

# Blind source separation using block-coordinate relative Newton method

Alexander M. Bronstein  
alexbron@ieee.org

Michael M. Bronstein  
bronstein@ieee.org

Michael Zibulevsky  
mzib@ee.technion.ac.il

*Department of Electrical Engineering,  
Technion – Israel Institute of Technology  
Haifa 32000*

## ABSTRACT

Presented here is a generalization of the relative Newton method, recently proposed for quasi-maximum likelihood blind source separation. Special structure of the Hessian matrix allows performing block-coordinate Newton descent, which significantly reduces the algorithm computational complexity and boosts its performance. Simulations based on artificial and real data showed that the separation quality using the proposed algorithm is superior compared to other accepted blind source separation methods.

**Keywords:** blind source separation, block-coordinate optimization, quasi-maximum likelihood, Newton algorithm.

## 1. INTRODUCTION

The term *blind source separation* (BSS) refers to a wide class of problems in acoustics, medical signal and image processing, hyperspectral imaging, etc., where one needs to extract the underlying 1D or 2D sources from a set of linear mixtures without any knowledge of the mixing matrix. As a particular case, consider the problem of equal number of sources and mixtures, in which a  $N$ -channel sensor signal  $\mathbf{x}(t)$  arises from  $N$  unknown scalar source signals  $s_i(t)$ , linearly mixed by an unknown  $N \times N$  invertible matrix  $\mathbf{A}$ :

$$\mathbf{x}(t) = \mathbf{A}s(t). \quad (1)$$

We wish to estimate the mixing matrix  $\mathbf{A}$  (or, alternatively, the *unmixing* matrix  $\mathbf{W} = \mathbf{A}^{-1}$ ) and the source signal  $s(t)$ . In the discrete-time case ( $t = 1, \dots, T$ ) we can use matrix notation

$$\mathbf{X} = \mathbf{A}\mathbf{S}, \quad (2)$$

where  $\mathbf{X}$  and  $\mathbf{S}$  are  $N \times T$  matrices, containing the signals  $x_i(t)$  and  $s_i(t)$  in the corresponding rows. A 2D case can be thought of in terms of (2) if the 2D mixture signals (images) are parsed into vectors.

Under the assumption that the sources are stationary and white (i.e.  $s_i(t)$  are i.i.d. for every  $t$ ), one can write down the normalized minus-log-likelihood of the observed data  $\mathbf{X}$ :

$$L(\mathbf{W}; \mathbf{X}) = -\log |\det \mathbf{W}| + \frac{1}{T} \sum_{i,t} h(\mathbf{W}_i \mathbf{x}(t)), \quad (3)$$

where  $\mathbf{W}_i$  is the  $i$ -th row of  $\mathbf{W}$ ,  $h(\cdot) = -\log f(\cdot)$ , and  $f(\cdot)$  is the probability density function (pdf) of the sources. Even when  $h(\cdot)$  is not exactly equal to  $-\log f(\cdot)$ , minimization of (3) leads to a consistent estimation, known as *quasi-maximum likelihood* [25]. Quasi-ML is convenient when the source pdf is unknown, or not well-suited for optimization.

The relative Newton approach was recently proposed in [26], as an improvement of the Newton method used in [25] for quasi-maximum likelihood blind source separation. It was noted that the block-diagonal structure of the Hessian allows its fast approximate inversion, leading to the *modified relative Newton step*. In current work, we extend this approach by introducing a block-coordinate relative Newton method, which possesses faster convergence in approximately constant number of iterations.

A particularly important case is when the sources are *sparse* or *sparsely representable*. In this case, the absolute value function, or its smooth approximation is a good choice for  $h(\cdot)$  [12], [20], [22], [27], [28], [29] (see Appendix A for explicit form of  $h$ ). However, the resulting objective function appears especially difficult for optimization. The widely accepted *natural gradient* method shows poor convergence when the approximation of the absolute value becomes too sharp. The relative Newton method allows to overcome this obstacle and shows better convergence [26].

This paper consists of five sections. The second section is dedicated to the idea of relative optimization and the fast approximate inversion of the Hessian, which are the core of the modified Newton algorithm proposed in [26]. In section 3, we describe the idea of block-coordinate optimization and show how the modified relative Newton algorithm can be improved using this technique. From complexity analysis of one iteration, we draw conclusions when the block-coordinate approach is advantageous. In section 4, we compare our block-coordinate method to the original modified relative Newton algorithm on simulated and real data. The focus of the simulation is on sparse and sparsely-representable data, which is especially hard, as mentioned before. Other state-of-the-art blind source separation methods are used as a reference point. Section 5 concludes the work.

## 2. RELATIVE NEWTON ALGORITHM

The following *relative optimization* (RO) algorithm for minimization of the quasi-ML function (2) was used in [26]:

1. Start with an initial estimate  $\mathbf{W}^{(1)}$  of the separation matrix;
2. **FOR**  $k=1,2,\dots$ , until convergence:

3. Compute current source estimate  $\mathbf{U}^{(k)} = \mathbf{W}^{(k)}\mathbf{X}$ ;
4. Starting with  $\mathbf{V} = \mathbf{I}$ , compute  $\mathbf{V}^{(k+1)}$  producing a Newton step of  $L(\mathbf{V}; \mathbf{U}^{(k)})$ .
5. Update the estimated separation matrix  $\mathbf{W}^{(k+1)} = \mathbf{V}^{(k+1)}\mathbf{W}^{(k)}$ .
6. **END.**

We should note that if instead of the Newton optimization, stage 4 is carried out using a standard gradient descent method, the *relative (natural) gradient* method [3], [11], [13] is obtained.

## 2.1. Gradient and Hessian

Using the Newton method on stage 4 of the RO algorithm requires the knowledge of the Hessian of  $L(\mathbf{W}; \mathbf{X})$ . Since  $L(\mathbf{W}; \mathbf{X})$  is a function of a matrix argument, its gradient is also a matrix

$$\mathbf{G}(\mathbf{W}) = \nabla_{\mathbf{W}} L(\mathbf{W}, \mathbf{X}) = -\mathbf{W}^{-\text{T}} + \frac{1}{T} h'(\mathbf{W}\mathbf{X})\mathbf{X}^{\text{T}}, \quad (4)$$

where  $h'(\mathbf{W}\mathbf{X})$  implies element-wise application of  $h'$ .

The Hessian  $\nabla^2 L$  can be written as a fourth-order tensor  $H$ , which is inconvenient in practice. Alternatively, one can convert the matrix  $\mathbf{W}$  into a  $N^2$ -long column vector  $\mathbf{w} = \text{vec}(\mathbf{W})$  by row-stacking, yielding the gradient

$$\mathbf{g}(\mathbf{w}) = \nabla_{\mathbf{w}} L(\mathbf{w}, \mathbf{X}) = \text{vec}(\nabla_{\mathbf{W}} L(\mathbf{W}, \mathbf{X})). \quad (5)$$

The Hessian is represented as a  $N^2 \times N^2$  matrix, via the differential of the gradient:

$$d\mathbf{g} = \mathbf{H}d\mathbf{w} = \text{vec}(d\mathbf{G}). \quad (6)$$

Omitting derivation details (see [26]), the  $k$ -th column of the Hessian of the first term in (3),  $-\log \det \mathbf{W}$ , can be expressed as

$$\mathbf{H}^k = \text{vec}(\mathbf{A}^j \mathbf{A}_i)^{\text{T}}, \quad (7)$$

where  $\mathbf{A} = \mathbf{W}^{-1}$  and  $\mathbf{A}_i$  and  $\mathbf{A}^j$  are its  $i$ -th row and  $j$ -th column, respectively, and  $k = (i-1)N + j$ . The Hessian of the second term,  $\frac{1}{T} \sum_{i,t} h(\mathbf{W}_i \mathbf{x}(t))$ , is a block-diagonal matrix with the following  $N \times N$  blocks:

$$\mathbf{B}^m = \frac{1}{T} \sum_t h''(\mathbf{W}_m \mathbf{x}(t)) \mathbf{x}(t) \mathbf{x}^{\text{T}}(t) ; m = 1, \dots, N \quad (8)$$

## 2.2. Basic relative Newton step

The *Newton direction*  $\mathbf{y}$  is given by the solution of the linear equation

$$\mathbf{H}\mathbf{y} = -\nabla L(\mathbf{w}; \mathbf{X}), \quad (9)$$

where  $\mathbf{H}$  denotes the Hessian matrix of  $L(\mathbf{w}; \mathbf{X})$ , defined in (7) – (8). New iterate  $\mathbf{w}^+$  is given by  $\mathbf{w}^+ = \mathbf{w} + \mathbf{a}\mathbf{y}$ , where the step size  $\mathbf{a}$  is determined either by exact line search

$$\mathbf{a} = \underset{\mathbf{a}}{\operatorname{argmin}} L(\mathbf{w} + \mathbf{a}\mathbf{y}; \mathbf{X}), \quad (10)$$

or by backtracking line search:

1. Start with  $\mathbf{a} = 1$ ;
2. **WHILE**  $L(\mathbf{w} + \mathbf{a}\mathbf{y}; \mathbf{X}) > L(\mathbf{w}; \mathbf{X}) + \mathbf{b}\mathbf{a}\nabla L(\mathbf{w}; \mathbf{X})^\top \mathbf{y}$  :
3. Update  $\mathbf{a} \leftarrow \mathbf{g}\mathbf{a}$  ;
4. **END.**

where the common selection is  $\mathbf{b} = \mathbf{g} = 0.3$ .

In the RO algorithm, the optimization on stage 4 is carried out using a single Newton iteration with exact or backtracking line search. Given  $\mathbf{V} = \mathbf{I}$ , the first term of the Hessian  $\nabla^2 L(\operatorname{vec}(\mathbf{I}); \mathbf{X})$  becomes

$$\mathbf{H}_k = \operatorname{vec}^\top(\mathbf{e}_i \mathbf{e}_j^\top) \quad (11)$$

( $\mathbf{e}_i$  is the standard basis vector, containing 1 at the  $i$ -th coordinate). The second term is block-diagonal [26]. The Newton step derived from this case will be referred to as the *basic relative Newton step*.

### 2.3. Modified (fast) relative Newton step

In [26], it was shown that the second term (8) of the Hessian can be approximated diagonally (see derivation details in Appendix B). The Newton step derived from this case will be referred to as the *modified or fast Newton step*.

Using this approximation, solution of the Newton system requires the solution of  $\frac{1}{2}N(N-1)$  systems of  $2 \times 2$  linear equations

$$\begin{aligned} \mathbf{D}_{ij} \mathbf{Y}_{ij} + \mathbf{Y}_{ji} &= \mathbf{G}_{ij} \quad ; \quad i=1, \dots, N, \quad j=1, \dots, i-1 \\ \mathbf{D}_{ji} \mathbf{Y}_{ji} + \mathbf{Y}_{ij} &= \mathbf{G}_{ji} \end{aligned} \quad (13)$$

in order to find the off-diagonal elements, and  $1 \times 1$  systems

$$\mathbf{D}_{ii} \mathbf{Y}_{ii} + \mathbf{Y}_{ii} = \mathbf{G}_{ii} \quad (14)$$

in order to find the diagonal elements ( $\mathbf{D}$  is a  $N \times N$  matrix containing the raw-packed diagonal of the second term Hessian).

Computing the Hessian diagonal according to (12) requires  $N^2 T$  operations; solution cost of the set of the  $2 \times 2$  equations (13) is about  $15N^2$  operations. This implies that the modified Newton step has the asymptotic complexity of a gradient descent step.

## 3. BLOCK-COORDINATE RELATIVE NEWTON ALGORITHM

*Block-coordinate optimization* is based on decomposition of the vector variable into components (blocks of coordinates) and producing optimization steps in the respective block subspaces in a

sequential manner. Such algorithms usually have two loops: a step over block (inner iteration), and a pass over all blocks (outer iteration).

The main motivation for the use of block-coordinate methods can be that when most variables are fixed, we often obtain subproblems in the remaining variables, which can be solved efficiently. In many cases, block-coordinate approaches require significantly less outer iterations compared to conventional methods [16].

In our problem, the Hessian is approximately separable with respect to the pairs of symmetric elements of  $W$ , i.e. the Newton system splits into a set of  $2 \times 2$  systems. This brings us to the idea of applying the Newton step block-coordinately on these pairs. As we will see from the complexity analysis, the relative cost of the nonlinearity computation becomes dominate in this case, therefore, we can do one step further and use blocks of larger size, pair-wise symmetric.

The matrix  $W$  can be considered as consisting of  $M = N/K$  blocks of size  $K \times K$  :

$$W = \begin{bmatrix} W_{1,1} & \dots & W_{1,M} \\ \dots & & \dots \\ W_{M,1} & \dots & W_{M,M} \end{bmatrix} \quad (15)$$

The block-coordinate modified relative Newton step (as opposed to the *full* modified relative Newton step described in section 2.4) is performed by applying the relative Newton algorithm to the subspace of two blocks  $W_{ij}$  and  $W_{ji}$  at a time, while fixing the rest of the matrix elements. In order to update all the entries of  $W$ ,  $N(N-1)/2K^2$  inner iterations are required.

We obtain the following block-coordinate relative Newton algorithm:

1. Start with an initial estimate  $W^{(1)}$ ;
2. **FOR**  $k = 1, 2, \dots$ , until convergence:
  3. **FOR**  $i = 1, 2, \dots, M$ ,
  4. **FOR**  $j = i, \dots, M$ ,
  5. Efficiently update current source estimate  $U^{(k)} = W^{(k)}X$ , as shown later on;
  6. Starting with  $V = I$ , compute  $V^{(k+1)}$  producing one block-coordinate Newton step with the blocks  $W_{ij}$  and  $W_{ji}$  using (13) – (14).
  7. Update  $W^{(k+1)} = V^{(k+1)}W^{(k)}$ .
8. **END.**
9. **END.**
10. **END.**

Since only few elements of  $W$  are updated at each inner iteration, evaluation of the cost function, its gradient and Hessian can be significantly simplified. In the term  $Wx(t)$ , only  $2K$  elements are updated and consequently, the non-linearity  $h$  is applied to a  $2K \times T$  stripe to update the sum  $\sum_{i,t} h(W_i x(t))$ .

Since at each inner step the identity matrix  $I$  is substituted as an initial value of  $W$ , the updated matrix will be of the form

$$\mathbf{W} = \begin{bmatrix} \mathbf{I}_{K \times K} & & \mathbf{W}_{ij} & \\ & \mathbf{I}_{K \times K} & & \\ \mathbf{W}_{ji} & & \mathbf{I}_{K \times K} & \\ & & & \mathbf{I}_{K \times K} \end{bmatrix}. \quad (16)$$

It can be easily shown that the computation of the determinant of  $\mathbf{W}$  in (16) can be reduced to

$$\det \mathbf{W} = \det \hat{\mathbf{W}} \quad ; \quad \hat{\mathbf{W}} = \begin{bmatrix} \mathbf{I}_{K \times K} & \mathbf{W}_{ij} \\ \mathbf{W}_{ji} & \mathbf{I}_{K \times K} \end{bmatrix}, \quad (17)$$

and carried out in  $2K^3$  operations.

Similarly, the computation of the gradient requires applying  $h'$  to the updated  $2K \times T$  stripe of  $\mathbf{W}\mathbf{X}$  and multiplying the result by the corresponding  $2K \times T$  stripe of  $\mathbf{X}^T$ . In addition, the gradient requires inversion of  $\mathbf{W}$ , which can be done using the matrix  $\hat{\mathbf{W}}$ . When  $i \neq j$ , the inverse matrix  $\mathbf{A} = \mathbf{W}^{-1}$  consists of a unit diagonal, two blocks on the diagonal ( $\mathbf{A}_{ii}, \mathbf{A}_{jj}$ ) and two off-diagonal blocks ( $\mathbf{A}_{ij}, \mathbf{A}_{ji}$ ). These blocks can be obtained by the inversion of  $\hat{\mathbf{W}}$ :

$$\hat{\mathbf{A}} = \hat{\mathbf{W}}^{-1} = \begin{bmatrix} \mathbf{A}_{ii} & \mathbf{A}_{ij} \\ \mathbf{A}_{ji} & \mathbf{A}_{jj} \end{bmatrix}. \quad (18)$$

Computation of (18) also requires  $2K^3$  operations. To compute the Hessian, one should update  $2K$  elements in  $\mathbf{x}(t)\mathbf{x}^T(t)$  for each  $t = 1, \dots, T$  and apply  $h''$  to the updated  $2K \times T$  stripe of  $\mathbf{W}\mathbf{X}$ .

### 3.1. Iteration complexity

For convenience, we denote as  $\mathbf{a}_1$ ,  $\mathbf{a}_2$  and  $\mathbf{a}_3$  the number of operations required for the computation of the non-linearity  $h$  and its derivatives  $h'$  and  $h''$ , respectively (see Appendix A). A reasonable estimate of these constants for  $h$  in (22)-(23) is  $\mathbf{a}_1 = 6$ ,  $\mathbf{a}_2 = 2$ ,  $\mathbf{a}_3 = 2$ . We will also use  $\mathbf{a} = \mathbf{a}_1 + \mathbf{a}_2 + \mathbf{a}_3$ .

A single block-coordinate relative Newton inner iteration involves computation of the cost function, its gradient and Hessian, whose respective complexities are  $2(K^2T + K^3 + \mathbf{a}_1KT)$ ,  $2(K^2T + K^3 + \mathbf{a}_2KT)$  and  $2(K^2T + (\mathbf{a}_3 + 1)KT)$ . In order to compute the Newton direction,  $K^2$  systems of equations of size  $2 \times 2$  have to be solved, yielding in total solution of  $\frac{1}{2}N(N-1)$  systems per outer iteration, independent of  $K$ . Other operations have negligible complexity.

Therefore, a single block-coordinate outer Newton iteration will require about  $N^2T(3 + (\mathbf{a} + 1)/K)$  operations. Substituting  $K = N$ , the algorithm degenerates to the relative Newton method, with the complexity of about  $3N^2T$ . Therefore, the block-coordinate approach with  $K \times K$  blocks is advantageous, if the full relative Newton method requires more iterations by the factor

$$\mathbf{b} > 1 + \frac{\mathbf{a} + 1}{3K}, \quad (19)$$

Since line search is likely to require more iterations in a high-dimensional problem, additional reduction of the computation complexity may be obtained when  $K$  is sufficiently small compared to  $N$ .

### 3.2. Update of the smoothing parameter

Optimization of the likelihood function becomes difficult with the decrease of the smoothing parameter  $\mathbf{I}$ . To overcome this problem, it was proposed in [26] to perform sequential optimization, gradually decreasing the value of  $\mathbf{I}$ .

Let us denote

$$L(\mathbf{W}; \mathbf{X}; \mathbf{I}) = -\log |\det \mathbf{W}| + \frac{1}{T} \sum_{i,t} h_l(\mathbf{W}_i \mathbf{x}(t)), \quad (20)$$

where  $h_l$  is a parametric nonlinearity in (22). The sequential optimization algorithm has the following form:

1. Start with  $\mathbf{I}^{(1)}$  and an initial estimate  $\mathbf{W}^{(1)}$ ;
2. **FOR**  $k = 1, 2, \dots$ , until convergence:
  3. **FOR**  $i = 1, 2, \dots, M$ ,
  4. **FOR**  $j = i, \dots, M$ ,
    5. Efficiently update current source estimate  $\mathbf{U}^{(k)} = \mathbf{W}^{(k)} \mathbf{X}$ , as shown later on;
    6. Starting with  $\mathbf{V} = \mathbf{I}$ , compute  $\mathbf{V}^{(k+1)}$  producing one block-coordinate Newton step with the blocks  $\mathbf{W}_{ij}$  and  $\mathbf{W}_{ji}$ .
    7. Update  $\mathbf{W}^{(k+1)} = \mathbf{V}^{(k+1)} \mathbf{W}^{(k)}$ .
  8. **END.**
9. **END.**
10. Update the smoothing parameter  $\mathbf{I}^{(k+1)} = \mathbf{m} \mathbf{I}^{(k)}$  ;  $\mathbf{m} < 1$ .
11. **END.**

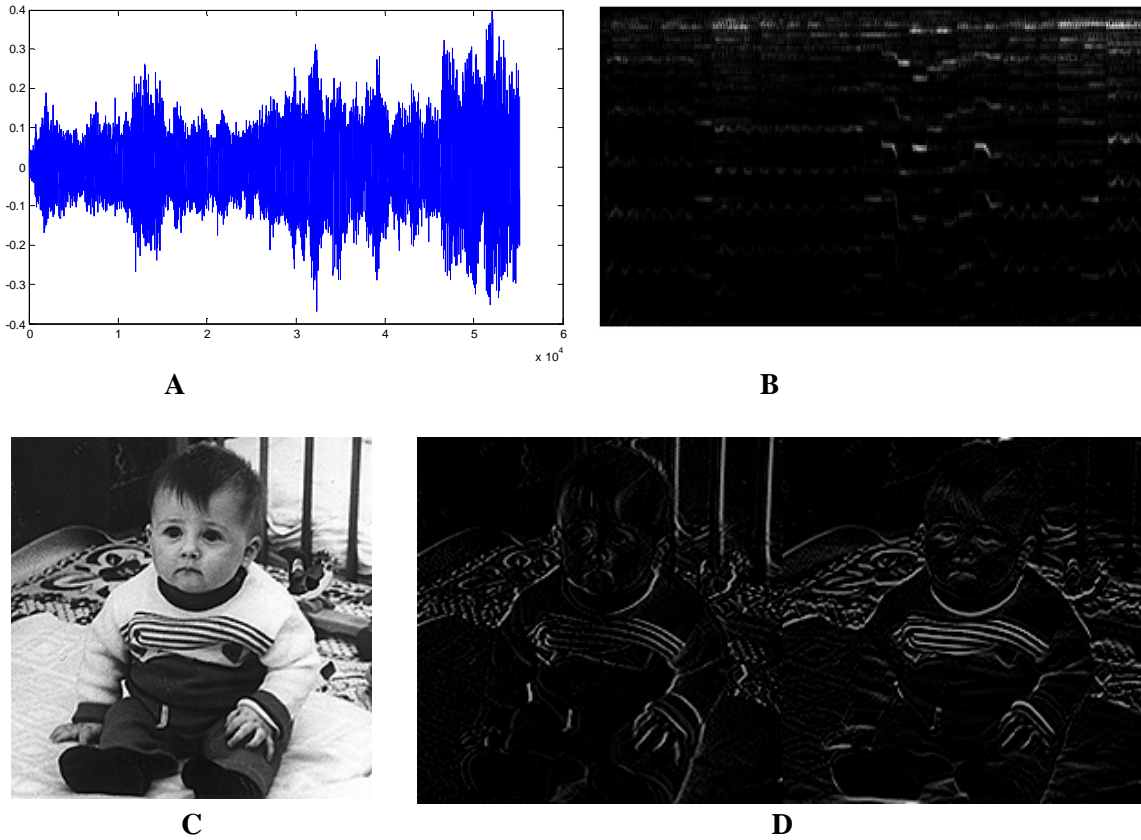


Figure 1 – Waveform of audio source 25, part from *Brahms Violin Sonata No. 3* (A) and its sparse representation using the STFT (B). Natural image source 30 (C) and its sparse representation using the discrete directional derivatives (B).

#### 4. COMPUTATIONAL RESULTS

The main focus of the simulation is on sparse and sparsely-representable sources. For numerical experiments, three data sets were used: artificial sparse signals (set I), audio signals (set II) and natural images (set III). These data sets are described in subsections 4.1 – 4.3. Data sets II and III were not originally sparse, and thus not the corresponding mixtures. However, if the sources are sparsely representable, i.e. there exists a linear transformation  $T$ , such that  $Ts_i(t)$  are sparse, due to linearity of (1), one can apply  $T$  on the mixtures  $x_i(t)$  rather than on the sources to obtain sparse data [27], [28], [29]. Then, separation algorithm is applied on  $Tx_i(t)$ .

Sparse representation of acoustic signals was obtained by the short time Fourier transform (STFT) as in [5] and [28]. In natural images, empirical observations show that the edges are sparse [6], hence one of the simplest sparse representation, the discrete derivative, was adopted here (see Figure 1). In all the experiments, the sources were artificially mixed using an invertible random matrix with uniform i.i.d. elements.

The separation quality (in terms of the interference-to-signal ratio (ISR) in dB units) of the relative Newton method was compared with this of stochastic natural gradient (Infomax) [13], [3], [11], [21], Fast ICA [17], [18] and JADE [9], [10]. In addition, the dependence of the

computational complexity (the total number of multiplications performed) on the block size was compared in the block-coordinate relative Newton algorithm. In all cases, the relative Newton optimization was stopped after the gradient norm reached  $10^{-10}$ .

We observed in numerous simulations that without sparse representation, Infomax, JADE and Fast ICA algorithms produced worse results (this corresponds to the recent observations, e.g. in [27], that sparse representation allows improve the performance of blind source separation algorithms). Hence, in order to make the comparison fair, we applied all the algorithms on “sparsified” data rather than on the original one.

#### 4.1. Sparse sources

First, the block coordinate relative Newton method was tested on sparse sources with the Bernoulli-Gaussian distribution,

$$f(s) = p\mathbf{d}(s) + (1-p) \frac{1}{\sqrt{2ps^2}} \exp\left(\frac{-s^2}{2s^2}\right) \quad (21)$$

generated using the MATLAB function `sprandn` (with  $p = 0.5$  and  $\mathbf{s} = 1$ ) and mixed using a random matrix. The block-coordinate algorithm (with block size  $K = 1, 3, 5$  and  $10$ ) was compared to the full relative Newton algorithm ( $K = N$ ) on problems of different size ( $N$  from 3 to 50 in integer multiplies of  $K$ ;  $T = 10^3$ ). The total number of the cost function, its gradient and Hessian evaluations was recorded and was used for complexity computation. On problems of different size, the experiments were repeated 10 times with different random mixing matrices. Figures 2 and 3 show the mean and the standard deviation values on these runs.

Remarkably, the number of outer iterations is approximately constant with the number of sources  $N$  in the block-coordinate method, as opposed to the full relative Newton method (see Figure 2). Particularly, for  $K = 1$  the number of outer iterations is about 10. Furthermore, the contribution of the non-linearity computation to the overall complexity is decreasing with the block size  $K$ . Hence, it explains why in Figure 3 the complexity normalized by the factor  $N^2T$  is almost the same for blocks of size  $K = 1, 3, 5$  and  $10$ . However, CPU architecture considerations may make larger blocks preferable.

The block-coordinate algorithm outperformed the relative Newton algorithm by about 3.5 times for  $N = 55$ . Figure 4 shows the average function and gradient evaluations on each line search step. It can be seen that this number increases with the block size.

As a reference point,  $N = 30$  sparse sources of length  $T = 10^4$ , mixed by a random matrix, were separated using the relative Newton method (with  $\mathbf{I} = 10^{-7}$ ), Infomax, Fast ICA and JADE algorithms. Table I shows that the ISRs of the separated sources are superior using our algorithm.

#### 4.2. Audio sources

As natural audio sources, we took  $N = 30$  instrumental and vocal music recordings (5 sec. at 11025 Hz sampling rate; the first 50000 samples were used). The mixtures were sparsely represented using the STFT (MATLAB function `specgram`), whose real and imaginary parts were concatenated and parsed into vectors, and then separated using the mentioned algorithms. In the relative Newton method, the smoothing parameter  $\mathbf{I} = 0.01$  was used. Table II compares the separation quality obtained using the relative Newton, Infomax, Fast ICA and JADE algorithms. In this experiment, our algorithm appears to be the best<sup>1</sup>.

Using part of the data set II (20 first sources), we performed a test of the block-coordinate relative Newton method with different block size. Figure 5 depicts the gradient norm (A) and the ISR of the separated sources (B) in the block-coordinate relative Newton method, as function of the computational complexity. The curves are average on 10 runs with random mixing matrices. Different curves correspond to blocks of size  $K = 2, 5, 10, 20$ , where the last one is the full relative Newton method [26]. Shaded areas denote the corresponding standard deviations. One can see that the block-coordinate approach yields faster convergence than full Newton.

### 4.3. Natural images

In the last experiment,  $N = 30$  natural  $200 \times 200$  images were taken (see Figure 6). The mixtures were sparsely represented using X- and Y-discrete derivatives concatenated and parsed into a vector ( $T = 80000$ ), and then separated using the mentioned algorithms. In the relative Newton method, the smoothing parameter  $\mathbf{I} = 0.01$  was used.

Table II compares the separation quality obtained using the relative Newton, Infomax, Fast ICA and JADE algorithms; in this experiment as well, our algorithm appears to be the best<sup>2</sup>. We also performed a test of the block-coordinate relative Newton method with different block size. Figure 7 depicts the gradient norm (A) and the ISR of the separated sources (B) in the block-coordinate relative Newton method, as function of the computational complexity. The curves are average on 10 runs with random mixing matrices. Different curves correspond to blocks of size  $K = 5, 15, 30$ , where  $K = 30$  is the full relative Newton method [26]. Shaded areas denote the corresponding standard deviations. One can see that the block-coordinate approach yields faster convergence than full Newton.

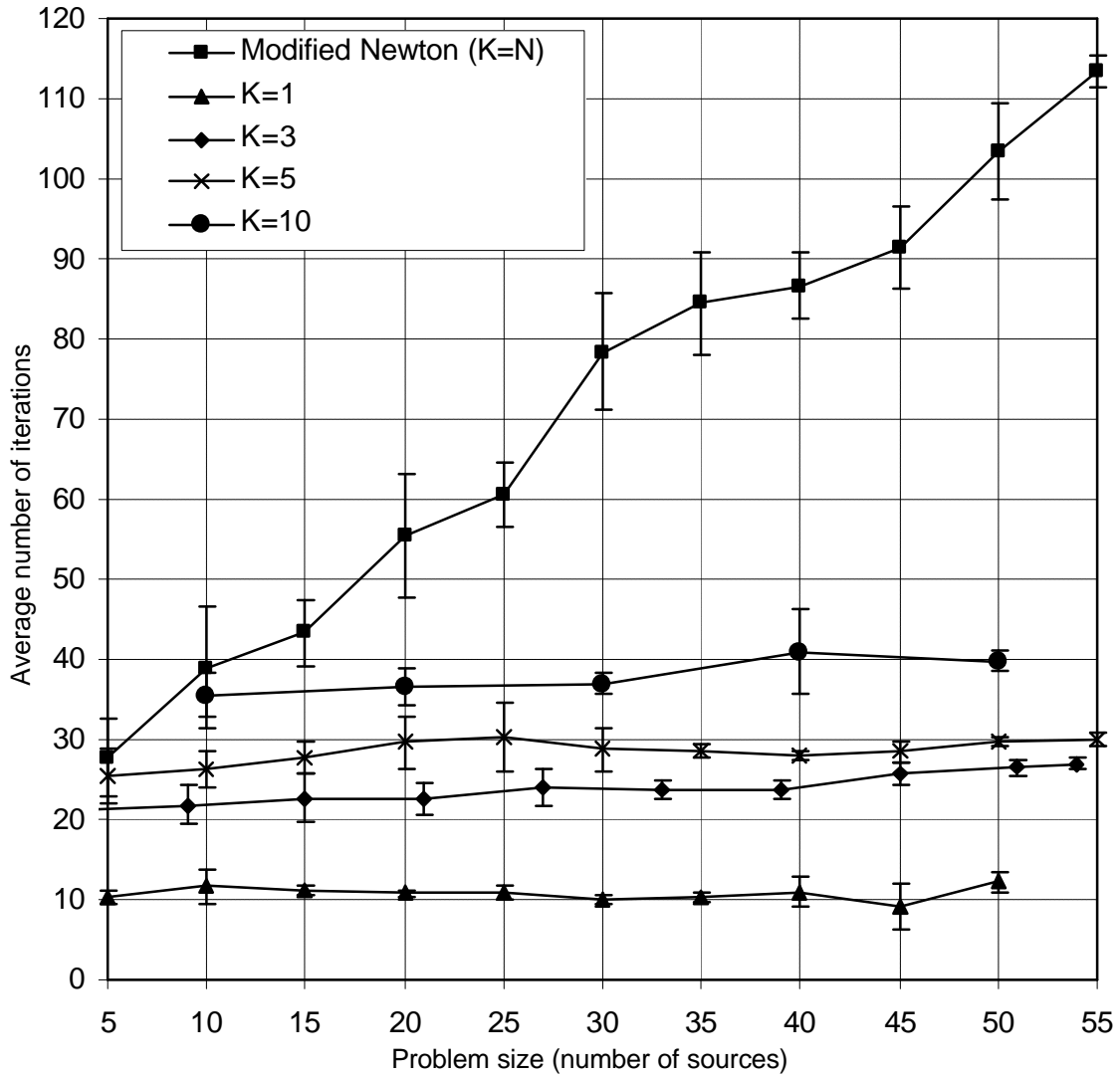


Figure 2 – Average number of outer iterations vs. the number of sources  $N$  for different block sizes  $K$ .

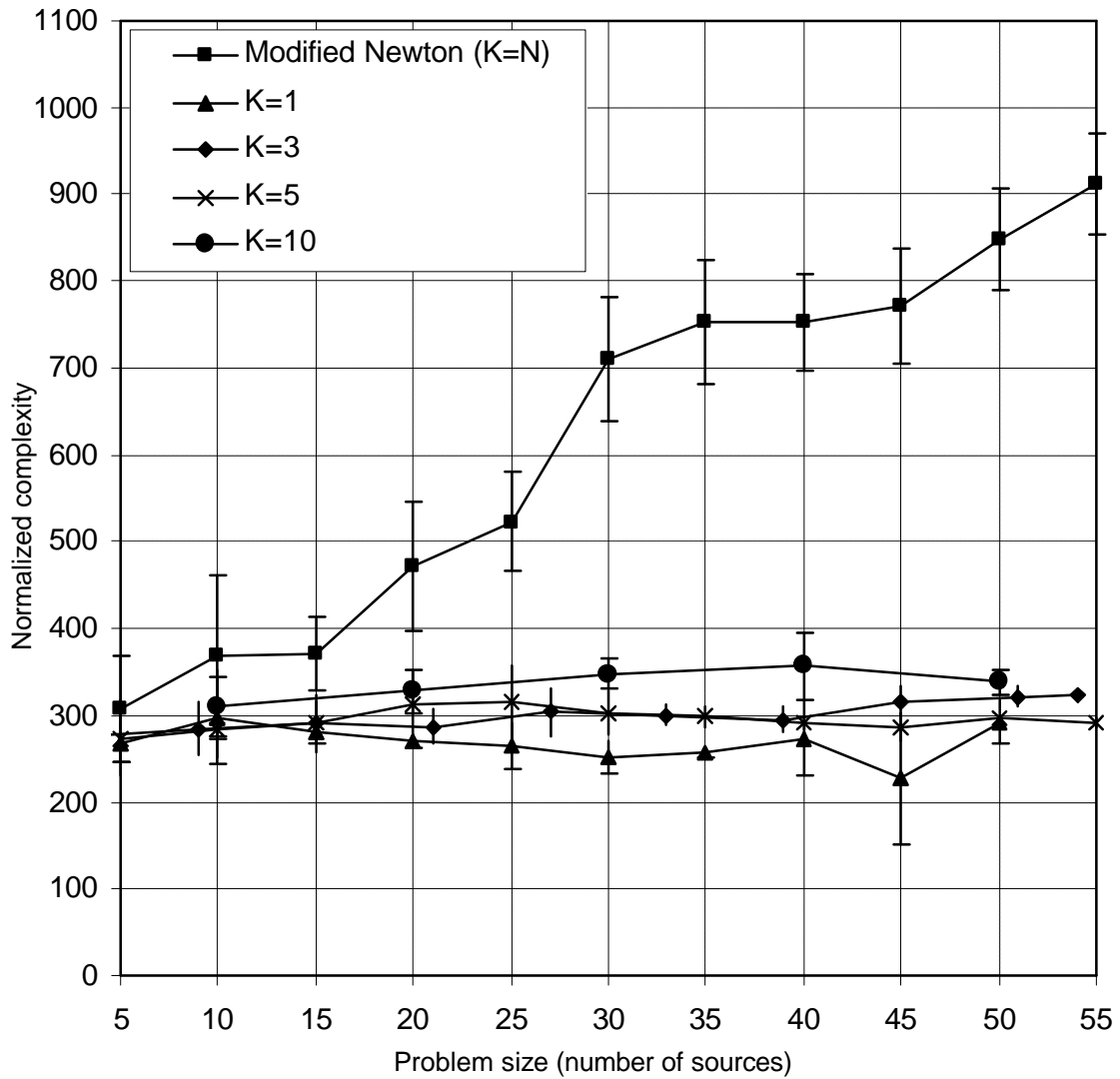


Figure 3 – Normalized complexity vs. the number of sources  $N$  for different block sizes  $K$ .

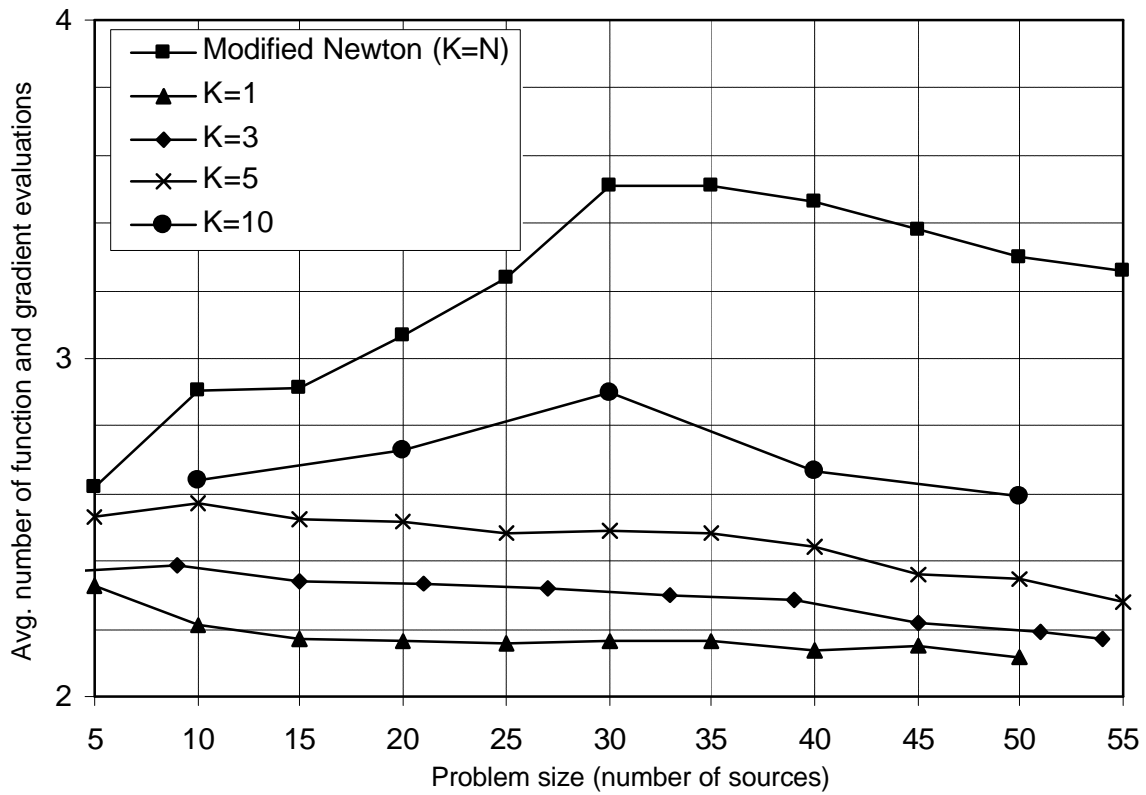


Figure 4 – Average number of function and gradient evaluations in line search vs. the number of sources  $N$  for different block sizes  $K$ .

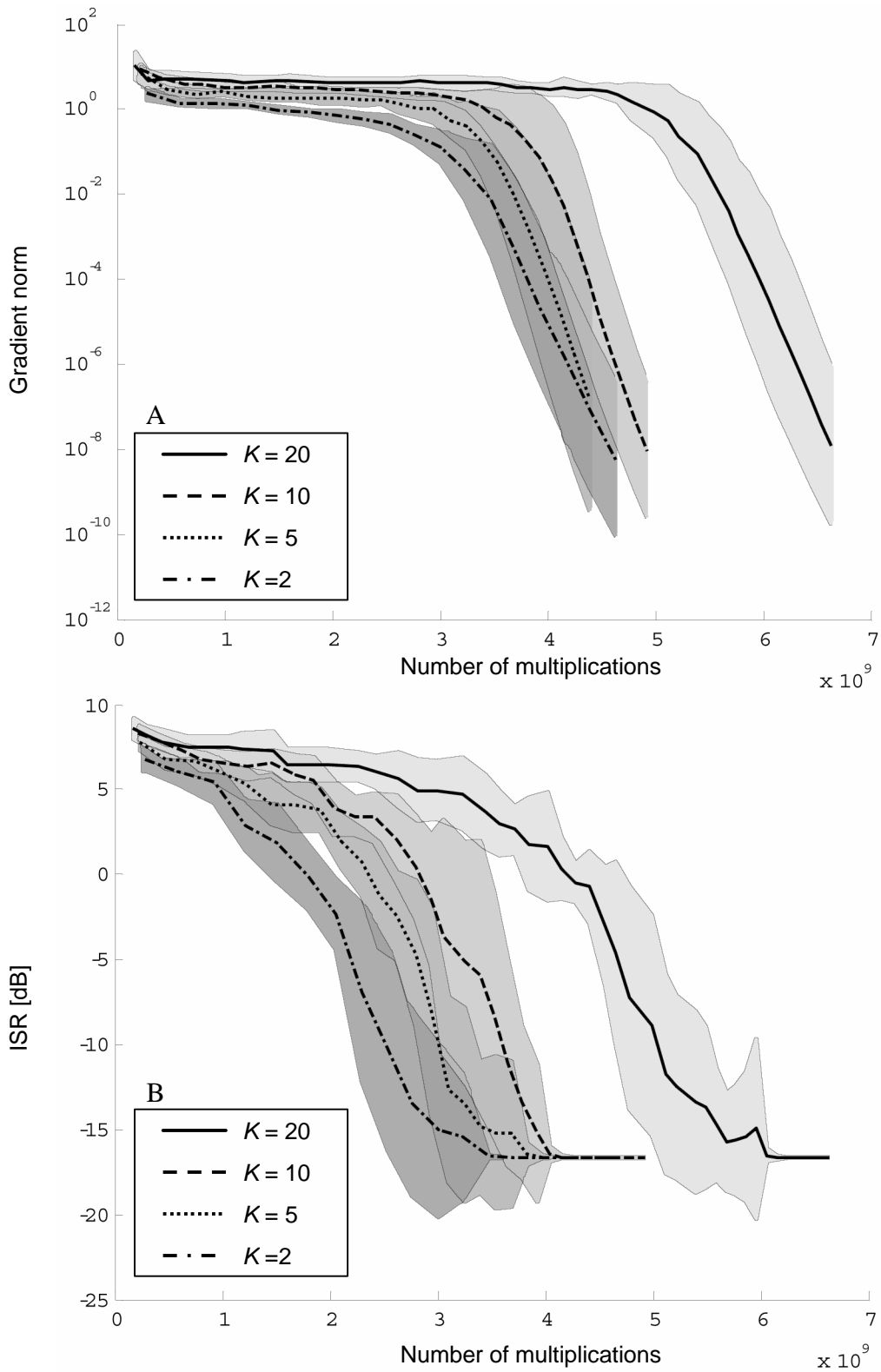


Figure 5 – Separation of 20 audio sources from data set II using the block-coordinate relative Newton method, for blocks of different size  $K$  ( $K=20$  corresponds to full relative Newton) Gradient norm (A) and ISR (B) vs. the number of multiplications (average on 10 runs with random mixing matrices. Shaded areas denote standard deviation).



Figure 6 – the sources (A) and examples of the first 6 mixtures (B).

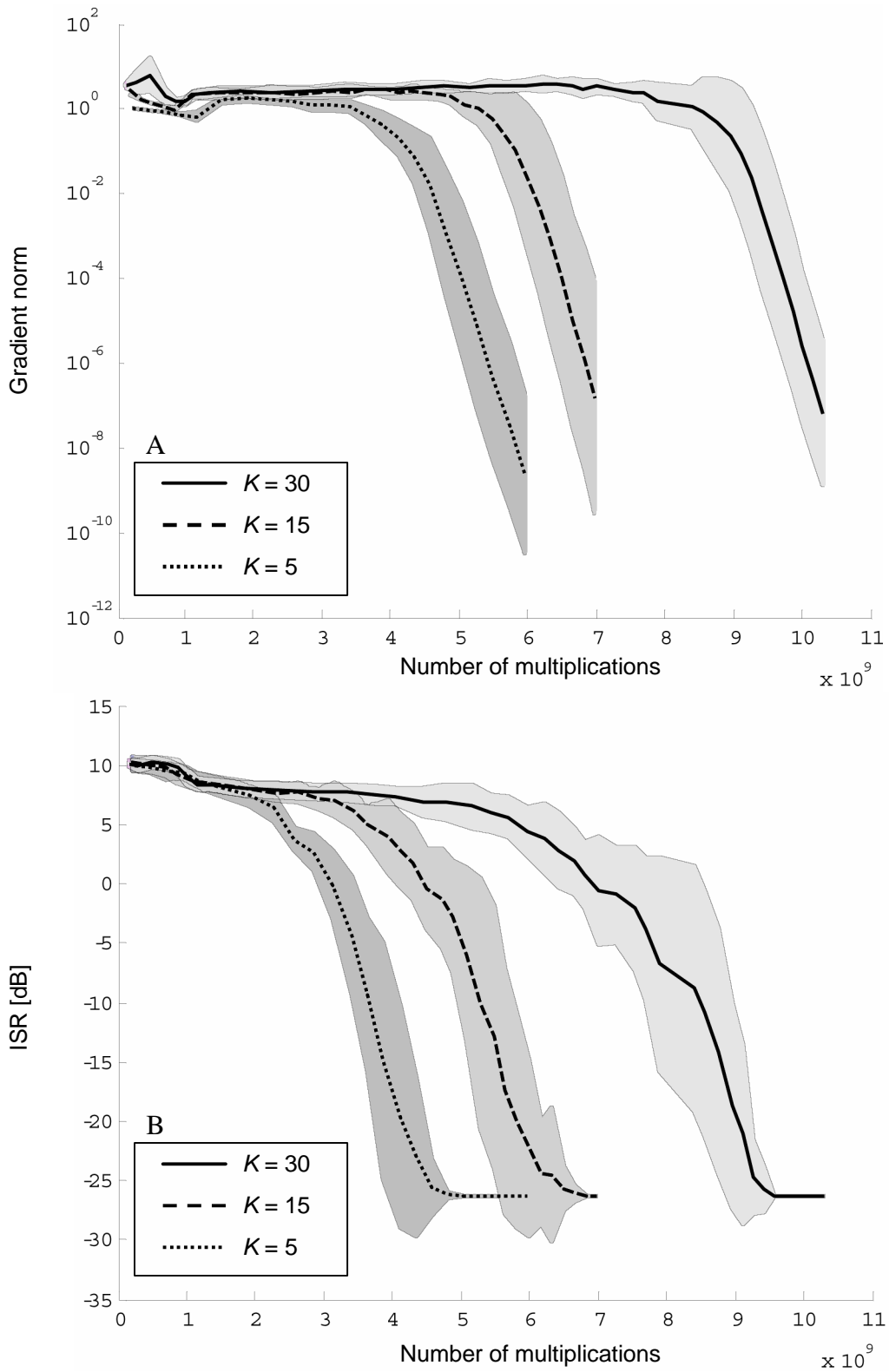


Figure 7 – Separation of 30 natural image sources from data set III using the block-coordinate relative Newton method, for blocks of different size  $K$  ( $K=30$  corresponds to full relative Newton) Gradient norm (A) and ISR (B) vs. the number of multiplications (average on 10 runs with random mixing matrices. Shaded areas denote standard deviation).

Table I – separation quality of sparse signals (data set I)

ISR [dB]	NEWTON	INFOMAX	FASTICA	JADE
Best	-172.9769 dB	-34.3449 dB	-23.8166 dB	-26.7835 dB
Worst	-167.9964 dB	-29.6393 dB	-18.6372 dB	-21.8914 dB
Mean	-170.3825 dB	-31.8812 dB	-21.3462 dB	-24.2529 dB

Table II – separation quality of audio signals (data set II)

ISR [dB]	NEWTON	INFOMAX	FASTICA	JADE
Best	-46.6823 dB	-37.3401 dB	-25.1501 dB	-25.7798 dB
Worst	-25.7237 dB	-23.3460 dB	-2.1066 dB	-9.0165 dB
Mean	-35.7882 dB	-29.8169 dB	-17.2189 dB	-18.8159 dB

Table III – separation quality of images (data set III)

ISR [dB]	NEWTON	INFOMAX	FASTICA	JADE
Best	-57.3494 dB	-38.5244 dB	-30.5364 dB	-32.3508 dB
Worst	-31.7391 dB	-25.6605 dB	-19.7479 dB	-22.1477 dB
Mean	-40.0111 dB	-33.1128 dB	-24.7007 dB	-27.8450 dB

## 5. CONCLUSIONS

We presented a block-coordinate version of the relative Newton algorithm for quasi-ML blind source separation. The most intriguing property, demonstrated by computational experiments, is the almost constant number of iterations (independent of the number of sources) of the block-coordinate relative Newton algorithm. Though formal mathematical explanation of this phenomenon is an open question at this point, it is of very high importance for practical applications.

In large problems, we observed a nearly three-fold reduction of the computational burden of the modified Newton method by using the block-coordinate approach. The use of an accurate approximation of the absolute value nonlinearity in the quasi-ML function leads to accurate separation of sources, which have sparse representation (e.g. by means of STFT, wavelets, discrete derivative, etc.). We must stress that the method is general and is applicable to other distributions of sources (not necessarily sparse).

## 6. ACKNOWLEDGEMENTS

The authors would like to acknowledge support for this project by the Ollendorff Minerva Center and the HASSIP Research Network Program HPRN-CT-2002-00285, sponsored by the European Commission.

## 7. REFERENCES

- [1] T. Akuzawa and N. Murata, "Multiplicative nonholonomic Newton-like algorithm," *Chaos, Solitons and Fractals*, vol. 12, p. 785, 2001.
- [2] T. Akuzawa, "Extended quasi-Newton method for the ICA," tech. rep., Laboratory for Mathematical Neuroscience, RIKEN Brain Science Institute, 2000.  
Available: <http://www.mns.brain.riken.go.jp/~akuzawa/publ.html>.
- [3] S. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems 8*, MIT Press, 1996.
- [4] A. J. Bell and T. J. Sejnowski, "An information maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.
- [5] P. Bofill P, M. Zibulevsky, "Underdetermined blind source separation using sparse representations", *Signal Processing*, Vol.81, No 11, pp.2353-2362, 2001.  
Available: <http://iew3.technion.ac.il/~mcib/>
- [6] A. M. Bronstein, M. M. Bronstein, M. Zibulevsky and Y. Y. Zeevi, "Separation of reflections via sparse ICA," *ICIP 2003*, submitted.  
Available: <http://visl.technion.ac.il/bron/works>
- [7] J.-F. Cardoso, "On the performance of orthogonal source separation algorithms," in *EUSIPCO*, (Edinburgh), pp. 776–779, Sept. 1994.
- [8] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proceedings of the IEEE*, vol. 9, pp. 2009–2025, Oct. 1998.
- [9] J.-F. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157–192, 1999.
- [10] J.-F. Cardoso, "JADE for real-valued data," 1999.  
Available: <http://sig.enst.fr:80/~cardoso/guidesepsou.html>.
- [11] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Transactions on Signal Processing*, vol. 44, no. 12, pp. 3017–3030, 1996.
- [12] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.
- [13] A. Cichocki, R. Unbehauen, and E. Rummert, "Robust learning algorithm for blind separation of signals," *Electronics Letters*, vol. 30, no. 17, pp. 1386–1387, 1994.

- [14] A. Cichocki, S. Amari, and K. Siwek, “ICALAB toolbox for image processing – benchmarks,” 2002.  
Available: <http://www.bsp.brain.riken.go.jp/ICALAB/ICALABImageProc/benchmarks/>.
- [15] P. E. Gill, W. Murray, and M. H. Wright, Practical Optimization. New York: Academic Press, 1981.
- [16] L. Grippo, and M. Sciandrone, “Globally convergent block-coordinate techniques for unconstrained optimization,” Optimization Methods and Software, vol. 10(4), pp.587-637, 1999.
- [17] A. Hyvärinen, “The Fast-ICA MATLAB package,” 1998.  
Available: <http://www.cis.hut.fi/~aapo>
- [18] A. Hyvärinen, “Fast and robust fixed-point algorithms for independent component analysis,” IEEE Transactions on Neural Networks, vol. 10, no. 3, pp. 626–634, 1999.
- [19] M. Joho and K. Rahbar, “Joint diagonalization of correlation matrices by using Newton methods with application to blind signal separation,” SAM 2002, 2002.  
Available: [http://www.phonak.uiuc.edu/~joho/research/publications/sam\\_2002\\_2.pdf](http://www.phonak.uiuc.edu/~joho/research/publications/sam_2002_2.pdf).
- [20] M. S. Lewicki and B. A. Olshausen, “A probabilistic framework for the adaptation and comparison of image codes,” Journal of the Optical Society of America, vol. 16, no. 7, pp. 1587–1601, 1999. in press.
- [21] S. Makeig, “ICA toolbox for psychophysiological research.” Computational Neurobiology Laboratory, the Salk Institute for Biological Studies, 1998.  
Available: <http://www.cnl.salk.edu/~ica.html>
- [22] B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by v1?,” Vision Research, vol. 37, pp. 3311–3325, 1997.
- [23] D. Pham, “Joint approximate diagonalization of positive definite matrices,” SIAM J. on Matrix Anal. and Appl., vol. 22, no. 4, pp. 1136–1152, 2001.
- [24] D. Pham and J.-F. Cardoso, “Blind separation of instantaneous mixtures of non stationary sources,” IEEE Transactions on Signal Processing, vol. 49, no. 9, pp. 1837–1848, 2001.
- [25] D. Pham and P. Garrat, “Blind separation of a mixture of independent sources through a quasi-maximum likelihood approach,” IEEE Transactions on Signal Processing, vol. 45, no. 7, pp. 1712–1725, 1997.
- [26] M. Zibulevsky, “Relative Newton method for quasi-ML blind source separation,” Journal of Machine Learning Research, 2002, submitted.  
Available: <http://ie.technion.ac.il/~mcib>

[27] M. Zibulevsky, P. Kisilev, Y. Y. Zeevi, and B. A. Pearlmutter, “Blind source separation via multinode sparse representation,” in *Advances in Neural Information Processing Systems 12*, MIT Press, 2002.

[28] M. Zibulevsky and B. A. Pearlmutter, “Blind source separation by sparse decomposition in a signal dictionary,” *Neural Computations*, vol. 13, no. 4, pp. 863–882, 2001.

[29] M. Zibulevsky, B. A. Pearlmutter, P. Bofill, and P. Kisilev, “Blind source separation by sparse decomposition,” in *Independent Components Analysis: Principles and Practice* (S. J. Roberts and R. M. Everson, eds.), Cambridge University Press, 2001.

## APPENDIX A – SMOOTH APPROXIMATION OF THE ABSOLUTE VALUE

We use a parametric family of functions

$$h_0(t) = |t| + \frac{1}{|t|+1}; \quad h(t; \mathbf{I}) = \mathbf{I} h_0\left(\frac{t}{\mathbf{I}}\right), \quad (22)$$

which smoothly approximates  $|t|$  up to an additive constant, where  $\mathbf{I} > 0$  is a smoothing parameter. The corresponding derivatives of  $h$  are given by

$$h'_0(t) = \text{sign}(t) - \frac{\text{sign}(t)}{(|t|+1)^2}; \quad h''_0(t) = \frac{2}{(|t|+1)^3}. \quad (23)$$

This type of nonlinearity has a relatively low computational complexity.

## APPENDIX B – MODIFIED RELATIVE NEWTON STEP

Following [26], we use a diagonal approximation of the second term (8) of the Hessian. Under the assumption of independent and zero mean sources, we have the following zero expectation:

$$\mathbb{E}\{h''(s_m(t))s_i(t)s_j(t)\} = 0 \quad ; \quad m, i \neq j, \quad (24)$$

where  $\mathbf{S} = \mathbf{U}^{(k)}$ . Hence, the off-diagonal elements  $\mathbf{B}_{ij}^m$  in (8) vanish as the sample size  $T$  grows, which yields a further simplification of the second term of the Hessian:

$$\mathbf{B}_{ii}^m = \frac{1}{T} \sum_t h''(\mathbf{u}_m(t)) \mathbf{u}_i(t); \quad i = 1, \dots, N; \quad m = 1, \dots, N, \quad (25)$$

( $\mathbf{u}_m(t)$  are current estimates of the sources).

The diagonal Hessian approximation greatly simplifies the Newton direction computation. Let us pack the diagonal of the  $N^2 \times N^2$  Hessian matrix in (25) into a  $N \times N$  matrix  $\mathbf{D}$ , row by row. The differential of the gradient obtains the form [26]:

$$d\mathbf{G} = \mathbf{H}d\mathbf{W} = d\mathbf{W}^T + \mathbf{D} \odot d\mathbf{W}, \quad (26)$$

where  $\odot$  denotes the Hadamard product (element-wise matrix multiplication). For an arbitrary matrix  $Y$ ,

$$HY = Y^T + D \odot Y, \quad (27)$$

Solution of the Newton system

$$Y^T + D \odot Y = G, \quad (28)$$

requires the solution of  $\frac{1}{2}N(N-1)$  systems of  $2 \times 2$  linear equations

$$\begin{aligned} D_{ij} Y_{ij} + Y_{ji} &= G_{ij} \quad ; \quad i = 1, \dots, N, \quad j = 1, \dots, i-1 \\ D_{ji} Y_{ji} + Y_{ij} &= G_{ji} \end{aligned} \quad (29)$$

in order to find the off-diagonal elements, and  $1 \times 1$  systems

$$D_{ii} Y_{ii} + Y_{ii} = G_{ii} \quad (30)$$

in order to find the diagonal elements. In order to guarantee global convergence, the Newton system is modified by forcing positive eigenvalues [15], [26].

## FOOTNOTES

<sup>1</sup> Complete results are available at <http://visl.technion.ac.il/bron/works/bss/newton/audio.html>.

<sup>2</sup> Complete results are available at <http://visl.technion.ac.il/bron/works/bss/newton/images.html>.