4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), April 2003, Nara, Japan

BLIND SOURCE SEPARATION WITH RELATIVE NEWTON METHOD

Michael Zibulevsky

Department of Electrical Engineering, Technion, Haifa 32000, Israel
Email: mzib@ee.technion.ac.il

ABSTRACT

We study a relative optimization framework for the quasi-maximum likelihood blind source separation and relative Newton method as its particular instance. Convergence of the Newton method is stabilized by the line search and by the modification of the Hessian, which forces its positive definiteness. The structure of the Hessian allows fast approximate inversion. We demonstrate the efficiency of the presented approach on example of sparse sources. The nonlinearity in this case is based on smooth approximation of the absolute value function. Sequential optimization with the gradual reduction of the smoothing parameter leads to the super-efficient separation.

1. INTRODUCTION

Several Newton-like methods for blind source separation have been studied in the literature. They are based on negentropy approximation with orthogonality constraint [1], cumulant model [2, 3] and joint diagonalization of correlation matrices [4, 5, 6]. In this work we study a Newton method for quasi-maximum likelihood source separation [7, 8] in batch mode, without orthogonality constraint. This criterion provides improved separation quality [9, 10], and is particularly useful in separation of sparse sources.

Consider the blind source separation problem, where an N-channel sensor signal $x(t)$ arises from N unknown scalar source signals $s_i(t)$, $i = 1, .., N$, linearly mixed together by an unknown $N \times N$ matrix A

$$x(t) = As(t) \tag{1}$$

We wish to estimate the mixing matrix A and the N-dimensional source signal $s(t)$. In the discrete time case $t = 1, 2, \ldots, T$ we use matrix notation $X = AS$, where X and S are $N \times T$ matrices with the signals $x_i(t)$ and $s_i(t)$ in the corresponding rows. We also denote the unmixing matrix $W = A^{-1}$.

When the sources are *i.i.d*, stationary and white, the normalized minus-log-likelihood of the observed data X is

$$L(W; X) = -\log|\det W| + \frac{1}{T}\sum_{i,t} h\big(W_i x(t)\big), \tag{2}$$

The author would like to acknowledge support for this project by the Ollendorff Minerva Center and by the Israeli Ministry of Science

where W_i is i-th row of W, $h(\cdot) = -\log f(\cdot)$, and $f(\cdot)$ is the probability density function (pdf) of the sources. Consistent estimator can be obtained by minimization of (2), also when $h(\cdot)$ is not exactly equal to $-\log f(\cdot)$. Such *quasi-ML estimation* is practical when the source pdf is unknown, or is 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 $h(\cdot)$ [11, 12, 13, 14, 15, 16]. Here we will use a family of convex smooth approximations to the absolute value

$$h_1(c) = |c| - \log(1 + |c|) \tag{3}$$
$$h_\lambda(c) = \lambda h_1(c/\lambda) \tag{4}$$

with λ a proximity parameter: $h_\lambda(c) \to |c|$ as $\lambda \to 0^+$. Widely accepted natural gradient method does not work well when the approximation of the absolute value becomes too sharp. In this work we suggest the relative Newton method, which overcomes this obstacle, and provides fast and very accurate separation of sparse sources.

2. RELATIVE OPTIMIZATION (RO) ALGORITHM

We consider the following algorithm for minimization of the quasi-ML function (2)

1. Start with an initial estimate W_1 of the separation matrix;
2. **For** $k = 1, 2, ...,$ until convergence
3. Compute current source estimate $U_k = W_k X$;
4. Starting with $V = I$ (identity matrix), compute V_{k+1} producing one or few steps of a conventional optimization method, which sufficiently decreases the function $L(V; U_k)$;
5. Update the estimated separation matrix $W_{k+1} = V_{k+1} W_k$;
6. **End**

The relative (natural) gradient method [17, 18, 19] is a particular instance of this approach, when a standard gradient descent step is used in p.4. The following remarkable

property of the relative gradient is also preserved in general case: *given current source estimate U, the progress of the method does not depend on the original mixing matrix.* This means that even nearly ill-conditioned mixing matrix influences the convergence of the method not more than a starting point Convergence analysis of the RO-algorithm is presented in [20]. In the following we will use a Newton step in p.4 of the method.

3. HESSIAN EVALUATION

The likelihood $L(W; X)$ is a function of a matrix argument W. The corresponding gradient is also a matrix

$$G(W) = \nabla L(W; X) = -W^{-T} + \frac{1}{T} h'(WX) X^T, \quad (5)$$

where $h'(WX)$ is a matrix with the elements $h'\big((WX)_{ij}\big)$. The Hessian of $L(W; X)$ is a linear mapping $\mathcal{H}$ defined via the differential of the gradient

$$dG = \mathcal{H} dW \quad (6)$$

We can also express the Hessian in standard matrix form converting W into a long vector $w = \mathrm{vec}(W)$ using row stacking. We will denote the reverse conversion $W = \mathrm{mat}(w)$. Let

$$\hat{L}(w, X) \equiv L(\mathrm{mat}(w), X) \quad (7)$$

so that the gradient

$$g(w) = \nabla \hat{L}(w; X) = \mathrm{vec}(G(W)) \quad (8)$$

Then

$$dg = H dw \quad (9)$$

where H is $N^2 \times N^2$ Hessian matrix. We also have

$$dg = \mathrm{vec}(dG) \quad (10)$$

3.1. Hessian of $-\log\det W$

Using the expression

$$d(W^{-1}) = -W^{-1}(dW)W^{-1},$$

which follows from the equality

$$0 = d(WW^{-1}) = (dW)W^{-1} + W d(W^{-1}),$$

we obtain the differential of the first term in (5)

$$dG = d(W^{-T}) = -A^T (dW^T) A^T, \quad (11)$$

where $A = W^{-1}$. Particular element of the differential

$$dG_{ij} = -A_i (dW^T) A^j = -\mathrm{Trace} A^j A_i (dW^T), \quad (12)$$

where A_i and A^j are i-th row and j-th column of A respectively. Comparing this with (9) and (10), we conclude that the k-th row of H, where $k = (i-1)N + j$, contains the matrix $A^j A_i$ stacked column-wise

$$H_k = \mathrm{vec}^T (A^j A_i)^T \quad (13)$$

3.2. Hessian of $\frac{1}{T}\sum_{m,t} h\big(W_m x(t)\big)$

It is easy to see that the Hessian of the second term in $\hat{L}(w, X)$ is a block-diagonal matrix with the following $N \times N$ blocks

$$B^m = \frac{1}{T} \sum_t h''(W_m x(t)) x(t) x^T(t), \quad m = 1, .., N \quad (14)$$

4. NEWTON METHOD

Newton method is an efficient tool of unconstrained optimization. It often converges fast and provides quadratic rate of convergence. However, its iteration may be costly, because of the necessity to compute the Hessian matrix and solve the corresponding system of equations. In the next section we will see that this difficulty can be overcome using the relative Newton method.

First, let us consider a standard Newton approach, in which the direction is given by solution of the linear equation

$$Hy = -\nabla \hat{L}(w; X) \quad (15)$$

where $H = \nabla^2 \hat{L}(w; X)$ is the Hessian of (7). In order to guarantee descent direction in the case of nonconvex objective function, we use modified Cholessky factorization[1] [21], which automatically finds such a diagonal matrix R, that the matrix $H + R$ is positive definite, and provides a solution to the modified system

$$(H + R)y = -\nabla \hat{L}(w; X) \quad (16)$$

After the direction y is found, the new iterate w^+ is given by

$$w^+ = w + \alpha y \quad (17)$$

where the step size α is determined by exact line search

$$\alpha = \arg\min_\alpha \hat{L}(w + \alpha y; X) \quad (18)$$

or by backtracking line search [21].

Backtracking line search

$\alpha := 1$
While $\hat{L}(w + \alpha y; X) > \hat{L}(w; X) + \beta\alpha \nabla \hat{L}(w; X)^T d$
    $\alpha := \gamma\alpha$
end

The use of the line search guarantees monotonic decrease of the objective function at every iteration. In our computations, we use backtracking line search with the constants $\beta = \gamma = 0.3$.

[1] We use the MATLAB code of modified Cholessky factorization by Brian Borchers, available at http://www.nmt.edu/~borchers/ldlt.html

Computational complexity. The Hessian is a $N^2 \times N^2$ matrix; its computation requires N^4 operations in (13) and N^3T operations in (14). Solution of the Newton system (16) using modified Cholessky decomposition, requires $N^6/6$ operations for decomposition and N^4 operations for back/forward substitution. Totally, we need

$$2N^4 + N^3T + N^6/6$$

operations for one Newton step. Comparing this to the cost of the gradient evaluation (5), which is equal to N^2T, we conclude that Newton step costs about N gradient steps when the number of sources is small (say, up to 20). Otherwise, the third term become dominating, and the complexity grows as N^6.

5. RELATIVE NEWTON METHOD

In order to make the Newton algorithm invariant to the value of mixing matrix, we use the relative Newton method, which is a particular instance of the RO-algorithm. This approach simplifies the Hessian computation and the solution of the Newton system.

5.1. Basic relative Newton step

The optimization in p.4 of the RO-algorithm is produced by a single Newton-like iteration with exact or backtracking line search. The Hessian of $L(I;U)$ has a special structure, which permits fast solution of the Newton system. First, the Hessian of $-\log\det W$ given by (13), becomes very simple and sparse, when $W = A = I$: each row of H

$$H_k = \mathrm{vec}^T(e_i e_j^T), \tag{19}$$

contains only one non-zero element, which is equal to 1. Here e_j is an N-element standard basis vector, containing 1 at j-th position. Remaining part of the Hessian is block-diagonal. There are various techniques for solving sparse symmetric systems. For example, one can use sparse modified Cholessky factorization for direct solution, or alternatively, conjugate gradient-type methods, possibly preconditioned by incomplete Cholessky factor, for iterative solution. In both cases, Cholessky factor is often not as sparse as the original matrix, but it becomes sparser, when appropriate matrix permutation is applied before factorization (see for example MATLAB functions CHOLINC and SYMAMD.)

5.2. Fast relative Newton step

Further simplification of the Hessian is obtained by considering its structure at the solution point $U_k = S$. The elements of m-th block of the second term of $\nabla^2 L(I;S)$ given by (14), are equal to

$$B_{ij}^m = \frac{1}{T}\sum_t h''(s_m(t))s_i(t)s_j(t), \;\; i,j = 1,..,N.$$

When the sources are independent and zero mean, we have the following zero expectation

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

hence the off-diagonal elements B_{ij}^m converge to zero as sample size grows. Therefore we use a diagonal approximation of this part of the Hessian

$$B_{ii}^m = \frac{1}{T}\sum_t h''(u_m(t))u_i^2(t), i = 1,..,N; \;\; m = 1,..,N, \tag{20}$$

where $u_m(t)$ are current estimates of the sources. In order to solve the simplified Newton system, let us return to the matrix-space form (6) of the Hessian operator. Let us pack the diagonal of the Hessian given by (20) into $N \times N$ matrix D, row-by-row. Taking into account that $A = I$ in (11), we will obtain the following expression for the differential of the gradient

$$dG = \mathcal{H}dW = dW^T + D \odot dW, \tag{21}$$

where "$\odot$" denotes element-wise multiplication of matrices. For an arbitrary matrix Y

$$\mathcal{H}Y = Y^T + D \odot Y. \tag{22}$$

In order to solve the Newton system

$$Y^T + D \odot Y = G \tag{23}$$

we need to solve $N(N-1)/2$ systems of size 2×2 with respect to Y_{ij} and Y_{ji}

$$\begin{aligned} D_{ij}Y_{ij} + Y_{ji} &= G_{ij}, \;\; i = 1,..N; \; j = 1,..,i-1 \\ D_{ji}Y_{ji} + Y_{ij} &= G_{ji} \end{aligned} \tag{24}$$

The diagonal elements Y_{ii} can be found directly from the set of single equations

$$D_{ii}Y_{ii} + Y_{ii} = G_{ii} \tag{25}$$

In order to guarantee descent direction and avoid saddle points, we modify the Newton system (24), changing the sign of the negative eigenvalues [21]. Namely, we compute analytically the eigenvectors and the eigenvalues of 2×2 matrices

$$\begin{pmatrix} D_{ij} & 1 \\ 1 & D_{ji} \end{pmatrix},$$

invert the sign of the negative eigenvalues, and force small eigenvalues to be above some threshold (say, 10^{-8} of the

maximal one in the pair). Than we solve the modified system, using the eigenvectors already obtained and the modified eigenvalues[2].

Computational complexity. Computing the diagonal of the Hessian by (20) requires N^2T operations, which is equal to the cost of the gradient computation. Solution cost of the set of 2x2 linear equations (24) is about $15N^2$ operations, which is negligible compared to the gradient cost.

6. SEQUENTIAL OPTIMIZATION

When the sources are sparse, the quality of separation greatly improves with reduction of smoothing parameter λ in the absolute value approximation (4). On the other hand, the optimization of the likelihood function becomes more difficult for small λ. Therefore, we use sequential optimization with gradual reduction of λ. Denote

$$L(W; X, \lambda) = -\log|\det W| + \frac{1}{T}\sum_{i,t} h_\lambda\big(W_i x(t)\big), \quad (26)$$

where $h_\lambda(\cdot)$ is given by (3–4).

Sequential optimization algorithm

1. Start with λ_1 and W_1
2. **For** $k = 1, 2, .., K$
3. Compute current source estimate $U_k = W_k X$;
4. Starting with $V = I$
   find $V_{k+1} = \arg\min_V L(V, U_k, \lambda_k)$
5. Update the estimated separation matrix
   $W_{k+1} = V_{k+1} W_k$;
6. Update the smoothing parameter $\lambda_{k+1} = \mu\lambda_k$
7. **End**

In our computations we choose the parameters $\lambda_1 = 1$ and $\mu = 0.01$. Note that p.4 includes the whole loop of unconstrained optimization, which can be performed, for example, by the relative Newton method.

[2] After completing this work we have been aware that the Newton equations similar to (24) were obtained by Pham and Garat [7] using slightly different considerations. This algorithm was not used in practice because of possibility of convergence to spurious solutions. In our work we overcome this difficulty by introducing the line search and by forcing positive definiteness of the Hessian.

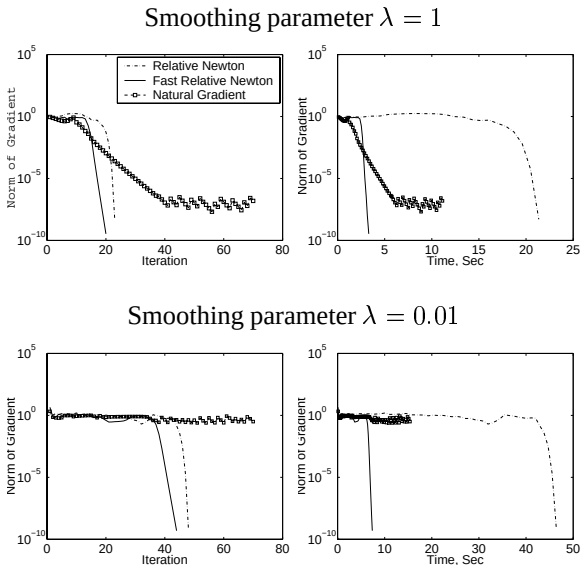

Fig. 1. Separation of artificial sparse data with 5 mixtures by 10k samples. Relative Newton with exact Hessian – dashed line, fast relative Newton – continuous line, natural gradient in batch mode – squares.

7. COMPUTATIONAL EXPERIMENTS

Two data sets were used. First group of sources was artificial sparse data with Bernoulli-Gaussian distribution

$$f(s) = p\delta(s) + (1-p)\frac{1}{\sqrt{2\pi\sigma^2}}\exp(-s^2/2\sigma^2),$$

generated by the MATLAB function SPRANDN. We used the parameters $p = 0.5$ and $\sigma = 1$. The second group of sources were four natural images from [22]. The mixing matrix was generated randomly with uniform *i.i.d.* entries.

In all experiments we used backtracking line search with the constants $\beta = \gamma = 0.3$. Figure 1 shows typical progress of different methods applied to the artificial data with 5 mixtures of 10k samples. The fast relative Newton method converges in about the same number of iterations as the relative Newton with exact Hessian, but significantly outperforms it in time. Natural gradient in batch mode requires much more iterations, and has a difficulty to converge when the smoothing parameter λ in (4) becomes too small.

In the second experiment, we demonstrate the advantage of the batch-mode quasi-ML separation, when dealing with sparse sources. We compared the the fast relative Newton method with stochastic natural gradient [17, 18, 19], Fast ICA [1] and JADE [23]. All three codes were obtained from public web sites [24, 25, 26]. Stochastic natural gradient and Fast ICA used $\tanh$ nonlinearity. Figure 2 shows separation of artificial stochastic sparse data: 5 sources of 500 samples, 30 simulation trials. The quality of separation

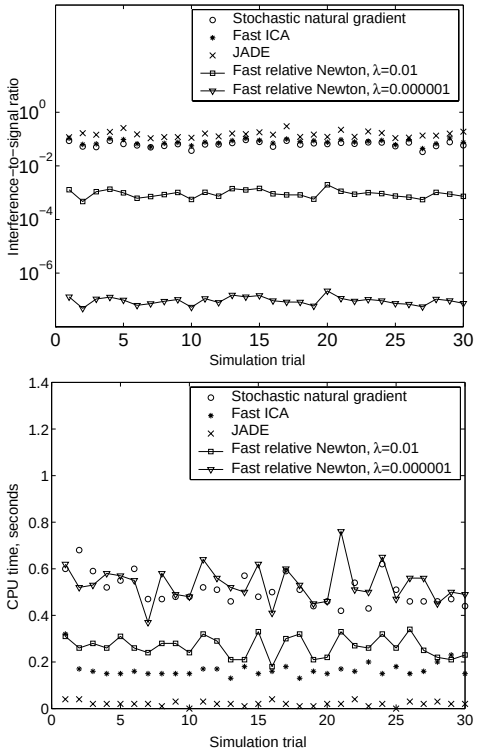

Fig. 2. Separation of stochastic sparse data. Top – interference-to-signal ratio, bottom – CPU time.

is measured by interference-to-signal ratio (ISR) in amplitude units. As we see, fast relative Newton significantly outperforms other methods, providing practically ideal separation with the smoothing parameter $\lambda = 10^{-6}$. Timing is of about the same order for all the methods, except of JADE, which is known to be much faster with relatively small matrices.

In the third experiment, we separated four natural images [22], presented in Figure 3. Sparseness of images can be achieved via various wavelet-type transforms [14, 15, 16], but even simple differentiation can be used for this purpose, since natural images often have sparse edges. Here we used the stack of horizontal and vertical derivatives of the mixture images as an input to separation algorithms. Figure 4 shows the separation quality achieved by stochastic natural gradient, Fast ICA, JADE and the fast relative Newton method. Like in the previous experiment, our method provides practically ideal separation with $\lambda = 10^{-6}$, achieving ISR of about 10^{-7}. It outperforms the other methods by several orders of magnitude.

8. CONCLUSIONS

We have studied a relative optimization framework for quasi-ML blind source separation, and the relative Newton method as its particular instance. Efficient approximate solution of the corresponding Newton system provides gradient-type computational cost of the Newton iteration.

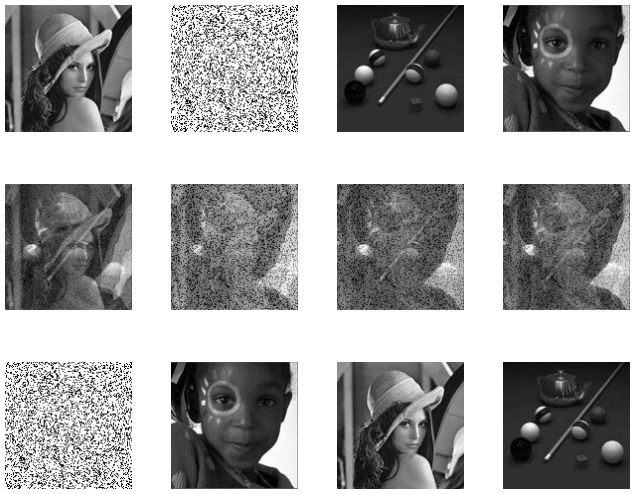

Fig. 3. Separation of images with preprocessing by differentiation. Top – sources, middle – mixtures, bottom – separated by the fast relative Newton method

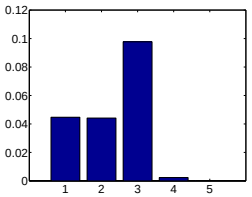

Fig. 4. Interference-to-signal ratio (ISR) of image separation: 1 – stochastic natural gradient; 2 – Fast ICA; 3 – JADE; 4-5 – the fast relative Newton with λ equal to 10^{-2} and 10^{-6}, respectively. Bar 5 is not visible because of very small ISR, of order 10^{-7}.

Experiments with sparsely representable artificial data and natural images show that quasi-ML separation is practically perfect when the nonlinearity approaches the absolute value function. The corresponding optimization problem is solved efficiently by the relative Newton method using sequential optimization with gradual reduction of smoothing parameter.

Currently we are conducting more experiments with non-sparse source distributions and various kinds of nonlinearities. Preliminary results confirm fast convergence of the relative Newton method.

9. REFERENCES

[1] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transac-*

tions on Neural Networks, vol. 10, no. 3, pp. 626–634, 1999.

[2] T. Akuzawa and N.Murata, "Multiplicative nonholonomic Newton-like algorithm," *Chaos, Solitons and Fractals*, vol. 12, p. 785, 2001.

[3] T. Akuzawa, "Extended quasi-Newton method for the ICA," tech. rep., Laboratory for Mathematical Neuroscience, RIKEN Brain Science Institute, 2000. http://www.mns.brain.riken.go.jp/˜akuzawa/publ.html.

[4] D. Pham, "Joint approximate diagonalization of positive definite matrices," *SIAM J. on Matrix Anal. and Appl.*, vol. 22, no. 4, pp. 1136–1152, 2001.

[5] 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.

[6] M. Joho and K. Rahbar, "Joint diagonalization of correlation matrices by using Newton methods with application to blind signal separation," *SAM 2002*, 2002. http://www.phonak.uiuc.edu/˜joho/research/publications/sam_2002_2.pdf.

[7] 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.

[8] 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.

[9] J.-F. Cardoso, "On the performance of orthogonal source separation algorithms," in *EUSIPCO*, (Edinburgh), pp. 776–779, Sept. 1994.

[10] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proceedings of the IEEE*, vol. 9, pp. 2009–2025, Oct. 1998.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Transactions on Signal Processing*, vol. 44, no. 12, pp. 3017–3030, 1996.

[20] M. Zibulevsky, "Relative Newton method for quasi-ML blind source separation," *Journal of Machine Learning Research*, 2002, submitted. http://ie.technion.ac.il/˜mcib/.

[21] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. New York: Academic Press, 1981.

[22] A. Cichocki, S. Amari, and K. Siwek, "ICALAB toolbox for image processing – benchmarks," 2002. http://www.bsp.brain.riken.go.jp/ICALAB/ICALABImageProc/benchmarks/.

[23] J.-F. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157–192, 1999.

[24] S. Makeig, "ICA toolbox for psychophysiological research." Computational Neurobiology Laboratory, the Salk Institute for Biological Studies, 1998. http://www.cnl.salk.edu/˜ ica.html.

[25] A. Hyvärinen, "The Fast-ICA MATLAB package," 1998. http://www.cis.hut.fi/˜aapo/.

[26] J.-F. Cardoso, "JADE for real-valued data," 1999. http://sig.enst.fr:80/˜cardoso/guidesepsou.html.