Sparse Code Shrinkage: Denoising of Nongaussian Data by Maximum Likelihood Estimation

Aapo Hyvärinen
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O. Box 5400, FIN-02015 HUT, Finland
Email: aapo.hyvarinen@hut.fi

Second revised submission for Neural Computation

May 4, 1999

Abstract

Sparse coding is a method for finding a representation of data in which each of the components of the representation is only rarely significantly active. Such a representation is closely related to redundancy reduction and independent component analysis, and has some neurophysiological plausibility. In this paper, we show how sparse coding can be used for denoising. Using maximum likelihood estimation of nongaussian variables corrupted by gaussian noise, we show how to apply a soft-thresholding (shrinkage) operator on the components of sparse coding so as to reduce noise. Our method is closely related to the method of wavelet shrinkage, but it has the important benefit over wavelet methods that the representation is determined solely by the statistical properties of the data. The wavelet representation, on the other hand, relies heavily on certain mathematical properties (like self-similarity) that may be only weakly related to the properties of natural data.

1 Introduction

Sparse coding (Barlow, 1994; Field, 1994; Olshausen and Field, 1996; Olshausen and Field, 1997) is a method for finding a neural network representation of multidimensional data in which only a small number of neurons is significantly activated at the same time. Equivalently, this means that a given neuron is activated only rarely. In this paper, we assume that the representation is linear. Denote by $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ the observed n-dimensional random vector that is input to a neural network, and by $\mathbf{s} = (s_1, s_2, ..., s_n)^T$ the vector of the transformed component variables, which are the n linear outputs of the network. Denoting further the weight vectors of the neurons by $\mathbf{w}_i, i = 1, ..., n$, and by $\mathbf{W} = (\mathbf{w}_1, ..., \mathbf{w}_n)^T$ the weight matrix whose rows are the weight vectors, the linear relationship is given by

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

We assume here that that the number of sparse components, i.e., the number of neurons, equals the number of observed variables, but this need not be the case in general. The idea in sparse coding is to find the weight matrix $\mathbf{W}$ so that the components s_i are as 'sparse' as possible. A zero-mean random variable s_i is called sparse when it has a probability density function with a peak at zero, and heavy tails; for all practical purposes, sparsity is equivalent to supergaussianity (Hyvärinen and Oja, 1997) or leptokurtosis (positive kurtosis) (Kendall and Stuart, 1958).

Sparse coding is closely related to independent component analysis (ICA) (Bell and Sejnowski, 1995; Comon, 1994; Hyvärinen and Oja, 1997; Karhunen et al., 1997b; Jutten and Herault, 1991; Oja, 1997). In the data model used in ICA, one postulates that $\mathbf{x}$ is a linear transform of independent components: $\mathbf{x} = \mathbf{A}\mathbf{s}$. Inverting the relation, one obtains (1), with $\mathbf{W}$ being the (pseudo)inverse of $\mathbf{A}$. Moreover, it has been proven that the estimation of the ICA data model can be reduced to the search for uncorrelated directions in which the components are as nongaussian as possible (Comon, 1994; Hyvärinen, 1997b). If the independent components are sparse (more precisely, supergaussian), this amounts to the search for uncorrelated projections which have as sparse distributions as possible. Thus, estimation of the ICA model for sparse data is roughly equivalent to sparse coding if the components are constrained to be uncorrelated. This connection to ICA also shows clearly that sparse coding may be considered as a method for redundancy reduction, which was indeed one of the primary objectives of sparse coding in the first place (Barlow, 1994; Field, 1994).

Sparse coding of sensory data has been shown to have advantages from both physiological and information processing viewpoints (Barlow, 1994; Field, 1994). However, thorough analyses of the utility of such a coding scheme have been few. In this paper, we introduce and analyze a statistical method based on sparse coding. Given a signal corrupted by additive gaussian noise, we attempt to *reduce gaussian noise* by soft thresholding ('shrinkage') of the sparse components. Intuitively, because only a few of the neurons are active simultaneously in a sparse code, one may assume that the activities of neurons with small absolute values are purely noise and set them to zero, retaining just a few components with large activities. This method is then shown to be very closely connected to the wavelet shrinkage method (Donoho et al., 1995). In fact, sparse coding may be viewed as a principled, adaptive way for determining an orthogonal wavelet-like basis based on data alone. Another advantage of our method is that the shrinkage nonlinearities can be adapted to the data as well.

This paper is organized as follows. In Section 2, the problem is formulated as maximum likelihood estimation of nongaussian variables corrupted by gaussian noise. In Section 3, the optimal sparse coding transformation is derived. Section 4 presents the resulting algorithm of sparse code shrinkage. Section 5 discusses the connections to other methods, and Section 6 contains simulation results. Some conclusions are drawn in Section 7.

Some preliminary results have appeared in (Hyvärinen et al., 1998b). A somewhat related method was independently proposed in (Lewicki and Olshausen, 1998).

2 Maximum Likelihood Denoising of Nongaussian Variables

2.1 Maximum likelihood estimator in one dimension

The starting point of a rigorous derivation of our denoising method is the fact that the distributions of the sparse components are nongaussian. Therefore, we shall begin by developing a general theory that shows how to remove gaussian noise from nongaussian variables, making minimal assumptions on the data.

We consider first only scalar random variables. Denote by s the original nongaussian random variable, and by ν gaussian noise of zero mean and variance σ^2. Assume that we only observe the random variable y:

$$y = s + \nu \tag{2}$$

and we want to estimate the original s. Denoting by p the probability density of s, and by $f = -\log p$ its negative log-density, the maximum likelihood method gives the following estimator[1] for s:

$$\hat{s} = \arg\min_u \frac{1}{2\sigma^2}(y-u)^2 + f(u). \tag{3}$$

[1] This might also be called a maximum a posteriori estimator.

Assuming f to be strictly convex and differentiable, this minimization is equivalent to solving the following equation:

$$\frac{1}{\sigma^2}(\hat{s} - y) + f'(\hat{s}) = 0 \tag{4}$$

which gives

$$\hat{s} = g(y) \tag{5}$$

where the inverse of the function g is given by

$$g^{-1}(u) = u + \sigma^2 f'(u). \tag{6}$$

Thus, the ML estimator is obtained by inverting a certain function involving f', or the score function (Schervish, 1995) of the density of s. For nongaussian variables, the score function is nonlinear, and so is g.

In general, the inversion required in (6) may be impossible analytically. Here we show three examples (which will later be shown to have great practical value) where the inversion can be done easily.

Example 1 Assume that s has a Laplace (or double exponential) distribution of unit variance (Field, 1994). Then $p(s) = \exp(-\sqrt{2}|s|)/\sqrt{2}$, $f'(s) = \sqrt{2}\,\mathrm{sign}(s)$, and g takes the form

$$g(y) = \mathrm{sign}(y)\max(0, |y| - \sqrt{2}\sigma^2). \tag{7}$$

(Rigorously speaking, the function in (6) is not invertible in this case, but approximating it by a sequence of invertible functions, (7) is obtained as the limit). The function in (7) is a *shrinkage* function that reduces the absolute value of its argument by a fixed amount, as depicted in Fig 1. Intuitively, the utility of such a function can be seen as follows. Since the density of a supergaussian random variable (e.g., a Laplace random variable) has a sharp peak at zero, it can be assumed that small values of y correspond to pure noise, i.e., to $s = 0$. Thresholding such values to zero should thus reduce noise, and the shrinkage function can indeed be considered a soft thresholding operator.

Example 2 More generally, assume that the score function is approximated as a linear combination of the score functions of the gaussian and the Laplace distributions:

$$f'(s) = as + b\,\mathrm{sign}(s), \tag{8}$$

with $a, b > 0$. This corresponds to assuming the following density model for s:

$$p(s) = C \exp(-as^2/2 - b|s|), \tag{9}$$

where C is an irrelevant scaling constant. Then we obtain

$$g(u) = \frac{1}{1 + \sigma^2 a} \operatorname{sign}(u) \max(0, |u| - b\sigma^2). \tag{10}$$

This function is a shrinkage with additional scaling, as depicted in Fig 1.

Example 3 Yet another possibility is to use the following strongly supergaussian probability density:

$$p(s) = \frac{1}{2d} \frac{(\alpha + 2) \left[\alpha \left(\alpha + 1\right)/2\right]^{(\alpha/2+1)}}{\left[\sqrt{\alpha \left(\alpha + 1\right)/2} + \mid s/d \mid\right]^{(\alpha+3)}}. \tag{11}$$

with parameters $\alpha, d > 0$. When $\alpha \to \infty$, the Laplace density is obtained as the limit. The strong sparsity of the densities given by this model can be seen e.g. from the fact that the kurtosis (Field, 1994; Hyvärinen and Oja, 1997) of these densities is always larger than the kurtosis of the Laplace density, and reaches infinity for $\alpha \leq 2$. Similarly, $p(0)$ reaches infinity as α goes to zero. The resulting shrinkage function given by (6) can be obtained after some straightforward algebraic manipulations as:

$$g(u) = \operatorname{sign}(u) \max(0, \frac{|u| - ad}{2} + \frac{1}{2}\sqrt{(|u| + ad)^2 - 4\sigma^2(\alpha + 3)}\ ) \tag{12}$$

where $a = \sqrt{\alpha(\alpha + 1)/2}$, and $g(u)$ is set to zero in case the square root in (12) is imaginary. This is a shrinkage function that has a certain thresholding flavor, as depicted in Fig. 1.

Strictly speaking, the negative log-density of (11) is not convex, and thus the minimum in (5) might be obtained in a point not given by (12): in this case, the point 0 might be the true minimum. To find the true minimum, the value of likelihood at $g(y)$ should be compared with its value at 0, which would lead to an additional thresholding operation. However, such a thresholding changes the estimate only very little for reasonable values of the parameters d and α, and therefore we omit it, using (12) as a simpler and very accurate approximation of the minimization in (3).

Fig. 2 shows some densities corresponding to the above examples. In the general case, even if (6) cannot be inverted, the following first-order approximation of the ML estimator (with respect to noise level) is always possible:

$$\hat{s}^* = y - \sigma^2 f'(y), \tag{13}$$

still assuming f to be convex and differentiable. This estimator is derived from (4) simply by replacing $f'(\hat{s})$, which cannot be observed, by the observed quantity $f'(y)$; these two quantitites are equal to first order. The problem with the estimator in (13) is that the sign of $\hat{s}^*$ is often different from the sign of y even for symmetrical zero-mean densities. Such counterintuitive estimates are possible because f' is often discontinuous or even singular at 0, which implies that the first-order approximation is quite inaccurate near 0. To alleviate this problem of 'overshrinkage' (Efron and Morris, 1975), one may use the following modification:

$$\hat{s}^o = \mathrm{sign}(y)\max(0, |y| - \sigma^2|f'(y)|). \tag{14}$$

Thus we have obtained the exact maximum likelihood estimator (5) of a nongaussian random variable corrupted by gaussian noise, and its two approximations in (13) and (14).

2.2 Analysis of denoising capability

In this subsection, we analyze the denoising capability of the ML estimator given in (5). We show that, roughly, the more nongaussian the variable s is, the better gaussian noise can be reduced. Nongaussianity is here measured by Fisher information. Due to the intractability of the general problem, we consider here the limit of infinitesimal noise, i.e., all the results are first-order approximations with respect to noise level.

To begin with, recall the definition of Fisher information (Cover and Thomas, 1991) of a random variable s with density p:

$$I_F(s) = E\{[\frac{p'(s)}{p(s)}]^2\}. \tag{15}$$

The Fisher information of a random variable (or, strictly speaking, of its density) equals the conventional, 'parametric' Fisher information (Schervish, 1995) with respect to a hypothetical location parameter (Cover and Thomas, 1991).

Fisher information can be considered as a measure of nongaussianity. It is well-known (Huber, 1985) that in the set of probability densities of unit variance, Fisher information is minimized by the gaussian density, and the minimum equals 1. Fisher information is not, however, invariant to scaling; for a constant a, we have

$$I_F(as) = \frac{1}{a^2} I_F(s). \tag{16}$$

The main result on the performance of the ML estimator is the following theorem, proven in the Appendix:

Theorem 1 *Define by (5) the estimator $\hat{s} = g(y)$ of s in (2). For small σ, the mean-square error of the estimator $\hat{s}$ is given by*

$$E\{(s-\hat{s})^2\} = \sigma^2[1-\sigma^2 I_F(s)] + o(\sigma^4), \tag{17}$$

where σ^2 is the variance of the gaussian noise ν.

To get more insight into the Theorem, it is useful to compare the noise reduction of the ML estimator with the best *linear* estimator in the minimum mean square (MMS) sense. If s has unit variance, the best linear estimator is given by

$$\hat{s}_{lin} = \frac{y}{1+\sigma^2}. \tag{18}$$

This estimator has the following mean-square error:

$$E\{(s-\hat{s}_{lin})^2\} = \frac{\sigma^2}{1+\sigma^2}. \tag{19}$$

We can now consider the ratio of these two errors, thus obtaining an index that gives the percentage of additional noise reduction due to using the nonlinear estimator $\hat{s}$:

$$R_s = 1 - \frac{E\{(\hat{s}-s)^2\}}{E\{(\hat{s}_{lin}-s)^2\}}. \tag{20}$$

The following corollary follows immediately:

Corollary 1 *The relative improvement in noise reduction obtained by using the nonlinear ML estimator instead of the best linear estimator, as measured by R_s in (20), is given by*

$$R_s = (I_F(s)-1)\sigma^2 + o(\sigma^2), \tag{21}$$

for small noise variance σ^2, and for s of unit variance.

Considering the above-mentioned properties of Fisher information, Theorem 1 thus means that the more nongaussian s is, the better we can reduce noise. In particular, for sparse variables, the sparser s is, the better the denoising works. If s is gaussian, $R = 0$, which follows from the fact that the ML estimator is then equal to the linear estimator $\hat{s}_{lin}$. This shows again that for gaussian variables, allowing nonlinearity in the estimation does not improve the performance, whereas for nongaussian (e.g. sparse) variables, it can lead to significant improvement[2].

[2] For multivariate gaussian variables, however, improvement can be obtained by Stein estimators (Efron and Morris, 1975).

2.3 Extension to multivariate case

All the results in the preceding subsection can be directly generalized for n-dimensional random vectors. Denote by $\mathbf{y}$ an n-dimensional random vector, which is the sum of an n-dimensional nongaussian random vector $\mathbf{s}$ and the noise vector $\boldsymbol{\nu}$:

$$\mathbf{y} = \mathbf{s} + \boldsymbol{\nu}. \tag{22}$$

where the noise $\boldsymbol{\nu}$ is gaussian and of covariance $\sigma^2\mathbf{I}$. We can then estimate the original $\mathbf{s}$ in the same way as above. Denoting by p the n-dimensional probability density of $\mathbf{s}$, and by $f = -\log p$ its negative log-density, the maximum likelihood method gives the following estimator for $\mathbf{s}$

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{u}} \frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{u}\|^2 + f(\mathbf{u}) \tag{23}$$

which gives

$$\hat{\mathbf{s}} = \mathbf{g}(\mathbf{y}) \tag{24}$$

where the function $\mathbf{g}$ is defined by

$$\mathbf{g}^{-1}(\mathbf{u}) = \mathbf{u} + \sigma^2\nabla f(\mathbf{u}). \tag{25}$$

The counterpart of Theorem 1 is as follows

Theorem 2 *Define by (24) the estimator $\hat{\mathbf{s}} = \mathbf{g}(\mathbf{y})$ of $\mathbf{s}$ in (22). For small σ, the quadratic error of the estimator $\hat{\mathbf{s}}$ is given by*

$$E\{(\mathbf{s}-\hat{\mathbf{s}})(\mathbf{s}-\hat{\mathbf{s}})^T\} = \sigma^2[1 - \sigma^2 I_F(\mathbf{s})] + o(\sigma^4), \tag{26}$$

where the covariance matrix of the gaussian noise ν equals $\sigma^2\mathbf{I}$.

The multidimensional Fisher information matrix is defined here as

$$I_F(\mathbf{s}) = E\{\nabla f(\mathbf{s})\nabla f(\mathbf{s})^T\}. \tag{27}$$

However, the multivariate case seems to be of little importance in practice. This is because it is difficult to find meaningful approximations of the multivariate score function ∇f; the usual approximation by factorizable densities would simply be equivalent to considering the components y_i separately. Moreover, the inversion of (25) seems to be quite intractable for non-factorizable densities. Therefore, in the rest of this paper, we use only the 1-D results given in the previous subsections, applying them separately for each component of a random vector. If the components of the random vector are independent, this does not reduce the performance of the method; otherwise, this can be considered as a tractable approximation of the multivariate ML estimator.

2.4 Parameterization of 1-D densities

Above, it was assumed that the density of the original nongaussian random variable s is known. In practice, this is often not the case: the density of s needs to be modelled with a parameterization that is rich enough. In the following we present parametric density models that are especially suitable for our method. In the main practical applications of the ML estimation, the densities encountered are supergaussian, so we first describe two parameterizations for sparse densities, and then a more general method.

2.4.1 Models of sparse densities

We have developed two convenient parameterizations for sparse densities, which seem to describe very well most of the densities encountered in image denoising. Moreover, the parameters are easy to estimate, and the shrinkage nonlinearity g can be obtained in closed form. Both models use two parameters and are thus able to model different degrees of supergaussianity, in addition to different scales, i.e. variances. The densities are here assumed to be symmetric and of zero mean.

The *first model* is suitable for supergaussian densities that are not sparser than the Laplace distribution, and is given by the family of densities in (9). Indeed, since the score function (i.e., f') of the gaussian distribution is a linear function, and the score function of the typical supergaussian distribution, the Laplace density, is the sign function, it seems reasonable to approximate the score function of a symmetric, moderately supergaussian density of zero mean as a linear combination of these two functions. The corresponding shrinkage function is given by (10).

To estimate the parameters a and b in (9) and (10), we can simply project the score function (i.e. the derivative of the log-density) of the observed data on the two functions in (8). To define the projection, a metric has to be chosen; following (Pham et al., 1992), we choose here the metric defined by the density p. Thus we obtain (see Section 2.4.2 and Appendix)

$$\begin{aligned} b &= \frac{2p_s(0)E\{s^2\} - E\{|s|\}}{E\{s^2\} - [E\{|s|\}]^2} \\ a &= \frac{1}{E\{s^2\}}[1 - E\{|s|\}b] \end{aligned} \tag{28}$$

where $p_s(0)$ is the value of the density function of s at zero. Corresponding estimators of a and b can be then obtained by replacing the expectations in (28) by sample averages; $p_s(0)$ can be estimated, e.g., using a single kernel at 0. It is here assumed that one has access to a noise-free version of the random variable s; this assumption is discussed in the next Section. It is also a good idea to constrain the values of a and b to belong to the intervals $[0, 1/E\{s^2\}]$ and $[0, \sqrt{2/E\{s^2\}}]$, respectively, since we are here *interpolating* the score function between the score function of the gaussian density and the score function of the Laplace density, and values outside of these ranges would lead to an extrapolation whose validity may be very questionable.

The *second model* describes densities that are sparser than the Laplace density, and is given by (11). A simple method for estimating the parameters $d, \alpha > 0$ in (11) can be obtained e.g. from the relations

$$\begin{aligned} d &= \sqrt{E\{s^2\}} \\ \alpha &= \frac{2 - k + \sqrt{k(k+4)}}{2k - 1} \end{aligned} \tag{29}$$

with $k = d^2 p_s(0)^2$. The corresponding shrinkage function is given by (12).

Examples of the shapes of the densities given by (9) and (11) are given in Fig. 2, together with a Laplace density and a gaussian density. For illustration purposes, the densities in the plot are normalized to unit variance, but these parameterizations allow the variance to be choosen freely. The corresponding nonlinearities, i.e. shrinkage functions are given in Fig. 1.

Tests for choosing whether model (9) or (11) should be used are simply to construct. We suggest that if

$$\sqrt{E\{s^2\}} p_s(0) < \frac{1}{\sqrt{2}}, \tag{30}$$

the first model in (9) be used; otherwise use the second model in (11). The limit case $\sqrt{E\{s^2\}} p_s(0) = \sqrt{1/2}$ corresponds to the Laplace density, which is contained in both models.

2.4.2 General case

We present here a simple method for modelling the density of s in the general case, i.e. when the densities are not necessarily sparse and symmetric. In fact, considering the estimators in (5), (13), and (14), it can be seen that what one really needs is an model of the score function f' instead of the density itself. Assume that we approximate the score function $f' = -p'/p$ as the linear combination of two functions, one of which is a linear function:

$$f'(\xi) = a\xi + bh(\xi) \tag{31}$$

and where h is some function to be specified. To estimate the constants a and b, we can simply project f' on the two functions, as above.

Thus, after some quite tedious algebraic manipulations (see Appendix), we obtain the following values for a and b in (31)

$$b = \frac{E\{h'(s)\}E\{s^2\} - E\{sh(s)\}}{E\{h(s)^2\}E\{s^2\} - [E\{sh(s)\}]^2}$$
$$a = \frac{1}{E\{s^2\}}[1 - E\{sh(s)\}b] \tag{32}$$

Corresponding estimators of a and b can be obtained by replacing the expectations in (32) by sample averages. In fact, (28) is obtained as a special case of (32).

3 Finding the Sparse Coding Transformation

3.1 Transforming data to increase denoising capability

In the previous section, it was shown how to reduce additive gaussian noise in nongaussian random variables by means of ML estimation. Theorem 1 showed that the possible noise reduction is proportional to the Fisher information of the distribution of the nongaussian random variable. Fisher information measures roughly two aspects of the distribution: its nongaussianity, and its scale. The Fisher information takes larger values for distributions that are not similar to the gaussian distribution, and have small variances.

Assume now that we observe a multivariate random vector $\tilde{\mathbf{x}}$ which is a noisy version of the nongaussian random vector $\mathbf{x}$:

$$\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\nu}. \tag{33}$$

where the noise $\boldsymbol{\nu}$ is gaussian and of covariance $\sigma^2\mathbf{I}$. As was mentioned in Section 2.3, the ML method seems to be tractable only in one dimension, which implies that we treat every component of $\tilde{\mathbf{x}}$ separately. However, before applying the ML denoising method, we would like to *transform the data* so that the (component-wise) *ML method reduces noise as much as possible.* We shall here restrict ourselves to the class of linear, orthogonal transformations. This restriction is justified by the fact that orthogonal transformations leave the noise structure intact, which makes the problem more simply tractable. Future research may reveal larger classes of transformations where the optimal transformation can be easily determined. Given an orthogonal (weight) matrix $\mathbf{W}$, the transformed vector equals

$$\mathbf{W}\tilde{\mathbf{x}} = \mathbf{W}^T\mathbf{x} + \mathbf{W}^T\boldsymbol{\nu} = \mathbf{s} + \tilde{\boldsymbol{\nu}}. \tag{34}$$

The covariance matrix of $\tilde{\boldsymbol{\nu}}$ equals the covariance matrix of $\boldsymbol{\nu}$, which means that the noise remains essentially unchanged.

The noise reduction obtained by the ML method is, according to Theorem 1, proportional to the sum of the Fisher informations of the components $s_i = \mathbf{w}_i^T \mathbf{x}$. Thus, the optimal orthogonal transformation $\mathbf{W}_{opt}$ can be obtained as

$$\boxed{\mathbf{W}_{opt} = \arg\max_{\mathbf{W}} \sum_{i=1}^{n} I_F(\mathbf{w}_i^T \mathbf{x})} \tag{35}$$

where $\mathbf{W}$ is constrained to be orthogonal, and the $\mathbf{w}_i$ are the rows of $\mathbf{W}$.

To estimate the optimal orthogonal transform $\mathbf{W}_{opt}$, we assume that we have access to a random variable $\mathbf{z}$ that has the same statistical properties as $\mathbf{x}$, and can be observed without noise. This assumption is not unrealistic on many applications: for example, in image denoising it simply means that we can observe noise-free images that are somewhat similar to the noisy image to be treated, i.e., they belong to the same environment or context. This simplifies the estimation of $\mathbf{W}_{opt}$ considerably; the optimal transformation can then be determined by (35), using $\mathbf{z}$ instead of $\mathbf{x}$.

Let us remark that in addition to the above criterion of minimum mean-square error, the optimal transformation could also be derived using maximum likelihood estimation of a generative model. We shall not use this alternative method here; see instead (Hyvärinen et al., 1998a).

3.2 Approximating Fisher information: General case

To use (35) in practice, we need a simple approximation (estimator) of Fisher information. A rough but computationally simple approximation can be obtained by approximating the score function as a sum of a linear function and an arbitrary nonlinearity h, as in (31). This gives (see Appendix) the following approximation of Fisher information:

$$I_F(\mathbf{w}_i^T \mathbf{z}) \approx \frac{1}{E\{(\mathbf{w}_i^T \mathbf{z})^2\}}[1 + \frac{[E\{h'(\mathbf{w}_i^T \mathbf{z})\}E\{(\mathbf{w}_i^T \mathbf{z})^2\} - E\{\mathbf{w}_i^T \mathbf{z}\, h(\mathbf{w}_i^T \mathbf{z})\}]^2}{E\{h(\mathbf{w}_i^T \mathbf{z})^2\}E\{(\mathbf{w}_i^T \mathbf{z})^2\} - [E\{\mathbf{w}_i^T \mathbf{z}\, h(\mathbf{w}_i^T \mathbf{z})\}]^2}] \tag{36}$$

The quantity in (36) can be easily estimated by sample averages.

3.3 Approximating Fisher information: Sparse densities

In the case of sparse distributions, a much simpler approximation of Fisher information is possible. Instead of the general approximation in (36), we can make a local approximation in the vicinity of a known sparse distribution. It is proven in the Appendix that if the density of $\mathbf{w}_i^T\mathbf{z}$ is near a given density p_0, $I_F(\mathbf{w}_i^T\mathbf{z})$ can be approximated by

$$I_F(\mathbf{w}_i^T\mathbf{z}) = -E\{2(\log p_0)''(\mathbf{w}_i^T\mathbf{z}) + [(\log p_0)'(\mathbf{w}_i^T\mathbf{z})]^2\} + o(p - p_0).$$
$$= -E[2\frac{p_0''(\mathbf{w}_i^T\mathbf{z})}{p_0(\mathbf{w}_i^T\mathbf{z})} - (\frac{p_0'(\mathbf{w}_i^T\mathbf{z})}{p_0(\mathbf{w}_i^T\mathbf{z})})^2] + o(p - p_0). \quad (37)$$

For example, in the vicinity of the standardized Laplace distribution, we obtain

$$I_F(\mathbf{w}_i^T\mathbf{z}) \approx 4\sqrt{2}\, p_{\mathbf{w}_i^T\mathbf{z}}(0) - 2. \quad (38)$$

In practice, the probability at zero needed in (38) can be estimated, e.g., by a gaussian kernel. Thus the estimation of the optimal $\mathbf{W}$ becomes

$$\boxed{\mathbf{W}_{opt} = \arg\max_{\mathbf{W}} \sum_{i=1}^{n} E\{\exp(-\frac{(\mathbf{w}_i^T\mathbf{z})^2}{d^2})\}} \quad (39)$$

where $\mathbf{W}$ is constrained to be orthogonal, and d is the kernel width.

3.4 Algorithm for finding the sparse coding transform

Next we must choose a practical method to implement the optimization of (39). Of course, in some cases this step can be omitted, and one can use a well-known basis that gives sparse components. For example, the wavelet bases are known to have this property for certain kinds of data (Donoho et al., 1995; Olshausen and Field, 1996; Bell and Sejnowski, 1997).

We give here a (stochastic) gradient descent for the objective function in (39). Using the bigradient feedback (Karhunen et al., 1997b; Hyvärinen, 1997b), we obtain the following learning rule for $\mathbf{W}$:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mu(k)q(\mathbf{W}(k)\mathbf{z}(k))\mathbf{z}(k)^T + \frac{1}{2}(\mathbf{I} - \mathbf{W}(k)\mathbf{W}(k)^T)\mathbf{W}(k) \quad (40)$$

where $\mu(k)$ is the learning rate sequence, and the nonlinearity $q(u) = -u\exp(-u^2/d^2)$ is applied separately on every component of the vector $\mathbf{W}(k)\mathbf{z}(k)$, with d being a kernel width. The learning rule is very similar[3] to some of the ICA learning rules derived in (Hyvärinen, 1997b); indeed, if the data is preprocessed by whitening, the learning rule in (40) is a special case of the learning rules in (Hyvärinen, 1997b).

[3] Note that we use the notation $\mathbf{s} = \mathbf{W}\mathbf{x}$, whereas in (Karhunen et al., 1997b; Hyvärinen, 1997b), the notation $\mathbf{s} = \mathbf{W}^T\mathbf{x}$ is used.

3.5 Modifications for image denoising

In image denoising, the above results need to be slightly modified. These modifications are necessary because of the well-known fact that ordinary mean-square error is a rather inadequate measure of errors in images. Perceptually more adequate measures can be obtained e.g. by weighting the mean-square error so that components corresponding to lower frequencies have more weight. Since the variance of the sparse and principal components is larger for lower frequencies, such a perceptually motivated weighting can be approximated simply by the following objective function

$$J = \sum_{i=1}^{n} E\{(\mathbf{w}^T\mathbf{z})^2\} I_F(\mathbf{w}_i^T\mathbf{z}). \tag{41}$$

Using (16), this can be expressed as

$$J = \sum_{i=1}^{n} I_F(\frac{\mathbf{w}_i^T\mathbf{z}}{\sqrt{E\{(\mathbf{w}_i^T\mathbf{z})^2\}}}). \tag{42}$$

This is the normalized Fisher information, which is a scale-invariant measure of nongaussianity.

To maximize J, one could derive a gradient algorithm that would be similar to (40). Instead, we give here a very fast algorithm that requires some additional approximations, but which we have empirically found to work well with image data. This consists of first finding a matrix $\mathbf{W}_0$ that decomposes the data $\mathbf{z}$ into independent components as $\mathbf{s} = \mathbf{W}_0\mathbf{z}$. Any algorithm for independent component analysis (Amari et al., 1996; Bell and Sejnowski, 1995; Cardoso and Laheld, 1996; Comon, 1994; Hyvärinen and Oja, 1997) can be used for this purpose. Using ICA algorithms is justified by the fact that maximizing J under the constraint of decorrelation of the $\mathbf{w}_i^T\mathbf{z}$ is one way of to estimating the ICA data model; for the approximation in (39), this has been proven (Hyvärinen, 1997b). Thus the difference between ICA and the maximization of J is only a question of different constraints .

After estimating the ICA decomposition matrix $\mathbf{W}_0$, we transform it by

$$\mathbf{W} = \mathbf{W}_0(\mathbf{W}_0^T\mathbf{W}_0)^{-1/2} \tag{43}$$

to obtain an orthogonal transformation matrix. The utility of this method resides in the fact that there exist algorithms for ICA that are computationally highly efficient (Hyvärinen, 1997a; Hyvärinen and Oja, 1997). Therefore, the above procedure enables one to estimate the basis even for data sets of high dimensions. Empirically, we have found that the required approximations do not significantly deteriorate the statistical properties of the obtained sparse coding transformation.

4 Sparse Code Shrinkage

Now we summarize the algorithm of sparse code shrinkage as developed in the preceding sections. In this method, the ML noise reduction is applied on sparse components, first choosing the orthogonal transformation so as to maximize the sparseness of the components. This restriction to sparse variables is justified by the fact that in many applications, such as image processing, the distributions encountered are sparse. The algorithm is as follows:

1. Using a representative noise-free set of data $\mathbf{z}$ that has the same statistical properties as the n-dimensional data $\mathbf{x}$ that we want to denoise, estimate the sparse coding transformation $\mathbf{W} = \mathbf{W}_{opt}$ as explained in Sections 3.4–3.5.

2. For every $i = 1, ..., n$, estimate a density model for $s_i = \mathbf{w}_i^T \mathbf{z}$, using the models described in Section 2.4.1. Choose by (30) whether model (9) or (11) is to be used for s_i. Estimate the relevant parameters e.g. by (28) or (29), respectively. Denote by g_i the corresponding shrinkage function, given by (10) or by (12), respectively.

3. Observing $\tilde{\mathbf{x}}(t), t = 1, ..., T$, which are samples of a noisy version of $\mathbf{x}$ as in (33), compute the projections on the sparsifying basis:
$$\mathbf{y}(t) = \mathbf{W}\tilde{\mathbf{x}}(t). \tag{44}$$

4. Apply the shrinkage operator g_i corresponding to the density model of s_i on every component $y_i(t)$ of $\mathbf{y}(t)$, for every t, obtaining
$$\hat{s}_i(t) = g_i(y_i(t)); \tag{45}$$
where σ^2 is the noise variance (see below on estimating σ^2).

5. Transform back to original variables to obtain estimates of the noise-free data $\mathbf{x}(t)$:
$$\hat{\mathbf{x}}(t) = \mathbf{W}^T \hat{\mathbf{s}}(t). \tag{46}$$

If the noise variance σ^2 is not known, one might estimate it, following (Donoho et al., 1995), by multiplying by 0.6475 the mean absolute deviation of the y_i corresponding to the very sparsest s_i.

5 Discussion

5.1 Comparison with wavelet and coring methods

The resulting algorithm of sparse code shrinkage is closely related to wavelet shrinkage (Donoho et al., 1995), with the following differences:

1. Our method assumes that one first estimates the orthogonal basis using noise-free training data that has similar statistical properties. Thus our method could be considered as a principled method of choosing the wavelet basis for a given class of data: instead being limited to bases that have certain abstract mathematical properties (like self-similarity), we let the basis be determined by the data alone, under the sole constraint of orthogonality.

2. In sparse code shrinkage, the shrinkage nonlinearities are estimated separately for each component, using the same training data as for the basis. In wavelet shrinkage, the form of shrinkage nonlinearity is fixed, and the shrinkage coefficients are either constant for most of the components (and perhaps set to zero for certain components), or constant for each resolution level (Donoho et al., 1995). (More complex methods like cross-validation (Nason, 1996) are possible, though.) This difference stems from the fact that wavelet shrinkage uses minimax estimation theory, whereas our method uses ordinary ML estimation. Note that point 2 is conceptually independent from point 1, and further shows the adaptive nature of sparse code shrinkage.

3. Our method, though primarily intended for sparse data, could be directly modified to work for other kinds of nongaussian data.

4. An advantage of wavelet methods is that very fast algorithms have been developed to perform the transformation (Mallat, 1989), avoiding multiplication of the data by the matrix $\mathbf{W}$ (or its transpose).

5. Of course, wavelet methods avoid the computational overhead, and especially the need for additional, noise-free data required for estimating the matrix $\mathbf{W}$ in the first place. The requirement for noise-free training data is, however, not an essential part of our method. Future research will probably provide methods that enable the estimation of the sparsifying matrix $\mathbf{W}$ and the shrinkage nonlinearities even from noisy data.

The connection is especially clear if one assumes that both steps 1 and 2 of sparse code shrinkage in Section 4 are omitted, using a wavelet basis and the shrinkage function (10) with $a_i = 0$ and a b_i that is equal for all i (except perhaps some i for which it is zero). Such a method would be essentially equivalent to wavelet shrinkage.

A related method is Bayesian wavelet coring, introduced by Simoncelli and Adelson (1996) . In Bayesian wavelet coring, the shrinkage nonlinearity is estimated from the data to minimize mean-square error. Thus the method is more adaptive than wavelet shrinkage, but still uses a predetermined sparsifying transformation.

5.2 Connection to independent component analysis

Let us consider the estimation of the generative data model of independent component analysis (ICA) in the presence of noise. The noisy version of the conventional ICA model is given by

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \boldsymbol{\nu} \tag{47}$$

where the latent variables s_i are assumed to be independent and nongaussian (usually supergaussian), $\mathbf{A}$ is a constant mixing matrix, and $\boldsymbol{\nu}$ is a gaussian noise vector. Now, a reasonable method for denoising $\mathbf{x}$ would be to somehow find estimates $\hat{s}_i$ of the (noise-free) independent components, and then reconstruct $\mathbf{x}$ as $\hat{\mathbf{x}} = \hat{\mathbf{A}}\hat{\mathbf{s}}$. Such a method (Lewicki and Olshausen, 1998) is closely related to sparse code shrinkage. In (Hyvärinen, 1998) it was proven that if the covariance matrix of the noise and the mixing matrix fulfill a certain relation, the estimate $\hat{\mathbf{s}}$ can be obtained by applying a shrinkage nonlinearity on the components of $\hat{\mathbf{A}}^{-1}\mathbf{x}$. This relation is fulfilled, e.g. if $\mathbf{A}$ is orthogonal, and noise covariance is proportional to identity, and is thus true for the noise covariance and the transformation matrix $\mathbf{W}$ in sparse code shrinkage. Thus our method can be considered as a computationally efficient approximation of the estimation of the noisy ICA model, consisting of replacing the constraint of independence of the sparse components by the constraint of the orthogonality of the sparsifying matrix. Without this simplification, the computation of the sparse components would require an optimization procedure (gradient descent or a linear program) for every sample point (Hyvärinen, 1998; Lewicki and Olshausen, 1998).

6 Simulation results

6.1 Maximum likelihood estimation in one dimension

First we did simulations to illustrate the capability of the ML estimation to reduce gaussian noise in scalar nongaussian random variables. The mean-square error of the nonlinear ML estimator in (5) was compared to the mean-square error of the optimal (MMS) linear estimator using the index R_s defined in (20). This index shows how much the mean-square error was decreased by taking into account the nonlinear nature of the ML estimator.

Fig. 3 shows the estimated index for a Laplace random variable with different noise variances (the Laplace variable had unit variance). For small noise variances, the index increases in line with Theorem 1 and its corollary. The maximum attained is approximately 2%. After the maximum, the index starts decreasing. This decrease is not predicted by Theorem 1, which is valid for small noise levels only.

In Fig. 4, the same results are shown for a very supergaussian random variable, obtained by taking the cube of a gaussian variable. The optimal estimator was approximated using the method of Section 2.4.1, using the density in (11). Due to the strong nongaussianity of s, noise reductions of 30% are possible. The qualitative behavior was rather similar to Figure 3.

Next we illustrated how the ratio changes with increasing nongaussianity. We took a family of nongaussian variables defined as powers of gaussian variables:

$$s = \frac{\operatorname{sign}(v)|v|^\beta}{\sqrt{E\{|v|^{2\beta}\}}} \tag{48}$$

where v is a standardized gaussian random variable, and the division by the denominator is done to normalize s to unit variance. The parameter $\beta > 1$ controls the sparseness of the distribution; sparseness increases with increasing β. The density model used was chosen for each value of β according to (30). The ratio R_s, for different values of β, is plotted in Figure 5. This shows clearly how the denoising capability increases with increasing sparsity.

6.2 Experiments on image data

Here we present some examples of applications of sparse code shrinkage to image data. More detailed experiments will be described in (Hyvärinen et al., 1998a).

6.2.1 Data

The data consisted of 10 real-life images, mainly natural scenes, not unlike those used by other researchers (Olshausen and Field, 1996; Karhunen et al., 1997a). Most of the images were obtained directly from PhotoCDs, thus avoiding artifacts created by any supplementary processing. Two examples are given in Fig. 6.

The images were randomly divided into two sets. The first set was used for learning of the weight matrix $\mathbf{W}$ that gives the sparse coding transformation, as well as for estimating the shrinkage nonlinearities. The second set was used as a test set. It was artificially corrupted by Gaussian noise, and the sparse code shrinkage method in Section 4 was used to reduce the noise.

6.2.2 Methods

The images were used in the method in the form of subwindows of 8×8 pixels. Such windows were presented as 64-dimensional vectors of gray-scale values. The DC value, i.e., the mean of the gray-scale values, was subtracted from each vector as a preprocessing step. This resulted in a linear dependency between the components of the observed data, and therefore the dimensionality of the data was reduced by one dimension, using PCA to get rid of the component of zero variance. Thus one obtained the vectors $\mathbf{x}(t)$ used in the algorithm. In the results shown below, an inverse of these preprocessing steps was performed after the main algorithm.

After preprocessing, the sparse code shrinkage algorithm, as described in Section 4 was applied to the noisy images. The sparse code transformation $\mathbf{W}$ was computed by first using the fast fixed-point algorithm for ICA (Hyvärinen and Oja, 1997; Hyvärinen, 1997a), and then transforming as in (43). The obtained transformation matrix was qualitatively similar to the ICA or sparse coding matrices as estimated in (Bell and Sejnowski, 1997; Karhunen et al., 1997a; Olshausen and Field, 1996), for example. The variance of the noise was assumed to be known. The densities encountered were all modelled by (11), due to their strong sparsities.

6.2.3 Results

The results are shown for the two images depicted in Fig. 6. In Fig. 7, a first series of results is shown. An image which was artificially corrupted with Gaussian noise with standard deviation 0.5 (the standard deviations of the original images were normalized to 1), is shown in the upper left-hand corner. The result of applying our denoising method on that image is shown in the upper right-hand corner. For comparison, the corresponding denoising result using Wiener filtering is depicted in the lower row. Wiener filtering is in fact a special case of our framework, obtained when the distributions of the components are assumed to be all gaussian.

Visual comparison of the images in Fig. 7 shows that our sparse code shrinkage method cancels noise quite effectively. In comparison to Wiener (low-pass) filtering and related methods, one sees that contours and other sharp details are conserved better, while the overall reduction of noise is much stronger. This result is in line with those obtained by wavelet shrinkage (Donoho et al., 1995) and Bayesian wavelet coring (Simoncelli and Adelson, 1996).

The second experiment in Fig. 8 shows the corresponding results for a different image. The results are essentially similar to those of the first experiment.

In Figs. 9 and 10, corresponding results for a higher noise level (noise variance =1) are shown. In the presence of such a strong noise, the performance of the method cannot be expected to be very satisfactory. Nevertheless, comparison with the depicted Wiener filtering results show that at least the method reduced noise much better than Wiener filtering. It could be argued, though, that the image is too distorted for the results to be useful; the validity of such considerations depends on the practical application situation.

7 Conclusion

We derived the method of sparse code shrinkage using maximum likelihood estimation of nongaussian random variables corrupted by gaussian noise. In the method, we first determine an orthogonal basis in which the components of given multivariate data have the sparsest distributions possible. The sparseness of the components is utilized in ML estimation of the noise-free components; these estimates are then used to reconstruct the original noise-free data by inverting the transformation. In the general case, it was shown that the noise reduction is proportional to the sum of the Fisher informations of the sparse components (for small noise levels). Sparse code shrinkage is closely connected to wavelet shrinkage; in fact, it can be considered as a principled way of choosing the orthogonal wavelet-like basis based on data alone, as well as an alternative way of choosing the shrinkage nonlinearities.

Acknowledgement

The author is grateful to Mr. Patrik Hoyer for performing the experiments with image data, and to Prof. Erkki Oja for helpful comments.

References

Amari, S., Cichocki, A., and Yang, H. (1996). A new learning algorithm for blind source separation. In *Advances in Neural Information Processing Systems 8*, pages 757–763. MIT Press, Cambridge, MA.

Barlow, H. (1994). What is the computational goal of the neocortex ? In Koch, C. and Davis, J., editors, *Large-scale neuronal theories of the brain.* MIT Press, Cambridge, MA.

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

Bell, A. and Sejnowski, T. (1997). The 'independent components' of natural scenes are edge filters. *Vision Research*, 37:3327–3338.

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

Comon, P. (1994). Independent component analysis – a new concept? *Signal Processing*, 36:287–314.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory.* John Wiley & Sons.

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

Efron, B. and Morris, C. (1975). Data analysis using Stein's estimator and its generalizations. *J. of the American Statistical Association*, 70:311–319.

Field, D. (1994). What is the goal of sensory coding? *Neural Computation*, 6:559–601.

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

Hyvärinen, A. (1997a). A family of fixed-point algorithms for independent component analysis. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97)*, pages 3917–3920, Munich, Germany.

Hyvärinen, A. (1997b). Independent component analysis by minimization of mutual information. Technical Report A46, Helsinki University of Technology, Laboratory of Computer and Information Science.

Hyvärinen, A. (1998). Independent component analysis in the presence of gaussian noise by maximizing joint likelihood. *Neurocomputing*, 22:49–67.

Hyvärinen, A., Hoyer, P., and Oja, E. (1998a). Applications of sparse code shrinkage to image denoising. Technical report, Helsinki University of Technology, Laboratory of Computer and Information Science. Obsolete!

Hyvärinen, A., Hoyer, P., and Oja, E. (1998b). Sparse code shrinkage for image denoising. In *Proc. IEEE Int. Joint Conf. on Neural Networks*, pages 859–864, Anchorage, Alaska.

Hyvärinen, A. and Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492.

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

Karhunen, J., Hyvärinen, A., Vigario, R., Hurri, J., and Oja, E. (1997a). Applications of neural blind separation to signal and image processing. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97)*, pages 131–134, Munich, Germany.

Karhunen, J., Oja, E., Wang, L., Vigario, R., and Joutsensalo, J. (1997b). A class of neural networks for independent component analysis. *IEEE Trans. on Neural Networks*, 8(3):486–504.

Kendall, M. and Stuart, A. (1958). *The Advanced Theory of Statistics.* Charles Griffin & Company.

Lewicki, M. and Olshausen, B. (1998). Inferring sparse, overcomplete image codes using an efficient coding framework. In *Advances in Neural Information Processing Systems 10 (Proc. NIPS*97)*, pages 815–821. MIT Press.

Mallat, S. G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on PAMI*, 11:674–693.

Nason, G. P. (1996). Wavelet shrinkage using cross-validation. *Journal of the Royal Statistical Society, Series B*, 58:463–479.

Oja, E. (1997). The nonlinear PCA learning rule in independent component analysis. *Neurocomputing*, 17(1):25–46.

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

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325.

Pham, D.-T., Garrat, P., and Jutten, C. (1992). Separation of a mixture of independent sources through a maximum likelihood approach. In *Proc. EUSIPCO*, pages 771–774.

Schervish, M. (1995). *Theory of Statistics.* Springer.

Simoncelli, E. P. and Adelson, E. H. (1996). Noise removal via bayesian wavelet coring. In *Proc. Third IEEE International Conference on Image Processing*, pages 379–382, Lausanne, Switzerland.

A Proof of Theorems 1 and 2

We prove here directly the vector case, i.e. Theorem 2. Theorem 1 is just a special case.

From (5) we have

$$\hat{\mathbf{s}} = \mathbf{x} - \sigma^2 \nabla f(\mathbf{x}) + O(\sigma^4) \tag{49}$$

where ∇f is the gradient of the density $\mathbf{f}$. Thus we obtain

$$\hat{\mathbf{s}} - \mathbf{s} = \boldsymbol{\nu} - \sigma^2 \nabla f(\mathbf{x}) + O(\sigma^4) = \boldsymbol{\nu} - \sigma^2 [\nabla f(\mathbf{s}) + \nabla^2 f(\mathbf{s})\boldsymbol{\nu}] + O(\sigma^4) \tag{50}$$

and

$$\begin{aligned} &E\{(\hat{\mathbf{s}} - \mathbf{s})(\hat{\mathbf{s}} - \mathbf{s})^T\} \\ &\quad = E\{\boldsymbol{\nu}\boldsymbol{\nu}^T\} + \sigma^4 E\{\nabla f(\mathbf{s}) \nabla f(\mathbf{s})^T\} - 2\sigma^2 E\{\boldsymbol{\nu}\boldsymbol{\nu}^T\} E\{\nabla^2 f(\mathbf{s})\} + o(\sigma^4) \\ &\quad = \sigma^2 \mathbf{I} - \sigma^4 I_F(\mathbf{s}) + o(\sigma^4) \end{aligned} \tag{51}$$

where we have used the property (Schervish, 1995)

$$E\{\nabla^2 f(\mathbf{s})\} = I_F(\mathbf{s}) \tag{52}$$

B Proof of (28) and (32)

The estimators in (28) are obtained as a special case of the estimators (32), so we only prove (32) in the following.

In (Pham et al., 1992) it was shown that for any function r, the inner product of r with the score function f' with respect to the metric defined by p is obtained as:

$$< f', r >= \int p(\xi) f'(\xi) r(\xi) = E\{r'(s)\} \tag{53}$$

which has the benefit that it can be simply estimated as the corresponding sample average. Using (53), we obtain the inner products, denoting by i the identity function:

$$< f', i >= 1, \quad < f', h >= E\{h'(s)\}, \tag{54}$$

$$< i, h >= E\{sh(s)\}, \qquad < i, i >= E\{s^2\}, \tag{55}$$

$$< h, h >= E\{h(s)^2\}. \tag{56}$$

Now we can compute a function h_2 that is orthogonal to i:

$$h_2(\xi) = h(\xi) - \frac{E\{sh(s)\}}{E\{s^2\}}\xi \tag{57}$$

with

$$< h_2, h_2 >= E\{h(s)^2\} - \frac{[E\{sh(s)\}]^2}{E\{s^2\}}. \tag{58}$$

Projecting f' on i and h_2, we obtain finally

$$\begin{aligned} f'(\xi) &\approx \frac{1}{E\{s^2\}}\xi + \frac{1}{< h_2, h_2 >}[E\{h'(s)\} - \frac{E\{sh(s)\}}{E\{s^2\}}][h(\xi) - \frac{E\{sh(s)\}}{E\{s^2\}}\xi] \\ &= \frac{1}{E\{s^2\}}[1 - \frac{E\{sh(s)\}}{< h_2, h_2 >}[E\{h'(s)\} - \frac{E\{sh(s)\}}{E\{s^2\}}]]\xi \\ &\quad + \frac{1}{< h_2, h_2 >}[E\{h'(s)\} - \frac{E\{sh(s)\}}{E\{s^2\}}]h(\xi) \end{aligned} \tag{59}$$

which gives (32).

C Proof of (36)

Using the orthogonal decomposition in Section B, in particular Eq. (59), one obtains:

$$\begin{aligned} \int p(\xi)[f'(\xi)]^2 &\approx a^2 \int f(\xi)\xi^2 d\xi + b^2 \int f(\xi)h(\xi)^2 d\xi + 2ab \int f(\xi)\xi h(\xi) d\xi \\ &= \frac{1}{E\{s^2\}} + \frac{1}{< h_2, h_2 >}[E\{h'(s)\} - \frac{E\{sh(s)\}}{E\{s^2\}}]^2 \\ &= \frac{1}{E\{s^2\}}[1 + \frac{[E\{h'(s)\}E\{s^2\} - E\{sh(s)\}]^2}{E\{h(s)^2\}E\{s^2\} - [E\{sh(s)\}]^2}]. \end{aligned} \tag{60}$$

D Proof of (37)

Denote $p_\epsilon = p - p_0$. Assume that terms of order $o(p'_\epsilon)$ are of order $o(p_\epsilon)$; in other words, we are considering a Sobolev neighbourhood of p_0. We obtain

$$\int p(\xi)(\frac{p'(\xi)}{p(\xi)})^2 d\xi = \int p(\xi)\frac{p'_0(\xi)^2 + 2p'_0(\xi)p'_\epsilon(\xi) + o(p_\epsilon)}{p_0(\xi)^2 + 2p_0(\xi)p_\epsilon(\xi) + o(p_\epsilon)} d\xi$$

$$= \int p(\xi)[\frac{p'_0(\xi)^2}{p_0(\xi)^2} + 2\frac{p'_\epsilon(\xi)p'_0(\xi)}{p_0(\xi)^2} - 2\frac{p_\epsilon(\xi)p'_0(\xi)^2}{p_0(\xi)^3}]d\xi + o(p_\epsilon)$$

$$= \int p(\xi)(\frac{p'_0(\xi)}{p_0(\xi)})^2 d\xi + 2\int \frac{p'_0(\xi)}{p_0(\xi)} p'_\epsilon(\xi) d\xi - 2\int \frac{p'_0(\xi)^2}{p_0(\xi)^2} p_\epsilon(\xi) d\xi + o(p_\epsilon). \quad (61)$$

Using partial integration the second term can be modified:

$$\int (\log p_0(\xi))' p'_\epsilon(\xi) d\xi = -\int (\log p_0(\xi))'' p_\epsilon(\xi) d\xi. \quad (62)$$

On the other hand,

$$\int (\log p_0(\xi))'' p_0(\xi) d\xi + \int [(\log p_0(\xi))']^2 p_0(\xi) d\xi = 0. \quad (63)$$

Thus we obtain

$$\int p(\xi)(\frac{p'(\xi)}{p(\xi)})^2 d\xi$$

$$= \int p(\xi)(\frac{p'_0(\xi)}{p_0(\xi)})^2 d\xi - 2\int p(\xi)(\log p_0(\xi))'' d\xi - 2\int p(\xi)\frac{p'_0(\xi)^2}{p_0(\xi)^2} d\xi + o(p_\epsilon)$$

$$= \int p(\xi)[-((\log p_0)'(\xi))^2 - 2(\log p_0)''(\xi)]d\xi + o(p_\epsilon). \quad (64)$$

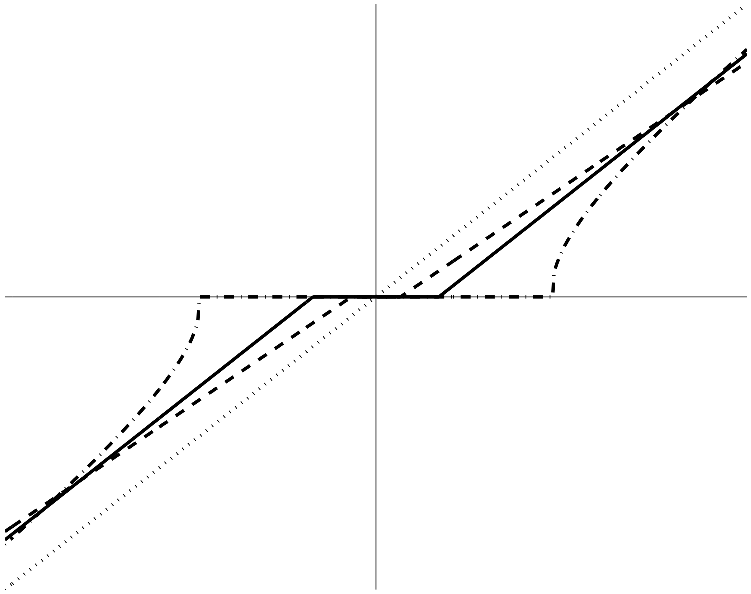

Figure 1: Plots of the shrinkage functions. The effect of the functions is to reduce the absolute value of its argument by a certain amount which depends on the noise level. Small arguments are set to zero. This reduces gaussian noise for sparse random variables. Solid line: shrinkage corresponding to Laplace density as in (7). Dashed line: typical shrinkage function obtained from (10). Dash-dotted line: typical shrinkage function obtained from (12). For comparison, the line $x = y$ is given by dotted line. All the densities were normalized to unit variance, and noise variance was fixed to .3.

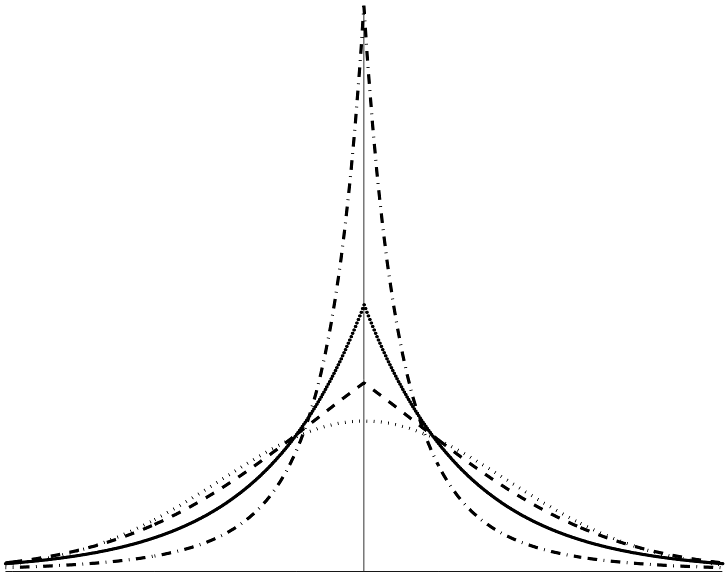

Figure 2: Plots of densities corresponding to models (9) and (11) of the sparse components. Solid line: Laplace density. Dashed line: a typical moderately supergaussian density given by (9). Dash-dotted line: a typical strongly supergaussian density given by (11). For comparison, gaussian density is given by dotted line.

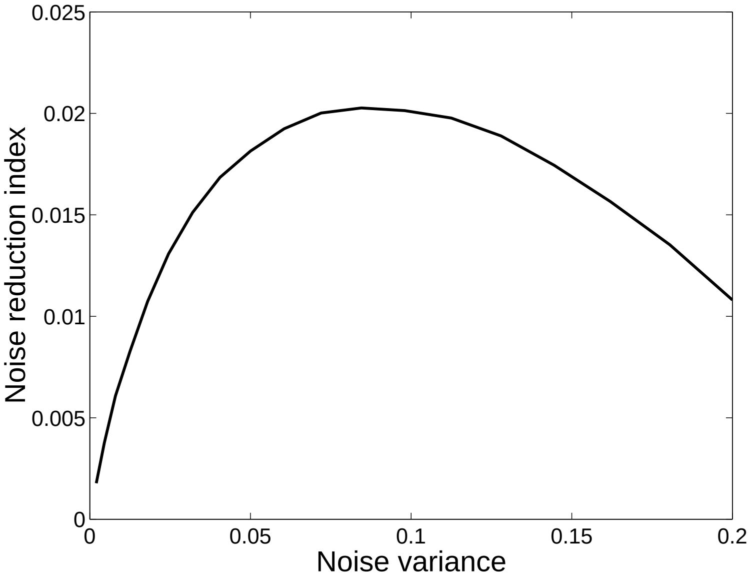

Figure 3: Illustration of the denoising capability of ML estimation in one dimension. The index of noise reduction R_s is plotted for a Laplace random variable of unit variance, for different values of noise variance σ^2.

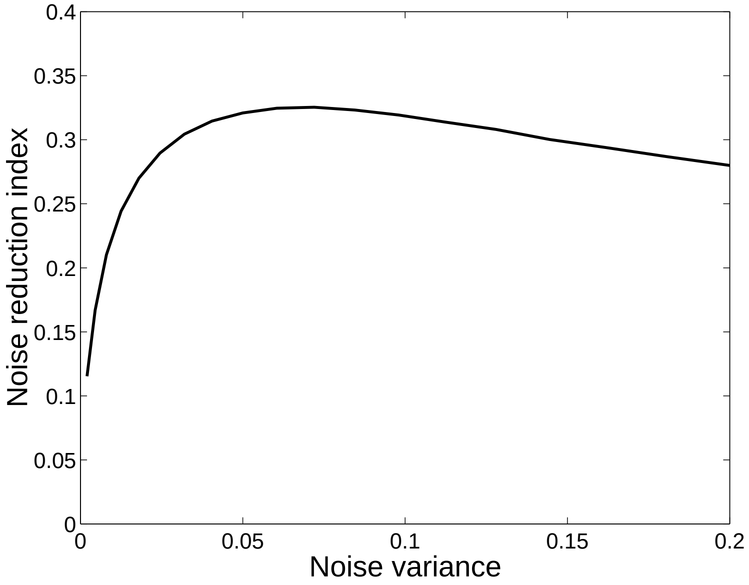

Figure 4: Illustration of the denoising capability of ML estimation in one dimension. The index of noise reduction R_s is plotted for a highly supergaussian random variable of unit variance, for different values of noise variance σ^2.

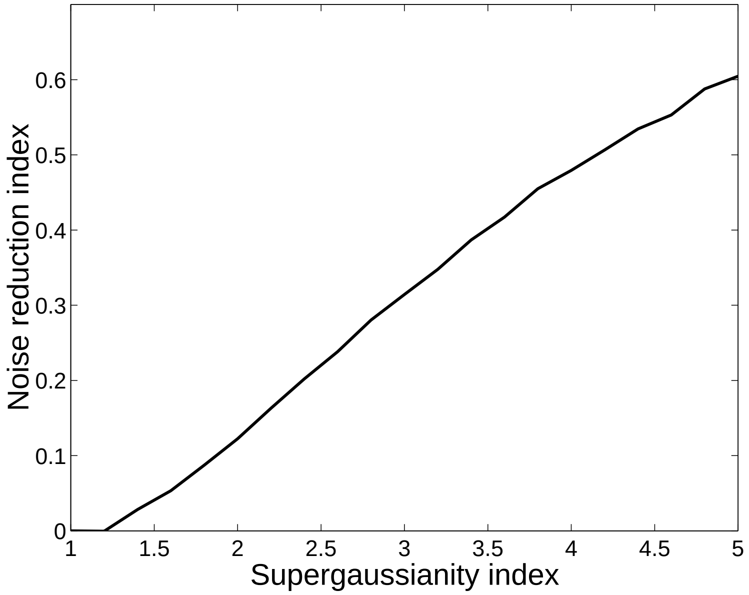

Figure 5: The denoising capability of ML estimation depends on nongaussianity. The index of noise reduction R_s is plotted for different supergaussian random variables of unit variance, parameterized by β as in (48). Noise variance $\sigma^2 = 0.2$ was constant. Supergaussianity increases with the value of the parameter β, and so does R_s.

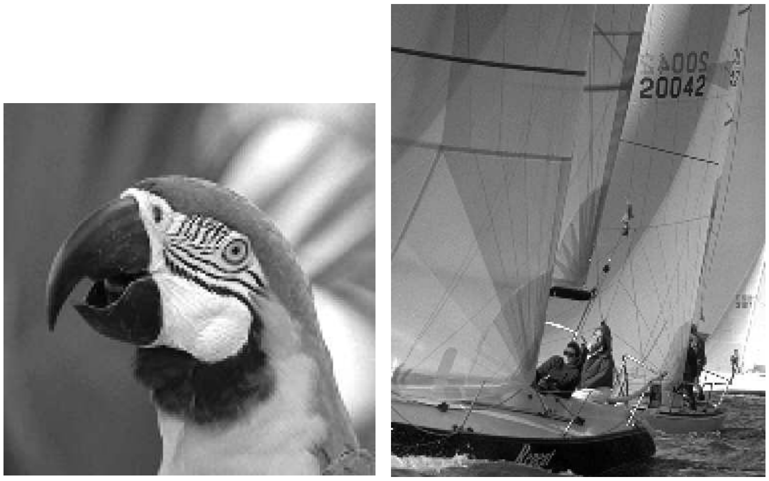

Figure 6: Two of the images used in the experiments.

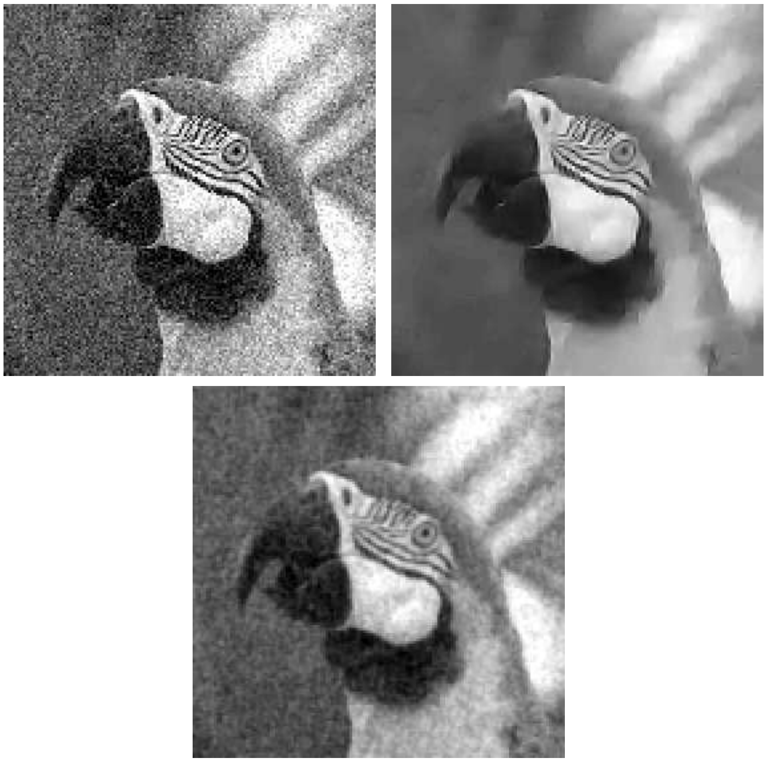

Figure 7: The first experiment in image denoising. Upper left: original image corrupted with noise. Upper right: the recovered image after applying sparse code shrinkage. Below: for comparison, the same image Wiener-filtered.

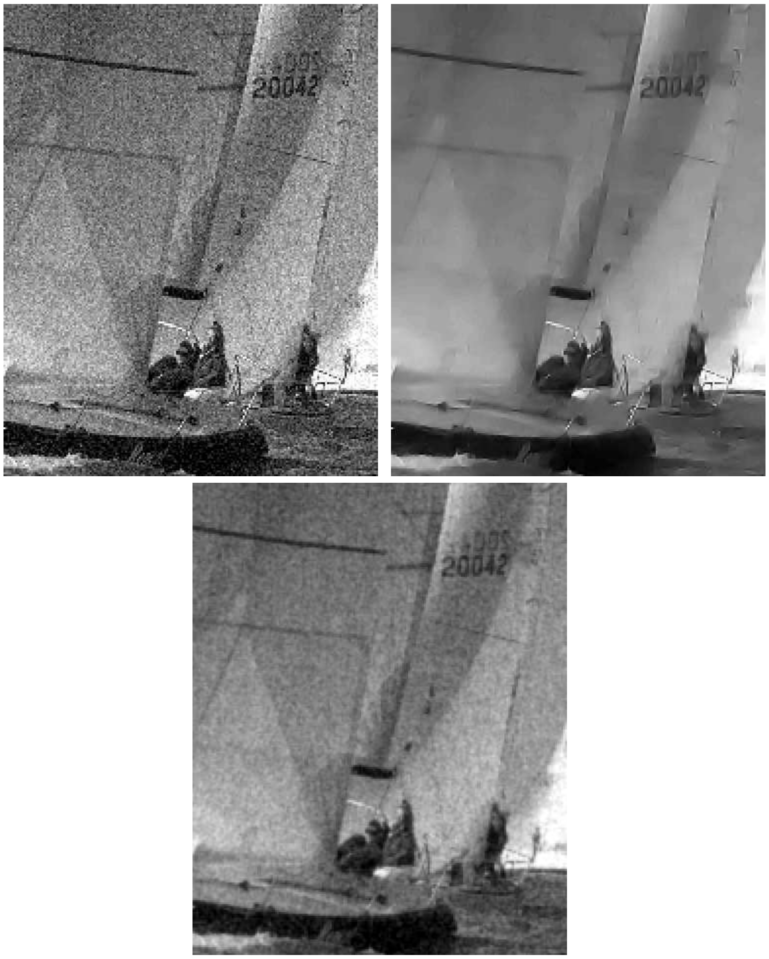

Figure 8: The second experiment in image denoising. Upper left: original image corrupted with noise. Upper right: the recovered image after applying sparse code shrinkage. Below: for comparison, the same image Wiener-filtered.

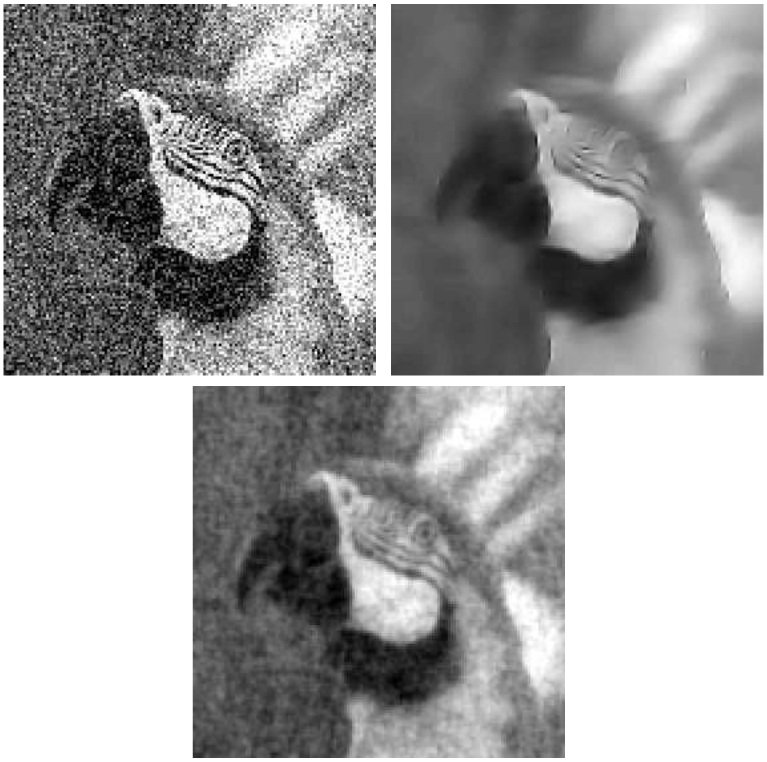

Figure 9: The third experiment in denoising, with a higher noise level than above. Upper left: original image corrupted with noise. Upper right: the recovered image after applying sparse code shrinkage. Below: for comparison, the same image Wiener-filtered.

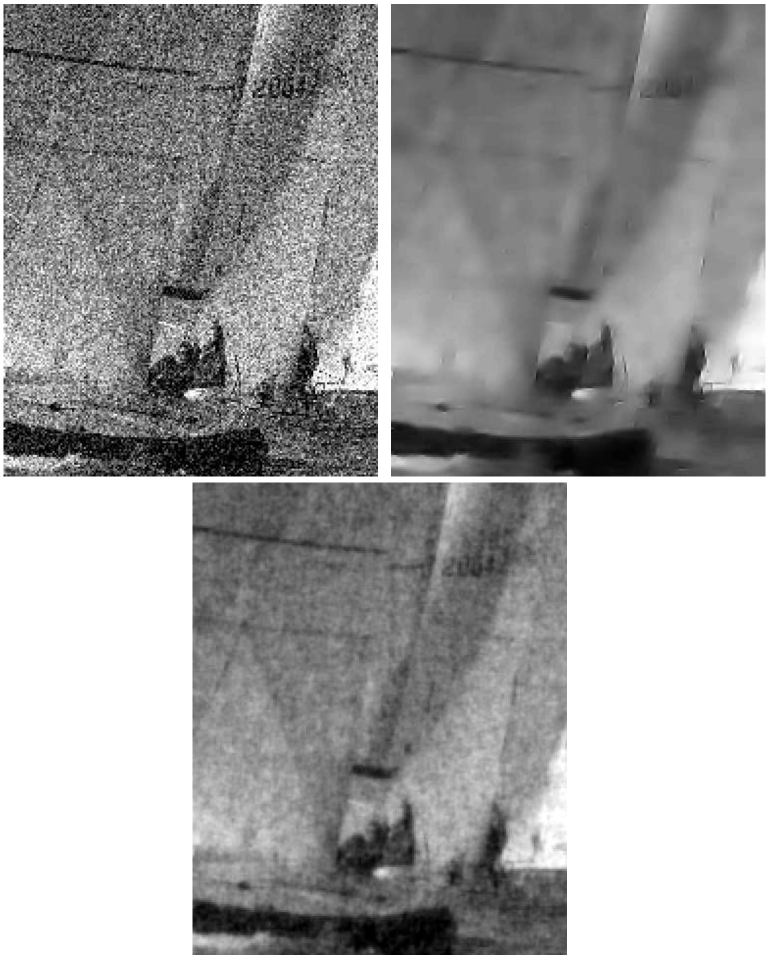

Figure 10: The fourth experiment in denoising, with a higher noise level than above. Upper left: original image corrupted with noise. Upper right: the recovered image after applying sparse code shrinkage. Below: for comparison, the same image Wiener-filtered.