RELATIVE NEWTON METHOD FOR SIGNAL SEPARATION

Michael Zibulevsky

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

ABSTRACT

Presented relative Newton method for quasi-maximum likelihood blind source separation significantly outperforms natural gradient descent in batch mode. The structure of the corresponding Hessian matrix allows its fast inversion without assembling. Experiments with sparsely representable signals and images demonstrate super-efficient separation.

1. INTRODUCTION

Several Newton-like methods have been already proposed for blind source separation, see [1] for more details. Here we present a Newton method for quasi-maximum likelihood source separation [2, 3] in batch mode, without orthogonality constraint. This criterion provides improved separation quality 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

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

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 $h(\cdot)$ [4]. 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 a relative Newton method, which overcomes this obstacle, and provides fast and very accurate separation of sparse sources.

2. RELATIVE OPTIMIZATION (RO) ALGORITHM

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

1. **For** $k = 1, 2, \ldots$, until convergence
2. Compute current source estimate $U_k = W_k X$;
3. 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 $L(V; U_k)$;
4. Update the estimated separation matrix $W_{k+1} = V_{k+1} W_k$;
5. **End**

The relative (natural) gradient method [5, 6] is a particular instance of this approach, when the 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 algorithm progress 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

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

starting point. Convergence analysis of the RO-algorithm is presented in [1]. In the following we will use one 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 $g(w) = \nabla\hat{L}(w;X) = \mathrm{vec}(G(W))$ Then $dg = Hdw$, where H is $N^2 \times N^2$ Hessian matrix.

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

The differential of the first term in (5)

$$dG = d(W^{-T}) = -A^T(dW^T)A^T, \quad (8)$$

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

$$dG_{ij} = -A_i(dW^T)A^j = -\mathrm{Trace}A^jA_i(dW^T), \quad (9)$$

where A_i and A^j are i-th row and j-th column of A respectively. Therefore the k-th row of H, where $k = (i-1)N+j$, contains the matrix A^jA_i stacked column-wise

$$H_k = \mathrm{vec}^T(A^jA_i)^T \quad (10)$$

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

is a block-diagonal matrix with the following $N \times N$ blocks

$$B^m = \frac{1}{T}\sum_t h''(W_m x(t))x(t)x^T(t), \quad m = 1,..,N \quad (11)$$

4. NEWTON METHOD

Newton method 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 the standard Newton approach, in which the direction is given by solution of the linear equation

$$Hy = -\nabla\hat{L}(w;X) \quad (12)$$

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] [7], which automatically finds such a diagonal matrix R, that the matrix $H+R$ is positive definite, and provides a solution to the modified system

$$(H+R)y = -\nabla\hat{L}(w;X) \quad (13)$$

After the direction y is found, the new iterate w^+ is given by $w^+ = w + \alpha y$ where the step size α is determined by exact or backtracking line search, which guarantees monotonic decrease of the objective function at every iteration.

Complexity of the Newton step The Hessian is a $N^2 \times N^2$ matrix; its computation requires N^4 operations in (10) and N^3T operations in (11). Solution of the Newton system (13) 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 introduce 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 (10), becomes very simple

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

and sparse, when $W = A = I$: each row of H

$$H_k = \mathrm{vec}^T(e_i e_j^T), \tag{14}$$

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 $U_k = S$. Off-diagonal elements of m-th block of the second term of $\nabla^2 L(I; S)$ given by (11), are equal to

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

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

$$B_{ii}^m = \frac{1}{T}\sum_t h''(u_m(t))u_i^2(t), i = 1,..,N; \;\; m = 1,..,N, \tag{15}$$

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 (15) into $N \times N$ matrix D, row-by-row. Taking into account that $A = I$ in (8), we will obtain the following expression for the differential of the gradient

$$dG = \mathcal{H}dW = dW^T + D \odot dW, \tag{16}$$

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

$$\mathcal{H}Y = Y^T + D \odot Y. \tag{17}$$

In order to solve the Newton system

$$Y^T + D \odot Y = G \tag{18}$$

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

$$\begin{aligned} D_{ij}Y_{ij} + Y_{ji} &= G_{ij}, \;\; i = 1,..N;\; j = 1,..,i-1 \\ D_{ji}Y_{ji} + Y_{ij} &= G_{ji} \end{aligned} \tag{19}$$

The diagonal elements Y_{ii} can be found directly from the set of single equations $D_{ii}Y_{ii} + Y_{ii} = G_{ii}$. In order to guarantee descent direction and avoid saddle points, we modify the Newton system (19), changing the sign of the negative eigenvalues [7]. 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.

Complexity of the fast Newton step. Computing the diagonal of the Hessian by (15) requires N^2T operations, which is equal to the cost of the gradient computation. Solution cost of the set of 2x2 linear equations (19) is about $15N^2$ operations, which is negligible compared to the gradient cost.

5.3. Sequential Optimization

Optimization of the likelihood function becomes more and more difficult with the decrease of the smoothing parameter λ. Therefore, we use sequential optimization with gradual reduction of λ, see [1] for details.

6. COMPUTATIONAL EXPERIMENTS

The sources were represented by 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 $\sigma = 1$.

In all experiments we used backtracking line search. 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

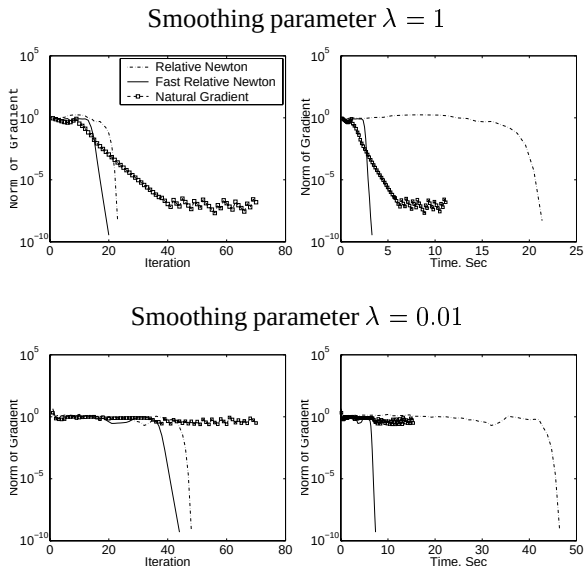

Fig. 1. Progress with iteration/time of different algorithms. Relative Newton with exact Hessian – dashed line, fast relative Newton – continuous line, natural gradient in batch mode – squares.

Newton method with stochastic natural gradient [5, 6], Fast ICA [8] and JADE [9]. 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 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 $\lambda = 10^{-6}$. More experiments with natural images are presented in [1].

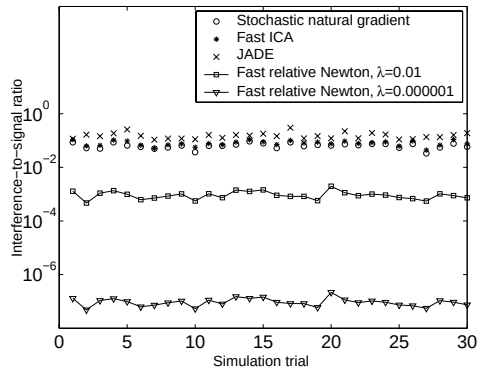

Fig. 2. Separation quality: 30 simulation trials.

7. CONCLUSIONS

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

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.

8. REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

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

[8] 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.

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