A NEW APPROACH FOR SPARSE DECOMPOSITION AND SPARSE SOURCE SEPARATION

Arash Ali AMINI[1], *Massoud BABAIE-ZADEH*[1], *and Christian JUTTEN*[2]

[1] Electrical engineering department, Sharif University of Technology, Tehran, Iran.
[2] Laboratoire des Images et des Signaux (LIS), Institut National Polytechnique de Grenoble (INPG), Grenoble, France.
aaamini57@yahoo.com, mbzadeh@sharif.edu, Christian.Jutten@inpg.fr

ABSTRACT

We introduce a new approach for sparse decomposition, based on a geometrical interpretation of sparsity. By sparse decomposition we mean finding sufficiently sparse solutions of underdetermined linear systems of equations. This will be discussed in the context of Blind Source Separation (BSS). Our problem is then underdetermined BSS where there are fewer mixtures than sources. The proposed algorithm is based on minimizing a family of quadratic forms, each measuring the distance of the solution set of the system to one of the coordinate subspaces (i.e. coordinate axes, planes, etc.). The performance of the method is then compared to the minimal 1-norm solution, obtained using the linear programming (LP). It is observed that the proposed algorithm, in its simplest form, performs nearly as well as LP, provided that the average number of active sources at each time instant is less than unity. The computational efficiency of this simple form is much higher than LP. For less sparse sources, performance gains over LP may be obtained at the cost of increased complexity which will slow the algorithm at higher dimensions. This suggests that LP is still the algorithm of choice for high-dimensional moderately-sparse problems. The advantage of our algorithm is to provide a trade-of between complexity and performance.

1. INTRODUCTION

Sparse decomposition which may be viewed as an attempt to uniquely identify a *relevant* solution of an underdetermined system of linear equations, has become a subject of interest in recent years. It is well-known that an underdetermined system (if consistent) has infinitely many solutions, and additional constraints should be imposed if we are to arrive at a unique solution. Sparsity is one such condition which is usually sufficient for the purpose. It also leads to a highly desirable solution from a practical point of view. In fact, in many situations, a sparse solution corresponds to an *efficient* representation of data as a linear combination of some collection of predetermined elements.

To be more specific, we pursue the problem in the context of a concrete example, namely the BSS problem. The objective of BSS is to recover a number of m (unknown) sources form n (known) mixtures when little or no information is available about the nature of sources or mixtures. We will consider the linear (noise-free) model, i.e. $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$, in which $\mathbf{s}(t)$ and $\mathbf{x}(t)$ are respectively, $m \times 1$ and $n \times 1$ vectors representing sources and mixtures, and $\mathbf{A}$ is the $n \times m$ mixing matrix. The number of samples available will be denoted by N, i.e. $t = 1, \ldots, N$. We also assume the so-called *instantaneous* situation in which no time structure is considered and each source sample is recovered only based on the corresponding mixture sample. Thus, the model we actually deal with is

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

Although the identification of the mixing matrix is an essential part of the BSS solution, we shall assume $\mathbf{A}$ to be estimated by other means (refer to [7] and [1] for *clustering*-based methods, and to [2] for a potential-function-based approach). In fact, we will neglect any estimation errors and consider $\mathbf{A}$ to be exactly known a priori. Our main interest is then in cases where $n < m$, i.e. fewer mixtures than sources, where the mere knowledge of $\mathbf{A}$ is not sufficient for source separation. This is where the (added) assumption of sparsity helps in uniquely identifying the sources. Many natural sources exhibit sparsity or may be converted to sparse signals by use of, for example, wavelet-related transforms[1]. We will use the BSS terminology and notation in all subsequent discussions, although the results are relatively context-free.

The sparsity of $\mathbf{s}$ is usually measured by its so-called l^0 norm, i.e. the number of its non-zero elements. Obtaining the sparsest solution by minimizing the l^0 norm is nearly intractable since it requires combinatorial search with a complexity growing exponentially with m. It is also prone to errors when dealing with noisy data. It has been found, first empirically [3] and then theoretically [5],[6] that in almost all cases the problem may be solved by minimizing the l^1 norm instead. In fact, it is shown that for most (large) underdetermined systems, if there is a sufficiently sparse solution, it will be the unique solution of the optimization problem : "minimize $\|\mathbf{s}\|_1$ subject to $\mathbf{x} = \mathbf{A}\mathbf{s}$" – see [4] for a detailed proof. This is a very interesting result, since the solution of the latter may be obtained by efficient linear programming (LP) algorithms.

In a more BSS-oriented approach, the minimal l^1 norm solution may also be obtained as the MAP estimator of source vector under Laplace prior. For the discussion of the minimal l^1 norm solution in the context of BSS, and its performance analysis, consult [2], [7] and [8].

In this paper, we will introduce a method for sparse decomposition based on minimizing cost functions of the form

This work has been partially funded by Sharif University of Technology, by French Embassy in Tehran, and by Center for International Research and Collaboration (ISMO).

[1] In terms of a more common terminology, we may sparsely respresent the signal as a linear combination of *atoms* from a proper *dictionary.* This is motivated by viewing the linear system as $\mathbf{x} = \sum_i \mathbf{a}_i s_i$ where $\{\mathbf{a}_i\}$, the columns of $\mathbf{A}$, are considered atoms or elements defining a signal dictionary.

$\mathbf{s}^{\mathrm{T}}\mathbf{W}\mathbf{s}$ over a family of weight matrices. We will then provide simulation results comparing the performance of the proposed method and that obtained by minimizing l^1 norm (i.e. the LP method).

2. MINIMIZING $\mathbf{s}^{\mathrm{T}}\mathbf{W}\mathbf{s}$ OVER A FAMILY OF WEIGHT MATRICES

2.1 Motivation

The idea behind the method is based on a geometrical interpretation of the sparsity. Consider the elements of the $\mathbf{s}$ vector, denoted by $\{s_i\}$, to be i.i.d. random variables, each being negligible with probability $1-\pi_1$. In other words, $\pi_1 \ll 1$ is the probability that each s_i assumes a considerable value. This simple model may be used to roughly characterize many sparse sources. Examining the sample distribution of $\mathbf{s}$, which is obtained by plotting in the s-space a large number of samples obtained from the source distribution, it appears that the points tend to concentrate first around the origin, then along the coordinate axes, then across the coordinate planes, etc. This is because for the most usual cases, the corresponding probabilities are respectively ordered according to

$$(1-\pi_1)^m > \binom{m}{1}\pi_1(1-\pi_1)^{m-1} > \binom{m}{2}\pi_1^2(1-\pi_1)^{m-2} > \cdots .$$

The situation is depicted in Fig.1 for the case of $m=3$, $n=2$. Also plotted in the figure is the solution set of $\mathbf{x}=\mathbf{A}\mathbf{s}$, for a typical (full rank) $\mathbf{A}$ and a fixed $\mathbf{x}$. In this special case ($m=3, n=2$), for a fixed $\mathbf{x}$, the equation $\mathbf{x}=\mathbf{A}\mathbf{s}$ identifies a line in the 3-dimensional s-space. Consequently, when we are looking for $\mathbf{s}$ for a fixed $\mathbf{x}$, every point on this line satisfies $\mathbf{x}=\mathbf{A}\mathbf{s}$ (however, we are looking for a sparse $\mathbf{s}$). The search for the sparse solution may now be viewed as finding a point on this line which is closest to the (sample) distribution of $\mathbf{s}$. It is intuitively pleasing that by selecting a point (on $\mathbf{x}=\mathbf{A}\mathbf{s}$) which is closest to the coordinate axes (of the s-space), we may reproduce the original source distribution with reasonable accuracy. This means that we preserve salient features of the distribution, while neglecting unimportant details. For example, the method clearly produces erroneous results for points around the origin, but the exact locations of these points are of minor importance.

For the above special case of $m=3$ and $n=2$, the distance of every point to s_1 axis is $s_2^2+s_3^2$, the distance to s_2 axis is $s_1^2+s_3^2$, and the distance to s_3 axis is $s_1^2+s_2^2$. Consequently, for this special case, the criterion for selecting the nearest point to the axes is stated as

$$\underset{all\ i\neq j}{\text{minimize}}\ (s_i^2+s_j^2) \quad \text{subject to} \quad \mathbf{x}=\mathbf{A}\mathbf{s}.$$

A more general objective function, readily applicable in higher dimensions, is $\mathbf{s}^{\mathrm{T}}\mathbf{W}\mathbf{s}$, with $\mathbf{W}$ being an $m\times m$ weight matrix. Our primary interest is in diagonal weight matrices with "zero or one" diagonal entries. For example the family of $m\times m$ diagonal matrices with exactly $m-1$ ones and a single zero on the diagonal, would measure the distances from the coordinate axes. In general we wish to minimize the cost function over an entire family of weight matrices (W), hence the criterion

$$\underset{\mathbf{W}\in W}{\text{minimize}}\ \ \underset{\mathbf{x}=\mathbf{A}\mathbf{s}}{\text{minimize}}\ \mathbf{s}^{\mathrm{T}}\mathbf{W}\mathbf{s}.$$

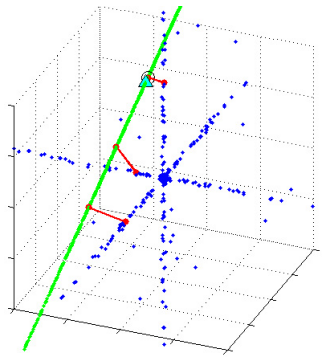

Figure 1: Geometrical interpretation of the method for $m=3$, $n=2$ – The solution set of $\mathbf{x}=\mathbf{A}\mathbf{s}$ is shown as the oblique line. The sparsest solution is obtained as the point on this line which is closest to the coordinate axes (marked by a circle). The minimal l^1 norm solution is also marked with a triangle.

2.2 Solution to the general case

It is well-known that if the system $\mathbf{x}=\mathbf{A}\mathbf{s}$ is consistent, i.e. it has at least one solution, then its minimal l^2 norm solution is given by the so-called pseudo-inverse of $\mathbf{A}$ multiplied by $\mathbf{x}$. In other words, the solution of the optimization problem

$$\text{minimize}\ \mathbf{s}^{\mathrm{T}}\mathbf{s}=\|\mathbf{s}\|_2^2 \quad \text{subject to} \quad \mathbf{x}=\mathbf{A}\mathbf{s} \qquad (1)$$

may be written as $\mathbf{s}_0=\mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{-1}\mathbf{x}$. It is then straightforward to show that if $\mathbf{s}^{\mathrm{T}}\mathbf{s}$ is replaced by $\mathbf{s}^{\mathrm{T}}\mathbf{W}\mathbf{s}$, in which the weight matrix $\mathbf{W}$ is (strictly) *positive definite,* (and hence non-singular) the solution would be $\mathbf{s}_0'=\mathbf{W}^{-1}\mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{W}^{-1}\mathbf{A}^{\mathrm{T}})^{-1}\mathbf{x}$.

As may be inferred from the previous discussion, we are mainly interested in situations where the weight matrix is *singular* (e.g. diagonal matrices with some zeros on its main diagonal), for which the above argument fails. In particular we shall obtain explicit formulas for the case where the weight matrix is diagonal with its first $m-p$ diagonal entries being unity and the rest being zero,

$$\mathbf{W}=\begin{pmatrix}\mathbf{I}_{m-p} & \mathbf{O}\\ \mathbf{O} & \mathbf{O}_p\end{pmatrix}. \qquad (2)$$

Partitioning the $\mathbf{A}$ matrix and the $\mathbf{s}$ vector accordingly, we obtain

$$\mathbf{A}=[\tilde{\mathbf{A}}\quad \hat{\mathbf{A}}], \qquad \mathbf{s}=[\tilde{\mathbf{s}}^{\mathrm{T}}\quad \hat{\mathbf{s}}^{\mathrm{T}}]^{\mathrm{T}}$$

where $\tilde{\mathbf{A}}$, $\hat{\mathbf{A}}$, $\tilde{\mathbf{s}}$, and $\hat{\mathbf{s}}$ are $n\times(m-p)$, $n\times p$, $(m-p)\times 1$, and $p\times 1$, respectively. We impose the following restriction on p : $1\leq p\leq \min\{m-n,n\}$. Now, the optimization problem may be reformulated as

$$\text{minimize}\ \|\tilde{\mathbf{s}}\|_2^2 \quad \text{subject to} \quad \mathbf{x}-\hat{\mathbf{A}}\hat{\mathbf{s}}=\tilde{\mathbf{A}}\tilde{\mathbf{s}}. \qquad (3)$$

Note that the cost function does not depend on $\hat{\mathbf{s}}$. Comparing (3) and (1), we see that the two problems are essentially the same after making the following associations : $\tilde{\mathbf{s}}\leftrightarrow\mathbf{s}$, $(\mathbf{x}-\hat{\mathbf{A}}\hat{\mathbf{s}})\leftrightarrow\mathbf{x}$, $\tilde{\mathbf{A}}\leftrightarrow\mathbf{A}$. The solution for $\tilde{\mathbf{s}}$ of the latter may then be stated, in terms of the parameter $\hat{\mathbf{s}}$, using the usual pseudo-inverse

$$\tilde{\mathbf{s}}_0=\tilde{\mathbf{A}}^{\mathrm{T}}(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^{\mathrm{T}})^{-1}(\mathbf{x}-\hat{\mathbf{A}}\hat{\mathbf{s}}). \qquad (4)$$

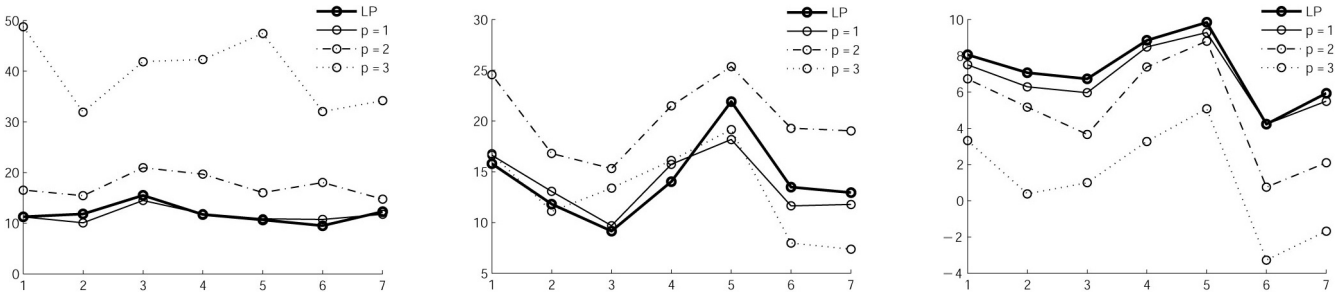

Figure 2: SNR (in dB) versus source index for Experiment 1 ($m = 7$, $n = 4$) – (left) noise-free: $\sigma_n = 0.001$, (middle) low-noise : $\sigma_n = 0.01$, (right) high-noise : $\sigma_n = 0.1$.

To simplify further equations, we introduce $\mathbf{P} \triangleq (\tilde{\mathbf{A}}\tilde{\mathbf{A}}^{\mathrm{T}})^{-1}$. Rewriting (4), we get

$$\tilde{\mathbf{s}}_0 = \tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{P}\mathbf{x} - (\tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{P}\hat{\mathbf{A}})\hat{\mathbf{s}} \triangleq \mathbf{c} - \mathbf{B}\hat{\mathbf{s}}.$$

With the restriction imposed on p, $\mathbf{B} \triangleq \tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{P}\hat{\mathbf{A}}$ would be a tall (or square) matrix. Hence, $\hat{\mathbf{s}}$, in general, cannot be selected to make $\tilde{\mathbf{s}}_0$ vanish. Instead, we may again use the corresponding pseudo-inverse to choose the $\hat{\mathbf{s}}$ which minimizes $\tilde{\mathbf{s}}_0$, i.e.

$$\hat{\mathbf{s}}_{opt} = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{c} = (\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{P}\hat{\mathbf{A}})^{-1}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{P}\mathbf{x}.$$

To summarize, the solution of $\mathbf{x} = \mathbf{A}\mathbf{s}$ which minimizes $\mathbf{s}^{\mathrm{T}}\mathbf{W}\mathbf{s}$, with $\mathbf{W}$ given by (2) may be written as

$$\begin{cases} \hat{\mathbf{s}}_{opt} = (\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{P}\hat{\mathbf{A}})^{-1}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{P}\mathbf{x} \\ \tilde{\mathbf{s}}_{opt} = \tilde{\mathbf{A}}^{\mathrm{T}}\mathbf{P}(\mathbf{x} - \hat{\mathbf{A}}\hat{\mathbf{s}}_{opt}) \end{cases}. \tag{5}$$

Note that the solution is still linear in $\mathbf{x}$. The minimum value of the cost function would be

$$\left\|\tilde{\mathbf{s}}_{opt}\right\|_2^2 = (\mathbf{x} - \hat{\mathbf{A}}\hat{\mathbf{s}}_{opt})^{\mathrm{T}}\mathbf{P}(\mathbf{x} - \hat{\mathbf{A}}\hat{\mathbf{s}}_{opt}).$$

To obtain the sparsest solution, the cost function will then be minimized over (at least) the entire family of diagonal weight matrices having exactly $m-p$ unity and p zero diagonal entries. This family will be denoted by W_p. The optimization is carried by the enumeration of all the $\binom{m}{p}$ possible cases. Some directions for the choice of p will be given in the next section when discussing simulation results. Empirical results suggest that in most cases, the algorithm attains its best performance for $p \approx m - n$. There may also be situations where minimizing over more than one family provides better results.

3. SIMULATION RESULTS

In this section simulation results regarding the performance of the proposed method will be examined. We will use the "minimal l^1 norm" solution[2] as a benchmark for comparison. Also the SNR[3] associated with each source, computed over the entire range of samples, will be used as the main performance measure. Although it may be argued that the SNR is not a suitable measure when dealing with sparse signals, it may still be used to make rough comparisons. The simulations are performed using *synthetically generated* sources. Each source is independently obtained from a source density which is modelled as a Gaussian mixture, i.e. the sum of weighted Gaussian densities. To be specific, each source is derived from a $N\ (0,1)$ density with probability π_1 and from a $N\ (0,\sigma_n^2)$ with probability $1-\pi_1$. The value of σ_n is usually much less than unity and it is mainly used to model noisy perturbations over the zero samples. The value of π_1, the probability that each source has a considerable value, is taken to be 0.05 (except for the third experiment). Each source is also normalized to unit-energy (over time) before the mixing matrix is applied. The mixing matrix is constructed columnwise by drawing m, $n \times 1$ vectors from a uniform distribution on the unit sphere S^{n-1} in $\mathbb{R}^n$.

The performance of the algorithm is now illustrated by summarizing the results of three experiments, examining the behavior of the method in noisy environments and for high dimensional problems. Complexity issues will also be addressed along the way.

3.1 Experiment 1 – Effect of noise

For the first experiment, we take $m = 7$ and $n = 4$. In Fig. 2, the SNRs obtained by the method are illustrated for different values of $p\ (= 1,2,3)$ and different levels of noise $(\sigma_n = 0.001,\ 0.01,\ 0.1)$. Recall that p defines the number of diagonal zeros for the family of weight matrices (i.e. W_p) over which the minimization is carried. The figures depict a typical behavior, not the average (or the worst case) one, i.e. we have tried to illustrate the most frequent behavior for each setting.

It is seen that the highest performance is due to $p = 3$ for the (relatively) noiseless case, while increasing the noise causes $p = 2$, and then $p = 1$ to take the place of $p = 3$. More experiments show that in low noise situations the choice $p = m - n$ is indeed a good one. The figure shows that for such low noise scenarios, the proposed method performs relatively better than LP. Also note that for all the three noise levels, the algorithm with $p = 1$ produces results nearly as good as those of the LP (this is not usually the case for higher dimensions). As we will see, the $p = 1$ case has the added advantage of being computationally less complex than LP.

[2] Since the minimum l^1 norm solution is obtained by Linear Programming, we may use the term 'LP method' when refering to it.

[3] By SNR, we mean the quantity $\left(\sum_t s_i^2(t)\right) / \sum_t (s_i(t) - \hat{s}_i(t))^2$ where $\hat{s}_i(.)$ is the estimated i-th source.

3.2 Experiment 2 - Complexity

In this experiment , we first take $m = 50$, $n = 30$ and a relatively low-noise environment ($\sigma_n = 0.01$) to study the performance of the algorithm in higher dimensions. Fig. 3 illustrates the results. The SNRs produced by the algorithm are shown for three values of p ($= 1, 2, 3$). We observe that progressively better results (toward that of LP) are obtained as we increase p, but the progress is slow. Implementing the algorithm for values of p greater than 3 is highly impractical in this case (i.e. at $m = 50$), due the computational complexity. We surmise that further increase of p would produce SNRs near (or even better) than those of LP.

The reason why the performance with $p = 1, 2$ is not satisfactory may intuitively be explained in terms of the mixture model we used. The model suggests that the expected number of *active* sources is $m \cdot \pi_1$. This is the expected value of the binomial distribution generating the hidden variables which decide whether or not each source has a considerable value. In this experiment, we have $m \cdot \pi_1 = 50 \times 0.05 = 2.5$, suggesting that, on the average, more than two sources are active at each instant of time. This means that the sample source distribution tends to concentrate around those subspaces of $\mathbb{R}^{50}$ which have dimensions greater than two. Thus, we may get better results if we try to minimize the distance of $\mathbf{x} = \mathbf{A}\mathbf{s}$ to such subspaces by taking $p > 2$.

The above argument also shows that in determining the performance of the algorithm both the *problem dimension* and the *degree of sparsity* of the sources should be taken into account. In particular, $m \cdot \pi_1$ may be considered an important parameter. For example, as long as this quantity is kept (much) below unity, we expect the algorithm with $p = 1$ to perform well no matter what the dimension of the problem is – more on this will be said in the context of the third experiment. The argument also suggests that, in general, we should take p at least equal to $m \cdot \pi_1$ (or of its order). Increasing the problem dimension, while keeping the sparsity of the sources (i.e. π_1) fixed, will then increase the required value of p ($\gtrsim m \cdot \pi_1$). In other words, *the required complexity of the algorithm will grow very fast with problem dimension (for fixed sparsity).*

In order to obtain an intuition for how complex the algorithm may get when p is increased, we have plotted the relative computational time (with respect to LP) as a function of the problem dimension. Fig. 4 illustrates the results for $p = 1, 2, 3$. The quantity used to measure *relative complexity* is

$$\Delta T(m) = \log_{10} \frac{\text{computational time of an algorithm at } m}{\text{computational time of LP at } m}. \tag{6}$$

With this definition, the complexity of LP would be zero at all dimensions. We will fix the mixture-to-source ratio at 0.6, i.e. $n = 0.6m$. The number of sources (m) will then be taken as a measure of the problem dimension.

The figure clearly shows that the complexity of the proposed algorithm with $p = 1$, is about an order of magnitude less than that of LP. The relative complexity even decreases as the dimension is increased. *It is then highly desirable to use the algorithm with $p = 1$ instead of LP, whenever the performances are comparable. This is the case for low-dimensional or highly sparse problems*. For other values of p, although the complexity of the algorithm may be

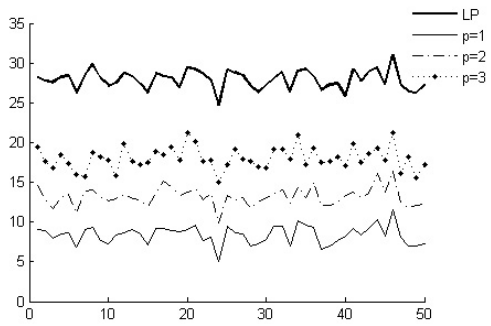

Figure 3: SNR (in dB) versus source index for Experiment 2 – $m = 50$, $n = 30$.

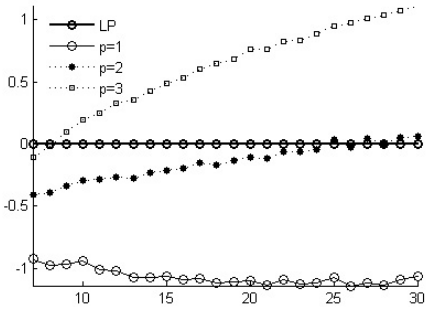

Figure 4: Relative complexity versus dimension for Experiment 2 – Relative complexity is a logarithmic measure based on computational time ratios, defined in (6) . It clearly shows the efficiency of $p = 1$ case and the exponential growth of complexity (with respect to LP) for higher values of p.

less than LP for lower dimensions, it eventually increases beyond that of LP. For $p = 2$, the relative complexity still remains below zero for the range of dimensions illustrated (i.e. $m = 7, \ldots, 30$). The $p = 3$ case, however, quickly grows in complexity. This exponential growth suggests that for *high-dimensional or moderately sparse problems* (where a large value of p is required for a comparable performance), LP is the preferred algorithm.

3.3 Experiment 3 – Sparsity and Dimension

In this experiment, we investigate the combined effect of sparsity and dimension on the performance of the proposed algorithm. Examining *SNR patterns* of Fig. 5 provides a good insight. In these plots, the horizontal and vertical axes represent, respectively, m (a measure of problem dimension) and π_1 (a measure of sparsity). The values of *average SNR per source* are mapped to shades of gray. In other words, each tile represents the average SNR obtained at a specific dimension and a sparsity level. A lighter color represent a higher SNR. White areas of each image then correspond to situations where the algorithm performs the best.

In all the plots, regions of high SNR tend to concentrate more across the left edge of the image which corresponds to high levels of sparsity. This is what to be expected. Note, however, that the general behavior of LP is different from that of the proposed algorithm. For a fixed level of sparsity,

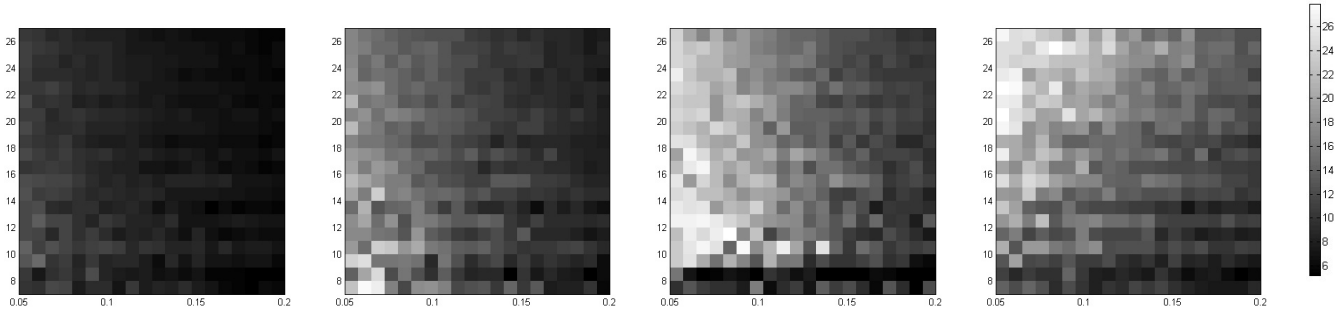

Figure 5: Average SNR patterns for Experiment 3 – The horizontal and vertical axes are respectively m (the number of sources) and π_1 (a measure of sparsity). SNR (in dB) is mapped to shades of gray. The mixture-to-source ratio is kept at $n/m = 0.6$. From left to right : the proposed algorithm with $p = 1,\ p = 2,\ p = 3,$ and LP.

performance of LP gets better as the dimension is increased (i.e. high SNR regions tend to be around the upper left corner of the image). In fact, as mentioned in the introduction, it has been shown that if sparsity is high enough, LP will eventually find the exact solution (almost surely).The performance of the proposed algorithm is, however, maximized around a specific dimension, for a fixed sparsity level. This is in accordance with the heuristics provided in the previous experiment which suggests that the quantity $m \cdot \pi_1$ is what determines the performance. Note, in particular, that if constant-SNR curves where to be plotted, they would be similar to constant-$(m \cdot \pi_1)$ curves.

In general, high-SNR regions of the algorithm are around the lower-left corner of the image for lower values of p and will gradually shift toward the upper-left corner (where LP performs the best) as p is increased. This again suggests that LP is not the best choice for all problems. With the proper choice of p we may obtain better results for a problem located in a specific region of the sparsity-dimension plane. This is specially the case for low-dimensional problems where the required value of p is usually small, leading to a reasonable complexity.

4. CONCLUDING REMARKS

We have shown that by viewing sparse decomposition as finding the solution which is nearest to the source (sample) distribution, a simple method of decomposition may be devised. In general, it is based on finding a solution of $\mathbf{x} = \mathbf{A}\mathbf{s}$ which minimizes the quadratic form $\mathbf{s}^{\mathrm{T}}\mathbf{W}\mathbf{s}$, over a family of weight matrices (W_p). The family is taken to be that of diagonal matrices with *zero or one* diagonal entries. The solution for minimization over $\mathbf{x} = \mathbf{A}\mathbf{s}$ may be obtained in closed form, while optimization over W_p is done via combinatorial search.

It was observed that the algorithm with $p = 1$ provides a fast and efficient alternative for LP, in low-dimensional or highly sparse problems. For higher dimensions or for moderately sparse sources, LP is a more effective choice which combines a good performance with a reasonable complexity. The proposed algorithm with higher values of p fills the gap between these two extremes in the sparsity-dimension plane. It provides a trade-off between performance and complexity through the proper choice of p.

The algorithm presented can be improved if more efficient methods are used to search in W_p-domain. One suggestion is the use of EM algorithm to implement the MAP estimator of sources under a Gaussian mixture prior. The E-step would then lead to a quadratic cost function similar to what presented here, but with weights determined by the posterior distribution of the hidden variables obtained at the previous iteration. Another approach is to somehow detect which sources are active and use the information to construct the appropriate weight matrix. This will eliminate the search in W_p-domain. An estimation of the number of active sources would also be helpful in deciding a proper value for the p parameter. Recent results on estimation, detection, and classification based on mixture models may be of use here.

REFERENCES

[1] M. Babaie-Zadeh, C. Jutten, A. Mansour, Sparse ICA via cluster-wise PCA, *to appear in Neurocomputing*

[2] P. Bofill, M. Zibulevsky, Underdetermined Blind Source Separation using Sparse Representations, *Signal Processing*, 81, 2353–2362, 2001.

[3] S. Chen, D.L. Donoho, M.A. Saunders, Atomic Decomposition by Basis Pursuit. *SIAM J. Sci.Comp.*, 20, 1, 33–61, 1999.

[4] D.L. Donoho (2004) For Most Large Underdetermined Systems of Linear Equations the Minimal l^1 norm Solution is also the sparsest Solution. URL : http://stat.stanford.edu/˜donoho/Reports/2004.

[5] D.L. Donoho, X. Huo, Uncertainty Principles and Ideal Atomic Decomposition. *IEEE Trans. Info. Thry*. 47(7), 2845-62, Nov. 2001.

[6] M. Elad and A.M. Bruckstein, A generalized uncertainty principle and sparse representations in pairs of bases. *IEEE Trans. Info Thry*. 48, 2558–2567, Sept. 2002.

[7] Y.Q. Li, A. Cichocki, S. Amari, Analysis of sparse representation and blind source separation, *Neural computation*, 16(6), 1193-1234, 2004.

[8] I. Takigawa, M. Kudo, J. Toyama, Performance analysis of Minimum l_1-norm solutions for underdetermined source separation, *IEEE Trans. Signal processing*, 52(3), 582-591, March 2004.

[9] M. Zibulevsky, B. A. Pearlmutter, Blind Source Separation by Sparse Decomposition, *Nueral Computation*, 13(4), 2001.