thresholding operator is then applied on the wavelet coefficients, and the data is transformed back into the original domain. Consider, for example, prediction (reconstruction) of missing pixels in image data. The utility of such a reconstruction scheme can be intuitively seen in the following way: The linear reconstructions of wavelet coefficients are linear estimates of edges or bars; thresholding them makes edges and bars sharper in the reconstructed image.

It has been shown that the independent components of image windows are quite similar to the wavelet coefficients; the wavelet transform can be thus considered as an approximation of ICA [15, 8]. As discussed above, the nonlinearity in the hidden layer of the MLP can be taken to be a thresholding function when the independent components are supergaussian, as usual with image data. Moreover, since the ICA transform is orthogonal due to whitening, the linear estimation of the independent components, as performed in the first layer of the MLP is equivalent to estimating the independent components as if the missing pixels were zero. Thus, we see that the regression by ICA, according to the approximation in (5), is very closely related to wavelet shrinkage for certain kinds of data, consisting of the same steps of transforming to sparse or independent components, thresholding, and inversion of the transform.

4 Simulations

We performed simple simulations to validate the accuracy of the approximations involved in our result. We generated artificially data according to the ICA model, and compared the true ICA regression with our approximation.

Our simulation data was 100-dimensional and there were $N = 101000$ data samples. The independent components, generated according to some probability density (see below) were mixed using a randomly generated $n \times n$ mixing matrix. The mixtures $\mathbf{x}$ were then divided into observed ($\mathbf{x}_o$) and missing ($\mathbf{x}_m$). The dimensionality of $\mathbf{x}_o$ was 99 and the dimensionality of $\mathbf{x}_m$ was 1. The latter was chosen to facilitate analysis and visualization of results.

In the preprocessing phase, the value of the missing variable $\mathbf{x}_m$ was first predicted by linear regression, and the residual of this regression was used in place of $\mathbf{x}_m$ in the sequel. After this linear prediction, the variables in $\mathbf{x}_o$ were uncorrelated and their variance was set to one; similarly, the variance of $\mathbf{x}_m$ was set to one. Thus the data were whitened.

After the above preprocessing the data was divided in two sets, a training data set of size 100 000 and a test data set of size 1 000. The ICA estimation on the training data set gave the estimated values for the source signals $\mathbf{s}$ and the mixing matrix $\mathbf{A} = \begin{pmatrix} \mathbf{A}_o \\ \mathbf{A}_m \end{pmatrix}$.

The test data set was used to compute estimates for the missing variable $\mathbf{x}_m$. The value of the missing variable $\mathbf{x}_m$ was predicted either using numerical integration as in (4), or using our approximation in (5). The success of the approximation was measured by the correlation coefficient between the two values. Furthermore, we computed the correlation coefficients between the true values of $\mathbf{x}_m$ are the results of numerical integration to see if the very principle of ICA regression is useful.

Three different distributions for the independent components were used, and the results were accumulated over 10 different random seeds.

In the following results, x_m denotes the true value of the missing variable, x_m^{num} is the estimated value computed by numerical integration, and x_m^{appr} is the value given by our MLP-like approximation

Strongly supergaussian data

In the first experiments the independent components $\mathbf{s}$ were generated according to the following strongly supergaussian density [8]:

$$p(s) = \frac{1}{2d} \frac{(\alpha+2)[\alpha(\alpha+1)/2]^{\alpha/2+1}}{[\sqrt{\alpha(\alpha+1)/2} + |s/d|]^{\alpha+3}} \tag{7}$$

where parameter values $\alpha = 1$ and $d = 1$ were chosen, giving

$$p(s) = \frac{1}{2} \frac{3}{(1+|s|)^4}. \tag{8}$$

The strong supergaussianity of this distribution is seen in the fact that the kurtosis is infinite. The score function of this probability density is

$$f'(s) = \frac{(\alpha+3)/d \,\mathrm{sign}(s)}{\sqrt{\alpha(\alpha+1)/2} + |s/d|}. \tag{9}$$

The correlation coefficient between the numerical integration result and our approximation $\rho(x_m^{num}, x_m^{appr})$ was equal to 0.9067, which shows that the approximation was quite good. The scatterplot is shown in Fig. 1 a). Interestingly, if we used the $-\tanh$ nonlinearity instead of the true score function (not shown), the correlation coefficient increased to 0.9303, probably because this is numerically more stable, avoiding the singularity at 0.

As for the succees of the very principle of predicting the actual values of x_m, the correlation coefficient between the true x_m and the numerical integration $\rho(x_m, x_m^{num})$ was 0.9044, which shows that the very principle of ICA regression was feasible: using the ICA model in the regression does indeed give a good regression. This seems to be due to the strong supergaussianity of the s_i. The scatterplot is shown in Fig. 1 b).

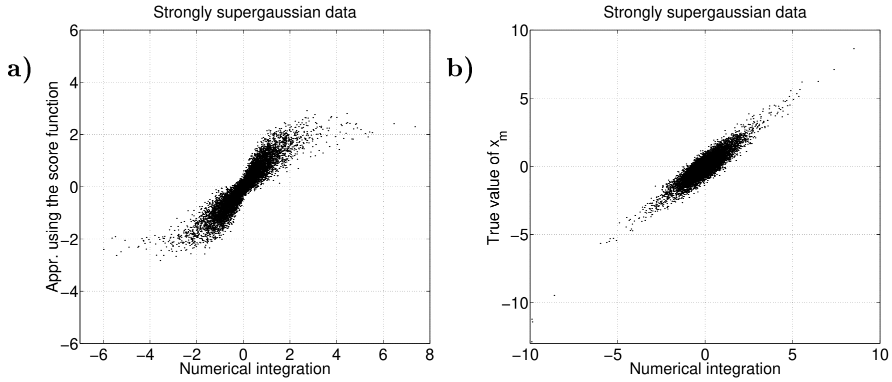

Figure 1: The results for strongly supergaussian data. a) scatterplot of optimal regression by numerical integration vs. regression using our approximation. b) scatterplot of optimal regression by numerical integration vs. true values of x_m.

Laplace distributed data

In the second set of experiments the $\mathbf{s}$ were generated according to the Laplace distribution:

$$p(s) = \frac{\exp(-\sqrt{2}|s|)}{\sqrt{2}} \tag{10}$$

for which the score function is

$$f'(s) = \sqrt{2}\,\text{sign}(s) \tag{11}$$

The correlation coefficient between the numerical integration result and our approximation $\rho(x_m^{num}, x_m^{appr})$ was equal to 0.9120, which shows that the approximation was quite good. See Fig. 2 a).

On the other hand, the estimator x_m^{num} obtained by numerical integration correlates rather poorly with the true value of the missing variable x_m: the correlation coefficient is only 0.6489. See Fig. 2 b). Thus, ICA regression does not work that well in this case. This is probably because its success depends on the nongaussianity of the s_i, and thus requires the s_i to be strongly nongaussian. Likewise, the MLP-like approximation is not very successful in predicting the true value of the missing variable, the correlation coefficient being 0.5843.

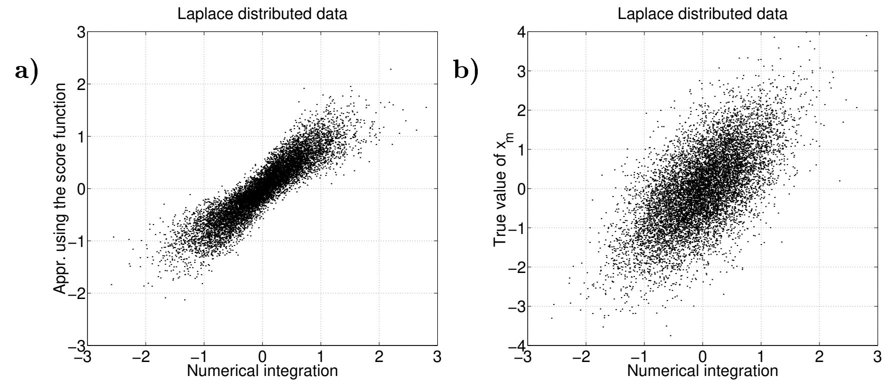

Figure 2: The results for Laplace (moderately supergaussian) data. a) scatterplot of optimal regression by numerical integration vs. regression using our approximation. b) scatterplot of optimal regression by numerical integration vs. true values of x_m.

Very weakly supergaussian data

In the third set of experiments the latent variables $\mathbf{s}$ were generated according to the Cosh distribution:

$$p(s) = \frac{1}{2}\frac{1}{\cosh^2 s} \tag{12}$$

for which the score function is

$$f'(s) = \tanh s. \tag{13}$$

With this weakly supergaussian data, our approximation of the regression function was very good, the correlation coefficient being 0.9965. This was in fact to be expected: Our approximation was a first-order approximation in the vicinity of the gaussian distribution for the s_i, and therefore it not surprising that it works best when the s_i have almost gaussian distributions. The scatterplot is in Fig. 3 a).

On the other hand, we see again that the principle of ICA regression itself does not work well at all due to the weak nongaussianity of the data. The correlation coefficient between the optimal regression computed by numerical integration and the true values of x_m was only 0.2969. See Fig. 3 b). Therefore, the approximating MLP cannot really predict the x_m, either, the correlation coefficient was 0.2954.

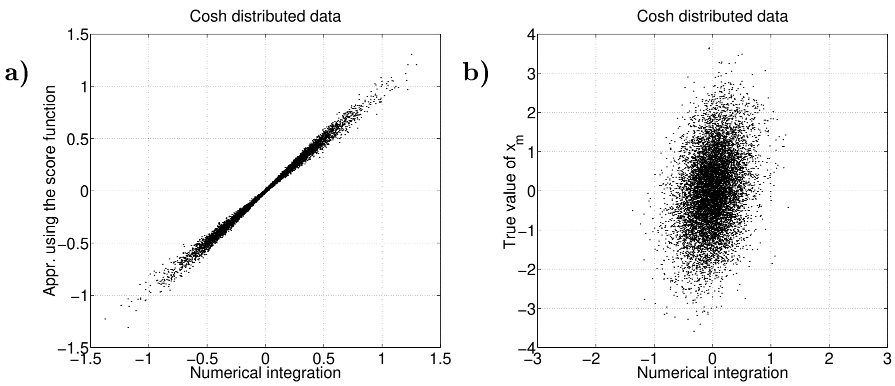

Figure 3: The results for weakly supergaussian data. a) scatterplot of optimal regression by numerical integration vs. regression using our approximation. b) scatterplot of optimal regression by numerical integration vs. true values of x_m.

Conclusion

Thus we see that our approximation works reasonably well. If the distributions of the independent components are close to gaussian, it gives excellent results. If they are strongly supergaussian, the approximation is less accurate but still quite reasonable in the range we experimented with.

Another point is whether ICA regression in itself gives good regression results. Here we consider the prediction of the residuals of linear regression, since linear regression is a standard procedure and does not require the use of nongaussian structure. If the data simply does not contain enough structure, even the optimal regression method fails. We saw that the stronger the supergaussianity, the better the quality of the regression. For strongly supergaussian components, the values can be predicted quite well. In contrast, for weakly supergaussian components, ICA regression does not really explain the data; this is natural since for gaussian data any regression beyond the linear one is impossible.

5 Discussion

We have shown a close connection between regression by ICA and regression by MLP's. Instead of developing a new method either for ICA estimation or nonlinear regression, our main contribution clearly lies in the theoretical insight on what multi-layer perceptrons are doing.

We showed that the output of each hidden-layer neuron in an MLP corresponds to the estimate of one independent component. This means that the problem of choosing the number of hidden units is somewhat equivalent to choosing the number of independent components in the ICA model. Thus this classical problem in MLP research can be seen as a problem of choosing the model order, which is a classical problem in statistical modeling. Likewise, the choice of the nonlinearity is seen to be basically a problem of estimating the probability densities of the independent components.[2] Further, overlearning in MLP's can be seen to correspond to modeling the data with too many independent components, which is a form of overlearning typical of ICA [12]. To avoid overlearning, regularization is often used in MLP's, and similarly, regularizing the mixing matrix in ICA could be most useful [10].

Regression by ICA is, in practice, computationally demanding, due to the (possibly multi-dimensional) integration in (4). Our theoretical result might thus have some practical significance, since it shows that the integration may be approximated by the computationally simple procedure of computing the outputs of an MLP.

It must be noted, however, that the equivalence we have shown is only true as a first-order approximation, for weakly nongaussian independent components. Only experiments can show whether this approximation is good enough in a given real-life application. Our simulations indicate that the approximation might quite well be useful. A second, independent question is, whether the very principle of ICA regression is useful in practice. Again, our simulations indicate that this might be so, if the independent components are strongly nongaussian, but assessing the utility in a real-life situation needs real-life experiments. In fact, we have a kind of contradiction: the approximation is based on the assumption that the components are weakly nongaussian, but the concept of regression by ICA seems to work only if the components are strongly nongaussian. However, the simulations above seem to indicate that our approximation is not bad even for strongly nongaussian variables. The assumption of weak nongaussianity could thus be considered as a technical assumption, allowing the derivation of an approximation that seems to be valid even for the more relevant case of strongly nongaussian components.

In conclusion, our result shows that the regression performed by MLP's, which is conventionally considered as nonparametric or semiparametric, can be interpreted in the framework of ICA as a model-based regression.

References

[1] A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[2] P. Comon. Independent component analysis—a new concept? *Signal Processing*, 36:287–314, 1994.

[3] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society, Ser. B*, 57:301–337, 1995.

[4] P.J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[5] A. Hyvärinen. Sparse regression: Utilizing the higher-order structure of data for prediction. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN'98)*, pages 541–546, Skövde, Sweden, 1998.

[6] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks*, 10(3):626–634, 1999.

[7] A. Hyvärinen. Regression using independent component analysis, and its connection to multi-layer perceptrons. In *Proc. Int. Conf. on Artificial Neural Networks*, pages 491–496, Edinburgh, UK, 1999.

[8] A. Hyvärinen. Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. *Neural Computation*, 11(7):1739–1768, 1999.

[9] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* Wiley Interscience, 2001.

[10] A. Hyvärinen and R. Karthikesh. Sparse priors on the mixing matrix in independent component analysis. In *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, pages 477–452, Helsinki, Finland, 2000.

[2] Note that the nonlinearities given by the score functions need not be known a priori: they can be estimated, just like the mixing matrix, by methods developed in ICA research, see [9]. The same is true for the number of independent components, though this is a much more difficult problem and satisfactory solutions may not be available [9].

[11] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[12] A. Hyvärinen, J. Särelä, and R. Vigário. Spikes and bumps: Artefacts generated by independent component analysis with insufficient sample size. In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, pages 425–429, Aussois, France, 1999.

[13] C. Jutten and J. Hérault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[14] M. Lewicki and B. Olshausen. A probabilistic framework for the adaptation and comparison of image codes. *J. Opt. Soc. Am. A: Optics, Image Science, and Vision*, 16(7):1587–1601, 1998.

[15] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[16] Z. Roth and Y. Baram. Multidimensional density shaping by sigmoids. *IEEE Trans. on Neural Networks*, 7(5):1291–1298, 1996.

A Appendix: Proof of (5)

Denote $h_i(s_i) = s_i^2/2 - \frac{1}{2}\log 2\pi + \log p_i(s_i)$. The variances of the s_i are equal to one by definition. Due to the assumption of near-gaussianity, $h_i(s_i)$ can thus be considered infinitesimal. We can write

$$E\{\mathbf{x}_m|\mathbf{x}_o\} = \mathbf{A}_m \frac{1}{(2\pi)^{n/2}} \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s}\exp(\sum_i[-s_i^2/2 + h_i(s_i)])d\mathbf{s} \tag{14}$$

Now, let us do a first-order approximation of $\sum_i h_i(s_i)$ in the vicinity of the point $\mathbf{A}_o^T\mathbf{x}_o$, i.e. the linear estimate of the independent components. This point is a linear approximation of the point where $p(\mathbf{s}|\mathbf{x}_o)$ is maximized. These approximations are likely to be rather exact if the dimension of $\mathbf{x}_o$ is large and the dimension of $\mathbf{x}_m$ is small. We obtain

$$E\{\mathbf{x}_m|\mathbf{x}_o\} \approx \mathbf{A}_m \frac{1}{(2\pi)^{n/2}} \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s}\exp(\sum_i[-s_i^2/2 + H_i(\mathbf{w}_i^T\mathbf{x}_o) + h_i'(\mathbf{w}_i^T\mathbf{x}_o)(s_i - \mathbf{w}_i^T\mathbf{x}_o)])d\mathbf{s} \tag{15}$$

where $\mathbf{w}_i$ denotes the i-th column of $\mathbf{A}_o$. Now we can use the fact that $\exp(h_i(\mathbf{w}_i^T\mathbf{x}_o))$ is of order $1 + O(h)$. We can ignore this constant, since any change it could make would be infinitesimal. Further, let us denote the constant $\exp(\sum_i -h'(\mathbf{w}_i^T\mathbf{x}_o)\mathbf{w}_i^T\mathbf{x}_o)$ by c_1. Thus we have

$$E\{\mathbf{x}_m|\mathbf{x}_o\} \approx \mathbf{A}_m \frac{c_1}{(2\pi)^{n/2}} \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s}\exp(-\|\mathbf{s}\|^2/2 + h'(\mathbf{A}_o^T\mathbf{x}_o)^T\mathbf{s}])d\mathbf{s}$$
$$\approx \mathbf{A}_m \frac{c_1c_2}{(2\pi)^{n/2}} \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s}\exp(-\frac{1}{2}\|\mathbf{s} - h'(\mathbf{A}_o^T\mathbf{x}_o)\|^2)d\mathbf{s} \tag{16}$$

where h denotes the function where the h_i are applied componentwise. Here we have defined the constant $c_2 = \exp(\sum_i h_i'(\mathbf{w}_i^T\mathbf{x}_o)^2)$.

Thus we have only a gaussian integral left. It can be evaluated by making a norm-preserving variable change that parameterizes the space of the $\mathbf{s}$ such that $\mathbf{x}_o = \mathbf{A}_o\mathbf{s}$. This is given as $\mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{u}$ where $\mathbf{u}$ is not constrained. Thus we obtain

$$\frac{c_1c_2}{(2\pi)^{n/2}} \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s}\exp(-\frac{1}{2}\|\mathbf{s} - h'(\mathbf{A}_o^T\mathbf{x}_o)\|^2)d\mathbf{s}$$
$$= \frac{c_1c_2}{(2\pi)^{n/2}} \int_{\mathbf{u}} [\mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{u}]\exp(-\frac{1}{2}\|[\mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{u}] - h'(\mathbf{A}_o^T\mathbf{x}_o)\|^2)d\mathbf{u}$$
$$= \frac{c_1c_2}{(2\pi)^{n/2}} \int_{\mathbf{u}} [\mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{u}]\exp(-\frac{1}{2}[\|\mathbf{x}_o\|^2 + \|\mathbf{u}\|^2 + \|h'(\mathbf{A}_o^T\mathbf{x})\|^2 - 2h'(\mathbf{A}_o^T\mathbf{x}_o)^T\mathbf{A}_o^T\mathbf{x}_o - 2h'(\mathbf{A}_o^T\mathbf{x}_o)^T\mathbf{A}_m^T\mathbf{u}])d\mathbf{u}$$
$$= \frac{c_1c_2}{(2\pi)^{n/2}} \exp(-\|\mathbf{x}_o\|^2/2 + h'(\mathbf{A}_o^T\mathbf{x}_o)^T\mathbf{A}_o^T\mathbf{x}_o) \int_{\mathbf{u}} [\mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{u}])\exp(-\frac{1}{2}\|\mathbf{u} - \mathbf{A}_m h'(\mathbf{A}_o^T\mathbf{x}_o)\|^2)d\mathbf{u} \tag{17}$$

where we have used the fact that the preprocessing implies that $\mathbf{A}_m\mathbf{A}_o^T = 0$ and $\mathbf{A}_m\mathbf{A}_m^T = \mathbf{A}_o\mathbf{A}_o^T = \mathbf{I}$. This can be evaluated by considering the gaussian integral as an expectation of a gaussian random vector. Furthermore, note that c_1 cancels the latter term in the exponential that is before the integral sign. Somewhat less rigorously, we could also assume that c_2 is approximately cancelled by the first term in that exponential; in any case this is only a scalar scaling. Thus we obtain

$$\frac{c_1 c_2}{(2\pi)^{n/2}} \int_{\mathbf{x}_o=\mathbf{A}_o\mathbf{s}} \mathbf{s} \exp(-\frac{1}{2}\|\mathbf{s} - h'(\mathbf{A}_o^T\mathbf{x}_o)\|^2) d\mathbf{s} \approx \mathbf{A}_o^T\mathbf{x}_o + \mathbf{A}_m^T\mathbf{A}_m h'(\mathbf{A}_o^T\mathbf{x}_o) \tag{18}$$

and we finally have

$$E\{\mathbf{x}_m|\mathbf{x}_o\} \approx \mathbf{A}_m h'(\mathbf{A}_o^T\mathbf{x}_o) \tag{19}$$

where we have again used the fact that the preprocessing implies that $\mathbf{A}_m\mathbf{A}_o^T = 0$. Here, $h_i'(u)$ is defined as $h_i'(u) = u + (\log p_i)'(u)$. On the other hand, $\mathbf{A}_m\mathbf{A}_o^T = 0$ implies that addition of any linear function to h' does not change the regression. Therefore, one can take $h_i'(u) = (\log p_i)'(u) + cu$, i.e. h_i' can be defined as the negative score function of s_i plus any linear function. The linear function must be the same for all i.