

# Blind source separation using the block-coordinate relative Newton method

Alexander M. Bronstein, Michael M. Bronstein, and Michael Zibulevsky \*

Technion - Israel Institute of Technology, Department of Electrical Engineering,  
32000 Haifa, Israel  
{alexbron, bronstein}@ieee.org, mzib@ee.technion.ac.il

**Abstract.** Presented here is a generalization of the modified relative Newton method, recently proposed in [1] for quasi-maximum likelihood blind source separation. Special structure of the Hessian matrix allows to perform block-coordinate Newton descent, which significantly reduces the algorithm computational complexity and boosts its performance. Simulations based on artificial and real data show that the separation quality using the proposed algorithm outperforms other accepted blind source separation methods.

## 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 an  $N$ -channel sensor signal arises from  $N$  unknown scalar source signals, linearly mixed by an unknown  $N \times N$  invertible matrix  $A$ :  $x(t) = As(t)$ . When a finite sample  $t = 1, \dots, T$  is given, the latter can be rewritten in matrix notation as  $X = AS$ , where  $X$  and  $S$  are  $N \times T$  matrices containing  $s_i(t)$  and  $x_i(t)$  as the rows. In the 2D case, images can be thought of as one-dimensional vectors. Our goal is to estimate the unmixing matrix  $W = A^{-1}$ , which yields the source estimate  $s(t) = Wx(t)$ .

Let us assume that the sources  $s_i(t)$  are zero-mean i.i.d. and independent on each other. The minus log likelihood of the observed data is given by

$$\ell(X; W) = -\log |W| + \frac{1}{T} \sum_{i,t} h_i(W_i x(t)), \quad (1)$$

where  $W_i$  is the  $i$ -th row of  $W$ ,  $h_i(s) = -\log p_i(s)$ , and  $p_i(s)$  is the PDF of the  $i$ -th source. We will henceforth assume for simplicity that  $h_i(s) = h(s)$  for all the sources, although the presented method is also valid in the general case. Many times, when  $h_i$  are not equal to the exact minus log PDFs of the sources, minimization of (1) leads to a consistent estimator, known as *quasi maximum likelihood* (QML) estimator.

---

\* This research has been supported by the HASSIP Research Network Program HPRN-CT-2002-00285, sponsored by the European Commission, and by the Ollendorff Minerva Center.

QML estimation is convenient when the source PDF is unknown, or not well-suited for optimization. For example, when the sources are sparse or sparsely representable, the absolute value function, or its smooth approximation is a good choice for  $h(s)$  [2, 3]. We use a parametric family of functions

$$h_\lambda(s) = |s| + \frac{1}{|s| + \lambda^{-1}} \quad (2)$$

with a smoothing parameter  $\lambda > 0$ . Up to an additive constant,  $h_\lambda(s) \rightarrow |s|$  when  $\lambda \rightarrow 0^+$ . Evaluation of this type of non-linearity and its first- and second-order derivatives has relatively low complexity.

The widely accepted *natural gradient* method shows poor convergence when the approximation of the absolute value becomes too sharp. In order to overcome this obstacle, a relative Newton approach was recently proposed in [1], which is an improvement of the Newton method used in [4]. 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.

## 2 Relative Newton algorithm

The following *relative optimization* (RO) algorithm for minimization of the QML function (1) was used in [5]:

### *Relative optimization algorithm*

1. Start with initial estimates of the unmixing matrix  $W^{(0)}$  and the sources  $X^{(0)} = W^{(0)}X$ .
2. For  $k = 0, 1, 2, \dots$ , until convergence
  3. Start with  $W^{(k+1)} = I$ .
  4. Using an unconstrained optimization method, find  $W^{(k+1)}$  such that  $\ell(X^{(k)}; W^{(k+1)}) < \ell(X^{(k)}; I)$ .
  5. Update source estimate:  $X^{(k+1)} = W^{(k+1)}X^{(k)}$ .
6. End

The use of a single gradient descent iteration on Step 4 leads to the natural (relative) gradient method [6, 7], whereas the use of a Newton iteration leads to the relative Newton method [1].

### 2.1 Gradient and Hessian of $\ell(X; W)$

The use of the Newton method on Step 4 of the RO algorithm requires the knowledge of the Hessian of  $\ell(X; W)$ . Since  $\ell(X; W)$  is a function of a matrix argument, its gradient w.r.t.  $W$  is also a matrix

$$G(W) = \nabla_W \ell(X; W) = -W^{-T} + \frac{1}{T} h'(WX) X^T, \quad (3)$$

where  $h'$  is applied element-wise to  $WX$ .

The Hessian of  $\ell(X; W)$  can be thought as a fourth-order tensor  $\mathcal{H}$ , which is inconvenient in practice. Alternatively, one can convert the matrix  $W$  into an  $N^2$ -long column vector  $w = \text{vec}(W)$  by row-stacking. Using this notation, the Hessian is an  $N^2 \times N^2$  matrix, which can be found from the differential of  $g(w)$  (see [1] for derivation). The  $k$ -th column of the Hessian of the log-determinant term of  $\ell(X; W)$  is given by

$$H^k = \text{vec}(A^j A_i), \quad (4)$$

where  $A = W^{-1}$ , and  $A_i, A^j$  are its  $i$ -th row and  $j$ -th column, respectively, and  $k = (i-1)N + j$ . The Hessian of the second term of  $\ell(X; W)$  containing the sum is a block-diagonal matrix, whose  $m$ -th block is an  $N \times N$  matrix of the form

$$B^m = \frac{1}{T} \sum_t h''(W_m x(t)) x(t) x^T(t). \quad (5)$$

## 2.2 The modified Relative Newton step

At each relative Newton iteration, the Hessian is evaluated for  $W = I$ , which simplifies the Hessian of the log-determinant term in (4) to

$$H^k = \text{vec}(e_i e_j^T), \quad (6)$$

where  $e_i$  is the standard basis vector containing 1 at the  $i$ -th coordinate. The second term (5) becomes

$$B^m = \frac{1}{T} \sum_t h''(x_m(t)) x(t) x^T(t). \quad (7)$$

At the solution point,  $x(t) = s(t)$ , up to scale and permutation. For a sufficiently large sample, the sum approaches the corresponding expected value yielding  $B^m \approx \mathbf{E}\{h''(x_m) x x^T\}$ . Invoking the assumption that  $s_i(t)$  are mutually-independent zero-mean i.i.d. processes,  $B^m$  become approximately diagonal.

Using this approximation of the Hessian, the modified (fast) relative Newton method is obtained. The diagonal approximation significantly simplifies both Hessian evaluation and Newton system solution. Computation of the diagonal approximation requires about  $N^2 T$  operations, which is of the same order as the gradient computation. Approximate solution of the Newton system separates to solution of  $\frac{1}{2} N(N-1)$  symmetric systems of size  $2 \times 2$

$$\begin{pmatrix} Q_{ij} & 1 \\ 1 & Q_{ji} \end{pmatrix} \begin{pmatrix} D_{ij} \\ D_{ji} \end{pmatrix} = - \begin{pmatrix} G_{ij} \\ G_{ji} \end{pmatrix}, \quad (8)$$

for the off-diagonal elements ( $i \neq j$ ), and  $N$  additional linear equations

$$Q_{ii} D_{ii} + D_{ii} = -G_{ii} \quad (9)$$

for the diagonal elements, where  $D$  is the  $N \times N$  Newton direction matrix,  $G$  is the gradient matrix, and  $Q$  is an  $N \times N$  matrix, in which the Hessian diagonal is packed row-by-row.

In order to guarantee global convergence, the  $2 \times 2$  systems are modified by forcing positive eigenvalues [1]. Approximate Newton system solution requires 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 method

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 [8].

In our problem, the Hessian is approximately separable with respect to the pairs of symmetric elements of  $W$ . This brings us to the idea of applying the Newton step block-coordinately on these pairs. As it will appear from the complexity analysis, the relative cost of the nonlinearity computation becomes dominant in this case, therefore, we can do one step further and use pair-wise symmetric blocks of larger size. The matrix  $W$  can be considered as consisting of  $M = N/K$  blocks of size  $K \times K$ ,

$$W = \begin{pmatrix} W_{11} & W_{12} & \dots & W_{1M} \\ W_{21} & W_{22} & \dots & W_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M1} & W_{M2} & \dots & W_{MM} \end{pmatrix} \quad (10)$$

The *block-coordinate* modified relative Newton step (as opposed to the *full* modified relative Newton step described before) 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:

#### ***Block-coordinate relative Newton algorithm***

1. Start with initial estimates of the unmixing matrix  $W^{(0)}$  and the sources  $X^{(0)} = W^{(0)}X$ .
2. For  $k = 0, 1, 2, \dots$ , until convergence
  3. For  $i = 1, \dots, K$ 
    4. For  $j = 1, \dots, K$ 
      5. Start with  $W^{(k+1)} = I$ .

6. Update the blocks  $W_{ij}$  and  $W_{ji}$  using one block-coordinate relative Newton iteration to find  $W^{(k+1)}$  such that  $\ell(X^{(k)}; W^{(k+1)}) < \ell(X^{(k)}; I)$ .
7. Efficiently update the source estimate:  $X^{(k+1)} = W^{(k+1)}X^{(k)}$ .
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  $\sum h(W_i x(t))$ .

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

$$W = \begin{pmatrix} I_{K \times K} & & & W_{ij} \\ & I_{K \times K} & & \\ W_{ji} & & \ddots & \\ & & & I_{K \times K} \end{pmatrix} \quad (11)$$

It can be easily shown that the computation of the determinant of  $W$  having this form can be reduced to

$$\det W = \det \begin{pmatrix} I & W_{ij} \\ W_{ji} & I \end{pmatrix} \quad (12)$$

and carried out in  $2K^3$  operations. Similarly, the computation of the gradient requires applying  $h'$  to the updated  $2K \times T$  stripe of  $WX$  and multiplying the result by the corresponding  $2K \times T$  stripe of  $X^T$ . In addition, the gradient requires inversion of  $W$ . When  $i \neq j$ , the inverse matrix has the form

$$W^{-1} = \begin{pmatrix} I & & & \\ A_{ii} & A_{ij} & & \\ & I & & \\ A_{ji} & A_{jj} & & \\ & & & I \end{pmatrix}, \quad (13)$$

where the  $K \times K$  blocks  $A_{ii}$ ,  $A_{ij}$ ,  $A_{ji}$  and  $A_{jj}$  are obtained from

$$\begin{pmatrix} A_{ii} & A_{ij} \\ A_{ji} & A_{jj} \end{pmatrix} = \begin{pmatrix} I & W_{ij} \\ W_{ji} & I \end{pmatrix}^{-1}, \quad (14)$$

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

### 3.1 Computational complexity

For convenience, we denote as  $\alpha$ ,  $\alpha'$  and  $\alpha''$  the number of operations required for the computation of the non-linearity  $h$  and its derivatives  $h'$  and  $h''$ , respectively. A reasonable estimate of these constants for  $h$  given in (2) is  $\alpha = 6, \alpha' = 2, \alpha'' = 2$  [9]. We will also denote  $\beta = \alpha + \alpha' + \alpha''$ . A single block-coordinate relative Newton inner iteration involves computation of the cost function, its gradient and Hessian, whose respective complexities are  $2(K^2T + K^3 + \alpha KT)$ ,  $2(K^2T + K^3 + \alpha' KT)$  and  $2(K^2T + (\alpha'' + 1)KT)$ . In order to compute the Newton direction,  $K$  systems of equations of size  $2 \times 2$  have to be solved, yielding in total solution of systems per outer iteration, independent of  $K$ . Other operations have negligible complexity. Therefore, a single block-coordinate outer Newton iteration will require about  $N^2T(3 + (\beta + 1)/K)$  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 its runtime is shortened by the factor  $\gamma > 1 + (\beta + 1)/3K$  compared to the full relative Newton method.

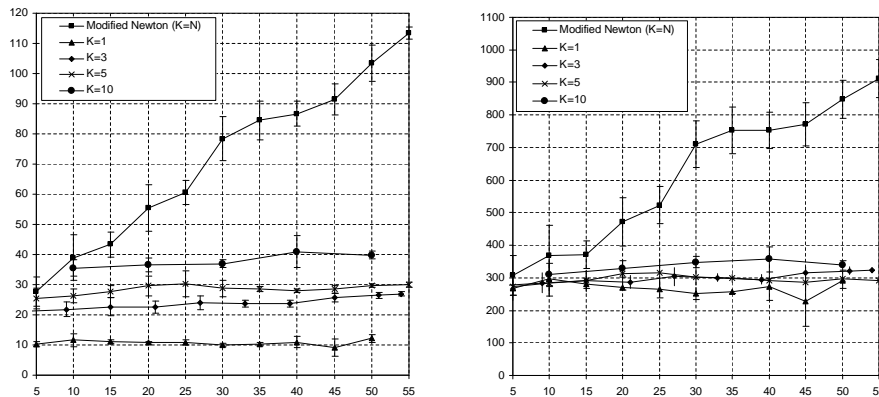
## 4 Numerical results

For numerical experiments, three data sets were used: sparse normal signals generated using the MATLAB function `sprandn`, 50,000 samples from instrumental and vocal music recordings sampled at 11025 Hz, and natural images. In all the experiments, the sources were artificially mixed using an invertible random matrix with uniform i.i.d. elements. The modified relative Newton algorithm with backtracking line search was used, stopped after the gradient norm reached  $10^{-10}$ . Data sets containing audio signals and images were not originally sparse, and thus not the corresponding mixtures. Short time Fourier transform (STFT) and discrete derivative were used to sparsify the audio signals and the images, respectively, as described in [10, 11, 2, 3]. In Table 1, the separation quality (in terms of the signal-to-interference ratio (SIR) in dB units) of the relative Newton method is compared with that of stochastic natural gradient (Infomax) [7, 6, 12], Fast ICA [13, 14] and JADE [15]. We should note that without the sparse representation stage, all algorithms produced very poor separation results. Figure 2 depicts the convergence of the full modified relative Newton algorithm and its block-coordinate version for different block sizes, with audio signals and images. Complete comparison can be found at <http://visl.technion.ac.il/bron/works/bss/newton>.

The block-coordinate algorithm (with block size  $K = 1, 3, 5$  and  $10$ ) was compared to the full modified relative Newton algorithm ( $K = N$ ) on problems of different size ( $N$  from 3 to 50 in integer multiplies of  $K$ ;  $T = 10^3$ ) with the sparse sources. The total number of the cost function, its gradient and Hessian evaluations were recorded and used for complexity computation. 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 1, left). 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 1 (right) the complexity normalized by the

**Table 1.** Separation quality (best and worst SIR in dB) of sparse signals, audio signals and images.

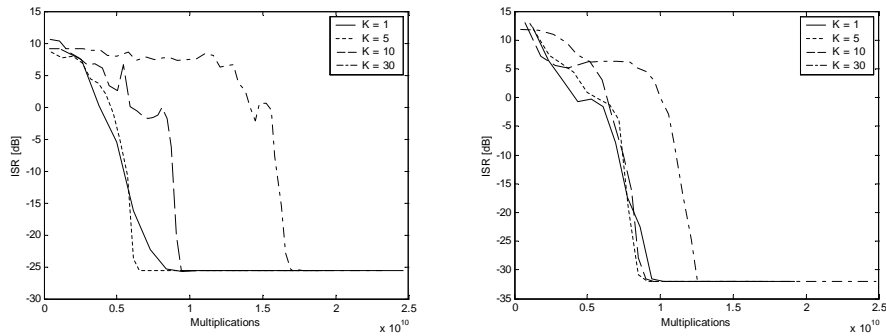
SIR	Newton	InfoMax	FastICA	JADE
Sparse	172.98 ÷ 167.99	34.35 ÷ 18.64	23.82 ÷ 21.89	26.78 ÷ 21.89
Audio	46.68 ÷ 25.72	37.34 ÷ 23.35	25.15 ÷ 2.11	25.78 ÷ 9.02
Images	57.35 ÷ 31.74	38.52 ÷ 25.66	30.54 ÷ 19.75	32.35 ÷ 27.85

**Fig. 1.** Average number of outer iterations (left) and the normalized complexity (right) vs. the number of sources  $N$  for different block sizes  $K$ .

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  $N = 55$ .

## 5 Conclusion

We presented a block-coordinate version of the relative Newton algorithm for QML blind source separation introduced in [1]. In large problems, we observed a nearly three-fold reduction of the computational complexity of the modified Newton step by using the block-coordinate approach. The use of an accurate approximation of the absolute value nonlinearity in the QML function leads to accurate separation of sources, which have sparse representation. Simulations showed that from the point of view of the obtained SIR, such optimization appears to outperform other accepted algorithms for 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 importance for practical applications.



**Fig. 2.** Convergence of the the block-coordinate relative Newton method for audio sources (left) and images (right) using blocks of different size  $K$  ( $K = 30$  corresponds to full relative Newton).

## References

1. Zibulevsky, M.: Sparse source separation with relative Newton method. In: Proc. ICA2003. (2003) 897–902
2. Zibulevsky, M., Pearlmutter, B.A.: Blind source separation by sparse decomposition in a signal dictionary. *Neural Comp.* **13** (2001) 863–882
3. Zibulevsky, M., Pearlmutter, B.A., Bofill, P., Kisilev, P.: Blind source separation by sparse decomposition. In Roberts, S.J., Everson, R.M., eds.: *Independent Components Analysis: Principles and Practice*. Cambridge University Press (2001)
4. Pham, D., Garrat, P.: Blind separation of a mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Trans. Sig. Proc.* **45** (1997) 1712–1725
5. Bell, A.J., Sejnowski, T.J.: An information maximization approach to blind separation and blind deconvolution. *Neural Comp.* **7** (1995) 1129–1159
6. S. Amari, A.C., Yang, H.H.: A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems* **8** (1996)
7. Cichocki, A., Unbehauen, R., Rummert, E.: Robust learning algorithm for blind separation of signals. *Electronics Letters* **30** (1994) 1386–1387
8. Grippo, L., Sciandrone, M.: Globally convergent block-coordinate techniques for unconstrained optimization. *Optimization Methods and Software* **10** (1999) 587–637
9. Bronstein, A.M., Bronstein, M.M., Zibulevsky, M.: Block-coordinate relative Newton method for blind source separation. Technical Report 445, Technion, Israel (2003)
10. Bofill, P., Zibulevsky, M.: Underdetermined blind source separation using sparse representations. *Sig. Proc.* **81** (2001) 2353–2362
11. Bronstein, A.M., Bronstein, M.M., Zibulevsky, M., Zeevi, Y.Y.: Separation of reflections via sparse ICA. In: Proc. IEEE ICIP. (2003)
12. Makeig, S.: ICA toolbox for psychophysiological research (1998)  
Online: <http://www.cnl.salk.edu/ica.html>.
13. Hyvärinen, A.: The Fast-ICA MATLAB package (1998)  
Online: <http://www.cis.hut.fi/aapo>.
14. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. Neural Net.* **10** (1999) 626–634
15. Cardoso, J.F.: JADE for real-valued data (1999)  
Online: <http://sig.enst.fr:80/cardoso/guidesepsou.html>.