Trust-Region ICA 알고리듬

최희열[O] 김숙정 최승진
포항공과대학교 컴퓨터공학과
{hychoi[O], koko, seungjin}@postech.ac.kr

A Trust-Region ICA algorithm

Heeyoul Choi[O] Sookjeong Kim Seungjin Choi
Department of Computer Science, POSTECH, Korea

Abstract

A trust-region method is a quite attractive optimization technique. It is, in general, faster than the steepest descent method and is free of a learning rate unlike the gradient-based methods. In addition to its convergence property (between linear and quadratic convergence), ifs stability is always guaranteed, in contrast to the Newton's method. In this paper, we present an efficient implementation of the maximum likelihood independent component analysis (ICA) using the trust-region method, which leads to trust-region-based ICA (TR-ICA) algorithms. The useful behavior of our TR-ICA algorithms is confirmed through numerical experimental results.

1. Introduction

The simplest form of Independent Component Analysis (ICA) considers the noise-free linear generative model where the observation data $x(t)$ is assumed to be generated by

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

where $A \in \mathbf{R}^{n \times n}$ contains n basis vectors in its columns and $s(t)$ is a latent variable vector whose elements are mutually independent.

In general, the objective function of ICA has the form

$$R = KL[p^0(x) \,\mathrm{P}\, p(x)] = \int p^0(x) \log \frac{p^0(x)}{p(x)} dx, \tag{2}$$

where $p^0(x)$ and $p(x)$ are the observed density and model density, respectively [2].

Popular ICA algorithms were derived from the minimization of the loss function Eq. (2) using the natural gradient where the update rule is given by

$$W^{(k+1)} = W^{(k)} + \eta\{I - E\{\varphi(y)y^T\}\}W^{(k)}. \tag{3}$$

Although gradient-based algorithms are simple and guarantee the global convergence, but they are relatively slow and require a careful choice of a learning rate, which are cumbersome in practical applications. To overcome these drawbacks, Newton-type algorithms were recently proposed [1],[5].

A trust-region method is a quite attractive optimization technique, which finds a direction and a step size in an efficient and reliable manner with the help of a quadratic model of the objective function [3]. Its convergence is between linear and quadratic rate and its stability is always guaranteed, in contrast to the Newton's method.

In this paper, we present trust-region-based ICA (TR-ICA) algorithms in the framework of maximum likelihood ICA so that our algorithms carry the useful properties that trust-region methods have.

2. Trust-Region Method

Trust-region methods define a region around the current iterate within which they trust the model to be an adequate representation of the objective function, and then choose the step to be the approximate minimizer of the model in this trust region. In effect, they choose the direction and length of the step simultaneously. If a step is not acceptable, they reduce the size of the region and find a new minimizer. In general, the step direction changes whenever the size of the trust region is altered.

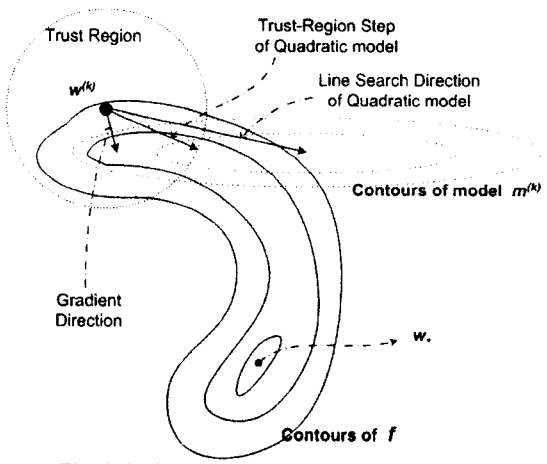

Fig. 1. An illustration of the trust-region method

Let us consider an objective function $f(w): \mathbf{R}^{n^2} \rightarrow \mathbf{R}$ to be minimized with respect to the parameter $w \in \mathbf{R}^{n^2}$. Fig. 1 illustrates a trust-region approach for the minimization of an objective function $f(w)$ in which the current point $w^{(k)}$ lies at one end of a curved valley while the minimizer w_* lies at the other end. A quadratic model function $m^{(k)}$ which has elliptical contours, is based on function and derivative information at $w^{(k)}$. The step $p \in \mathbf{R}^{n^2}$ is obtained by solving the following subproblem:

$$\min_{\|p\| \le \Delta^{(k)}} m^{(k)}(p) = f^{(k)} + [\nabla f^{(k)}]^T p + \frac{1}{2} p^T B^{(k)} p, \qquad (4)$$

where $\Delta^{(k)} > 0$ is the trust-region radius and $\|\cdot\|$ is the Euclidean norm. Here, $B^{(k)} \in \mathbf{R}^{n^2 \times n^2}$ is some symmetric matrix and

$$f^{(k)} = f(w^{(k)}), \quad \nabla f^{(k)} = \frac{\partial f}{\partial w}\Big|_{w=w^{(k)}}. \qquad (5)$$

The solution of Eq. (4) is the minimizer of $m^{(k)}$ in the ball of radius $\Delta^{(k)}$.

The first issue in defining a trust-region method is the strategy for choosing the trust-region radius $\Delta^{(k)}$ at each iteration. Our choice of $\Delta^{(k)}$ is based on the agreement between the model function $m^{(k)}$ and the objective function $f(w)$ at previous iterations. Given a step $p^{(k)}$, this agreement measure $\rho^{(k)}$ is defined as the ratio of *actual reduction* to *predicted reduction*, i.e.,

$$\rho^{(k)} = \frac{f^{(k)} - f(w^{(k)} + p^{(k)})}{m^{(k)}(0) - m^{(k)}(p^{(k)})}. \qquad (6)$$

3. TR-ICA

In general, trust-region methods require the Hessian matrix of the objective function and the evaluation of the objective value at the current parameter estimate. To this end, we consider the quasi maximum likelihood ICA [4] and describe our TR-ICA algorithms for exemplary objective functions for super- and sub-Gaussian sources so that the objective values can be easily evaluated. This can be easily generalized to any other objective functions in ICA. For the Hessian matrix calculation, we use the result in [5].

Here, we need to specify the probability density functions $p_i(\cdot)$. Hence, we consider two cases, each of which corresponds to the super- or sub-Gaussian source in Eq. (7).

$$\frac{\nabla p_i^+(y_i)}{p_i^+(y_i)} = -\tanh(y), \quad \frac{\nabla p_i^-(y_i)}{p_i^-(y_i)} = -y^3. \qquad (7)$$

In ICA, the objective function (corresponding the risk in the quasi maximum likelihood ICA) is written as

$$R = -\log|W| - E\{\sum_{i=1}^{n} \log p_i(y_i)\}, \qquad (8)$$

where the statistical average is replaced by the time average over N data points. Then the exact functions are given by

$$f^+(w) = -\log|W| + \frac{1}{\alpha} E\{\sum_i \log\cosh(\alpha y_i)\}, \qquad (9)$$

$$f^-(w) = -\log|W| + \beta E\{\sum_i (y_i)^4\}. \qquad (10)$$

We define an n-dimensional element-wise function $\psi(y) \in \mathbf{R}^n$ by its i-th element, $\psi_i(y_i) = -\log p_i(y_i)$. We denote the element-wise 1st-order derivative and 2nd-order derivative of ψ by ψ' and ψ'', respectively.

Regardless of super- or sub-Gaussian, the gradient and the Hessian matrix of the objective function are given by

$$f(w) = -\log|W| + \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{n} \psi_i(y_i(t)), \qquad (11)$$

$$\nabla f(w) = vec(-W^{-1} + \frac{1}{N}\sum_{t=1}^{N} x(t)(\psi'(y(t)))^T), \qquad (12)$$

$$\nabla^2 f(w) = H + D, \qquad (13)$$

where $vec(\cdot)$ is the *vec-function* which stacks the columns of the given matrix into one long vector. And $D \in \mathbf{R}^{n^2 \times n^2}$ is a block-diagonal matrix which consists of n blocks, $D_l \in \mathbf{R}^{n \times n}$, which have the form

$$D_l = \frac{1}{N}\sum_{t=1}^{N} \psi_l''(y_l(t)) x(t) x^T(t), \qquad (14)$$

and $H \in \mathbf{R}^{n^2 \times n^2}$ consists of n^2 row vectors, $\vec{h}_m$, that is given by

$$\vec{h}_m = [vec(a_j \vec{a}_i)]^T, \; m = (i-1)n + j, \qquad (15)$$

for $i=1,\cdots,n$ and $j=1,\cdots,n$. a_j and $\vec{a}_i$ denote the i-th column vector and the i-th row vector of $A = W^{-1}$. The TR-ICA algorithm with the dogleg method is summarized in Table 1.

4. Experiments

We used 2 different data sets for our experiments. Data1 is a set of binary data which consists of mixtures of three binary sources with 10000 data points for each source. Data2 consists of mixtures of two speeches and one music signal, all of them were sampled at 8 kHz. Three mixture signals were generated using the mixing matrix A given by

$$A = \begin{pmatrix} -0.4667 & 2.0636 & -0.5136 \\ 0.0680 & 2.3982 & -0.1961 \\ -2.5108 & 0.3002 & 0.2247 \end{pmatrix}, \qquad (16)$$

where the condition number of A is 11.12 (well-conditioned mixing).

In the gradient and the Newton method, the backtracking algorithm was used to select the optimal learning rate. Therefore the gradient (or the natural gradient) ICA algorithm achieves the convergence faster than the case where the constant or annealing learning rate was used.

Fig. 2 shows the convergence comparison of several numerical optimization methods in ICA, which include: (1)

gradient; (2) natural gradient; (3) trust-region (fminunc); (4) trust-region (dogleg); (5) Newton. As expected, the gradient method required more iterations for convergence, compared to the trust-region or Newton method. In this case, the Newton method required less number of iterations for convergence, but ate up the almost same amount of CPU time as trust-region methods, due to the high time complexity of the Newton method.

In addition to the convergence comparison, we also carried out the performance comparison in terms of the performance index (PI) that is defined as

$$PI = \frac{1}{n(n-1)} \sum_{i=1}^{n} \frac{\sum_{j=1}^{n} |g_{ij}|}{\max_j |g_{ij}|} - 1, \qquad (17)$$

where g_{ij} is the (i,j)-element of the global system matrix $G = WA$. Table. 2 summarizes the PI of the algorithms that we tested. There was no difference in terms of PI for several different optimization methods, which means, the final performance after the convergence was achieved, were similar.

Table 1. **Dogleg TR-ICA algorithm**

Given $\Delta > 0$, $\Delta^{(0)} \in (0, \Delta)$, and $\xi \in [0, 0.25)$:
for $k = 0, 1, 2, \ldots$
  if $\nabla^2 f^{(k)}$ is positive definite, then
    if $\| (\nabla^2 f^{(k)})^{-1} \nabla f^{(k)} \| \leq \Delta$, then
      $p^{(k)} = \| (\nabla^2 f^{(k)})^{-1} \nabla f^{(k)} \|$
    else, then
      $p^{(k)}$ =intersect(Dogleg path, TR boundary)
  else, then
    $$p^{(k)} = -\frac{\nabla f^T \nabla f}{\nabla f^T B \nabla f} \nabla f$$
  Evaluate $\rho^{(k)}$ from Eq. (6)

  if $\rho^{(k)} < 0.25$, then $\Delta^{(k+1)} = 0.25 \| p^{(k)} \|$
  else, then
    if $\rho^{(k)} > 0.75$ and $\| p^{(k)} \| = \Delta^{(k)}$, then
      $\Delta^{(k+1)} = \min(2\Delta^{(k)}, \Delta)$
    else, then $\Delta^{(k+1)} = \Delta^{(k)}$

  if $\rho^{(k)} > \xi$, then $w^{(k+1)} + p^{(k)}$
  else, then $w^{(k+1)} = w^{(k)}$
end (for)

5. Discussion

We have presented TR-ICA algorithms which employed the trust-region optimization scheme with the dogleg and the subspace method.

TR-ICA algorithms took much less number of iterations for convergence, compared to the gradient or the natural gradient ICA algorithms and took almost same number of iterations as Newton-type ICA algorithms. The TR-ICA showed the best convergence performance in terms of both iteration numbers and CPU time.

6. Acknowledgment

This work was supported by Korea Ministry of Science and Technology under Brain Science and Engineering Research Program, KOSEF 2000-2-20500-009-5, and Brain Korea 21 in POSTECH.

7. References

[1] T. Akuzawa, "Extended quasi-Newton method for the ICA," in Proc. ICA, Helsinki, Finland, 2000, pp. 521-525.

[2] S. Choi and A. Cichocki, "Correlation matching approach to source sepration in the presence of spatially correlated noise," in Proc. IEEE ISSPA, Kuala-Lumpur, Malaysia, 2001.

[3] J. Nocedal and S. J. Wright, Numerical Optimization. Springer, 1999.

[4] D. T. Pham and P. Garrat, "Blind separation of mixtures of independent sources a quasi maximum likelihood approach," IEEE Trans. Signal Processing, vol. 45, no. 7, pp. 1712-1725, 1997.

[5] M. Zibulevsky, "Blind source separation with relative Newton method," in Proc. ICA, Nara, Japan, 2003, pp. 897-902.

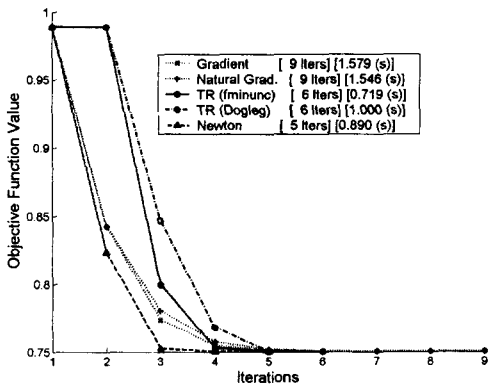

Fig. 2. Convergence comparison of several optimization methods for a set of binary data.

Table. 2. Performance Index

Methods	Binary Data	Sound Signal
Gradient	2.12085e-003	6.17350e-003
Natural Grad.	2.11971e-003	7.29381e-003
TR fminunc	2.12907e-003	7.90488e-003
TR dogleg	2.12096e-003	7.90666e-003
Newton	2.12096e-003	7.90652e-003