Trust-Region Learning for ICA

Heeyoul Choi, Sookjeong Kim, Seungjin Choi
Department of Computer Science
POSTECH
San 31 Hyoja-dong, Nam-gu
Pohang 790-784, Korea
Email: {hychoi, koko, seungjin}@postech.ac.kr

***Abstract*— 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. It is, in general, faster than the steepest descent method and is free of a pre-selected constant learning rate. In addition to its convergence property (between linear and quadratic convergence), its 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.**

I. Introduction

Independent component analysis (ICA) is a statistical method that decomposes a multivariate data into a linear sum of non-orthogonal basis vectors with basis coefficients being statistically independent. A variety of approaches to ICA have been developed. These include maximum likelihood estimation, mutual information minimization, output entropy maximization (infomax), and negentropy maximization (see [5], [7] and references therein). All these approaches lead to an identical objective function in ICA. A popular implementation in these approaches, is gradient-descent learning (including the natural gradient). Although gradient-based algorithms are simple and guarantee the local stability, but they are relatively slow and require a careful choice of a learning rate, which are cumbersome in practical applications. In order to overcome these drawbacks, Newton-type algorithms were recently proposed [1], [12].

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 [10]. It defines 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. The step direction changes whenever the size of the trust region is altered. It is, in general, faster than the steepest descent method and is free of a constant learning rate unlike the conventional gradient-based methods. Instead, the trust-region takes the place of learning rate. 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. As practical implementation, we consider the dogleg method, two-dimensional subspace method, and the Steihaug method which are briefly reviewed in Sec. III.

II. Independent Component Analysis

The simplest form of ICA considers the noise-free linear generative model where the observation data $\boldsymbol{x}(t) \in \mathbb{R}^n$ is assumed to be generated by

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t), \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ contains n basis vectors $\boldsymbol{a}_i \in \mathbb{R}^n$, $i = 1, \ldots, n$ in its columns and $\boldsymbol{s}(t) \in \mathbb{R}^n$ is a latent variable vector whose elements $s_i(t)$ are mutually independent.

In general, ICA can be illustrated by a probability density matching problem [2], [4] which, in fact, turned out to be equivalent to infomax, mutual information minimization, and maximum likelihood estimation [3].

Let us denote the observed density and model density by $p^o(\boldsymbol{x})$ and $p(\boldsymbol{x})$, respectively. The probability density matching finds the parameters, $\boldsymbol{A}$, which best match the observed density $p^o(\boldsymbol{x})$ and the model density $p(\boldsymbol{x})$. When the Kullback-Leibler divergence is used as a distance measure, the probability density matching is also referred to as the Kullback matching, which leads to the risk that has the form

$$\begin{aligned} \mathcal{R} &= KL\left[p^o(\boldsymbol{x}) || p(\boldsymbol{x})\right] \\ &= \int p^o(\boldsymbol{x}) \log \frac{p^o(\boldsymbol{x})}{p(\boldsymbol{x})} d\boldsymbol{x}. \end{aligned} \tag{2}$$

Note that the model density $p(\boldsymbol{x})$ satisfies the following relation:

$$\log p(\boldsymbol{x}) = -\log |\det \boldsymbol{A}| + \sum_{i=1}^{n} \log p_i(s_i). \tag{3}$$

Define $\boldsymbol{W} = \boldsymbol{A}^{-1}$, then the estimates of latent variables are $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$. With these definitions, the risk can be rewritten as

$$\mathcal{R} = -\log |\det \boldsymbol{W}| - E\left\{ \sum_{i=1}^{n} \log p_i(y_i) \right\}, \tag{4}$$

where $E\{\cdot\}$ denotes the statistical expectation operator. The natural gradient ICA algorithm updates $\boldsymbol{W}$ by

$$\boldsymbol{W}^{(k+1)} = \boldsymbol{W}^{(k)} + \eta \left\{ \boldsymbol{I} - E\left\{ \varphi(\boldsymbol{y}) \boldsymbol{y}^T \right\} \right\} \boldsymbol{W}^{(k)}, \quad (5)$$

where $\eta > 0$ is the learning rate and $\varphi(\boldsymbol{y})$ is the n-dimensional element-wise function whose ith element $\varphi_i(y_i)$ is the negative score function, i.e., $\varphi_i(y_i) = -\frac{d \log p_i(y_i)}{dy_i}$.

III. Trust-Region Methods

In this section, we briefly review a basic idea and practical implementation of trust-region methods. Refer to [10] for further details.

A. Basic Idea

Trust-region methods [10] 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.

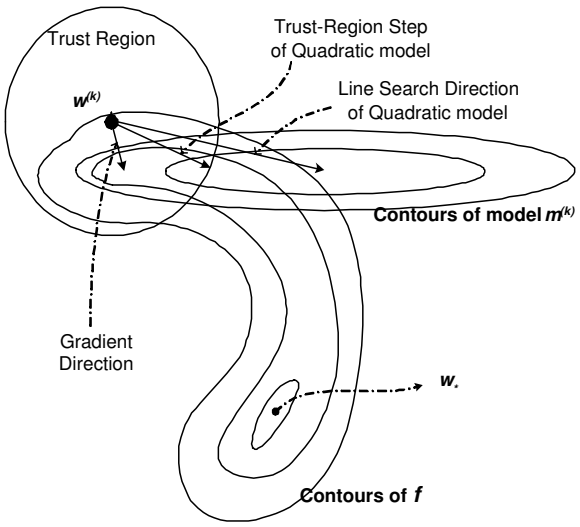

Fig. 1. An illustration of the trust-region method in determining a direction and a step size with the help of a quadratic model of the objective function.

Let us consider an objective function $f(\boldsymbol{w}) : \mathbb{R}^{n^2} \to \mathbb{R}$ to be minimized with respect to the parameter $\boldsymbol{w} \in \mathbb{R}^{n^2}$. Fig. 1 illustrates a trust-region approach for the minimization of an objective function f in which the current point $\boldsymbol{w}^{(k)}$ lies at one end of a curved valley while the minimizer $\boldsymbol{w}_*$ lies at the other end. A quadratic model function $m^{(k)}$ which has elliptical contours, is based on function and derivative information at $\boldsymbol{w}^{(k)}$ and possibly also on information accumulated from previous iterations and steps:

$$m^{(k)}(\boldsymbol{p}) = f^{(k)} + \left[\nabla \boldsymbol{f}^{(k)} \right]^T \boldsymbol{p} + \frac{1}{2} \boldsymbol{p}^T \boldsymbol{B}^{(k)} \boldsymbol{p}, \quad (6)$$

where $\boldsymbol{p} \in \mathbb{R}^{n^2}$ represents the step and $\boldsymbol{B}^{(k)} \in \mathbb{R}^{n^2 \times n^2}$ is some symmetric matrix and

$$\begin{aligned} f^{(k)} &= f(\boldsymbol{w}^{(k)}), \\ \nabla \boldsymbol{f}^{(k)} &= \left. \frac{\partial f}{\partial \boldsymbol{w}} \right|_{\boldsymbol{w}=\boldsymbol{w}^{(k)}}. \end{aligned} \quad (7)$$

A line search method based on this model searches along the step to the minimizer of model $m^{(k)}$, but this direction allows only a small reduction in f even if an optimal step is taken. A gradient direction does not use the information of $\boldsymbol{B}^{(k)}$, the rapid convergence can be expected only if $\boldsymbol{B}^{(k)}$ plays a role in determining the direction of the step as well as its length.

A trust-region method, on the other hand, steps to the minimizer of $m^{(k)}$ within the trust-region circle, which yields a more significant reduction in f and a better step. The step $\boldsymbol{p}$ is obtained by solving the following subproblem:

$$\min_{\|\boldsymbol{p}\| \le \triangle^{(k)}} m^{(k)}(\boldsymbol{p}) = f^{(k)} + \left[\nabla \boldsymbol{f}^{(k)} \right]^T \boldsymbol{p} + \frac{1}{2} \boldsymbol{p}^T \boldsymbol{B}^{(k)} \boldsymbol{p}, \quad (8)$$

where $\triangle^{(k)} > 0$ is the trust-region radius and $\| \cdot \|$ is the Euclidean norm. The solution $\boldsymbol{p}_*^{(k)}$ of Eq. (8) is the minimizer of $m^{(k)}$ in the ball of radius $\triangle^{(k)}$.

B. Algorithm

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

$$\rho^{(k)} = \frac{f\left(\boldsymbol{w}^{(k)}\right) - f\left(\boldsymbol{w}^{(k)} + \boldsymbol{p}^{(k)}\right)}{m^{(k)}(\boldsymbol{0}) - m^{(k)}\left(\boldsymbol{p}^{(k)}\right)}. \quad (9)$$

Note that the predicted reduction, $m^{(k)}(\boldsymbol{0}) - m^{(k)}\left(\boldsymbol{p}^{(k)}\right)$, is always nonnegative since the step $\boldsymbol{p}^{(k)}$ is obtained by minimizing the model $m^{(k)}$ over a region that includes the step $\boldsymbol{p} = \boldsymbol{0}$. Thus if $\rho^{(k)}$ is negative, the new objective value $f\left(\boldsymbol{w}^{(k)} + \boldsymbol{p}^{(k)}\right)$ is greater than the current value $f\left(\boldsymbol{w}^{(k)}\right)$, so the step must be rejected. On the other hand, if $\rho^{(k)}$ is close to 1, there is good agreement between the model $m^{(k)}$ and the function f over this step, so it is safe to expand the trust region for the next iteration. If $\rho^{(k)}$ is positive but not close to 1, we do not alter the trust region, but if it is close to zero or negative, we shrink the trust region.

In general, trust-region methods are faster than gradient methods and guarantee the stability regardless of initial conditions whereas Newton's method does not. In a practical consideration, a solution to Eq. (8) is very important and there are some approximate solutions such as the dogleg method, the two-dimensional subspace minimization, and the Steihaug method. In this paper we use the dogleg method and the

subspace method which is implemented through the *fminunc* function in Matlab Toolbox.

(Trust-Region Algorithm)
Given $\triangle > 0$, $\triangle^{(0)} \in (0, \triangle)$, and $\zeta \in [0, \frac{1}{4})$:
for $k = 0, 1, 2, \ldots$
  Find $\boldsymbol{p}^{(k)}$ which (approximately) solves Eq. (8);
  Evaluate $\rho^{(k)}$ from Eq. (9);
  if $\rho^{(k)} < \frac{1}{4}$, then $\triangle^{(k+1)} = \frac{1}{4}\|\boldsymbol{p}^{(k)}\|$
  else, then
    if $\rho^{(k)} > \frac{3}{4}$ and $\|\boldsymbol{p}^{(k)}\| = \triangle^{(k)}$, then
      $\triangle^{(k+1)} = \min(2\triangle^{(k)}, \triangle)$
    else, then $\triangle^{(k+1)} = \triangle^{(k)}$;

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

C. Dogleg, Subspace, and Steihaug

In Eq. (6), when $\boldsymbol{B}$ is positive definite, the unconstrained minimizer of m is the full step $\boldsymbol{p}_B = -\boldsymbol{B}^{-1}\nabla \boldsymbol{f}$. When this point is feasible for Eq. (8), we have $\boldsymbol{p}_*^{(k)} = \boldsymbol{p}_B$ for $\triangle \geq \|\boldsymbol{p}_B\|$. When $\triangle$ is tiny, the restriction $\|\boldsymbol{p}\| \leq \triangle$ ensures that the quadratic term in m has little effect on the solution of Eq. (8). The true solution $\boldsymbol{p}$ is approximately the same as the solution we would obtain by minimizing the linear function $f + \nabla \boldsymbol{f}^T \boldsymbol{p}$ over $\|\boldsymbol{p}\| \leq \triangle$, that is, $\boldsymbol{p} \approx -\triangle \frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}^T\|}$, when $\triangle$ is small.

For intermediate values of $\triangle$, the solution $\boldsymbol{p}_*$ typically follows a curved trajectory like the one in Fig. 2. The dogleg method finds an approximate solution by replacing the curved trajectory for $\boldsymbol{p}_*$ with a path consisting of two line segments. The first line segment runs from the origin to the unconstrained minimizer along the steepest descent direction defined by $\boldsymbol{p}_U = -\frac{\nabla \boldsymbol{f}^T \nabla \boldsymbol{f}}{\nabla \boldsymbol{f}^T \boldsymbol{B} \nabla \boldsymbol{f}} \nabla \boldsymbol{f}$, while the second line segment runs from $\boldsymbol{p}_U$ to $\boldsymbol{p}_B$ (see Fig. 2).

The dogleg algorithm is an effective method when $\boldsymbol{B}$ is positive definite. If $\boldsymbol{B}$ is not positive definite, its information is discarded so that only steepest descent direction is exploited. When $\boldsymbol{B}$ is positive definite and the full step is in the trust-region, then the $\boldsymbol{p}$ becomes the full step. Otherwise the step $\boldsymbol{p}$ is at the point of intersection of the dogleg path and the trust-region boundary.

Compared to the dogleg method, the subspace method widens the search for $\boldsymbol{p}$ to the entire two-dimensional subspace spanned by $\boldsymbol{p}_B$ and $\boldsymbol{p}_U$, when $\boldsymbol{B}$ is positive definite. For positive definite $\boldsymbol{B}$, the subspace method considers the

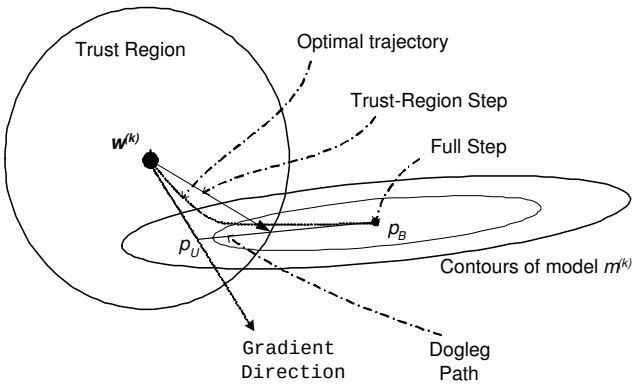

Fig. 2. The optimal trajectory and the dogleg approximation.

following subproblem:

$$\min_{\|\boldsymbol{p}\| \leq \triangle^{(k)}} m^{(k)}(\boldsymbol{p}) = f^{(k)} + \left[\nabla \boldsymbol{f}^{(k)}\right]^T \boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T \boldsymbol{B}^{(k)} \boldsymbol{p}, \quad \text{s.t. } \boldsymbol{p} \in \text{span}\left[\nabla \boldsymbol{f}^{(k)}, \left(\boldsymbol{B}^{(k)}\right)^{-1} \nabla \boldsymbol{f}^{(k)}\right]. \quad (10)$$

When $\boldsymbol{B}$ contains negative eigenvalues, the two-dimensional subspace in Eq. (10) is changed to

$$\text{span}\left[\nabla \boldsymbol{f}^{(k)}, \left(\boldsymbol{B}^{(k)} + \xi \boldsymbol{I}\right)^{-1} \nabla \boldsymbol{f}^{(k)}\right], \quad (11)$$

for some $\xi \in (-\lambda_1, -2\lambda_1]$ where λ_1 is the most negative eigenvalue of $\boldsymbol{B}^{(k)}$.

The Steihaug method is based on the conjugate gradient algorithm, an iterative algorithm for solving linear systems with symmetric positive definite coefficient matrices. Hence it is expected to converge to a solution faster, especially for high-dimensional data.

D. Local Stability Analysis

Trust-region methods guarantee the local stability, which is stated in the following theorem (See [10] for the proof).

Theorem 1: Suppose that $\|\boldsymbol{B}^{(k)}\| \leq \beta$ for some constant β, that f is bounded below on the level set $\{\boldsymbol{w} | f(\boldsymbol{w}) \leq f(\boldsymbol{w}^{(0)})\}$, and that all approximate solutions of Eq. (8) satisfy the inequalities

$$m^{(k)}(\boldsymbol{0}) - m^{(k)}\left(\boldsymbol{p}^{(k)}\right) \geq c_1 \|\nabla \boldsymbol{f}^{(k)}\| \min\left(\triangle^{(k)}, \frac{\left\|\nabla \boldsymbol{f}^{(k)}\right\|}{\left\|\boldsymbol{B}^{(k)}\right\|}\right),$$

where $0 < c_1 \leq 1$ and $\|\boldsymbol{p}^{(k)}\| \leq \gamma \triangle^{(k)}$ for some $\gamma \geq 1$. If $\zeta \in (0, \frac{1}{4})$ in the trust-region algorithm and f is Lipschitz continuously differentiable, then we have

$$\lim_{k \to \infty} \nabla \boldsymbol{f}^{(k)} = \boldsymbol{0}.$$

Remarks: The Cauchy point $\boldsymbol{p}_c^{(k)}$ is a point that minimizes $m^{(k)}$ along the steepest descent direction. It can be shown that the Cauchy point $\boldsymbol{p}_c^{(k)}$ satisfies above inequality with $c_1 = \frac{1}{2}$. This implies that the dogleg, two-dimensional subspace minimization and Steihaug method satisfy above inequality

with $c_1 = \frac{1}{2}$, because they all produce approximate solutions $\boldsymbol{p}^{(k)}$ for which $m^{(k)}\left(\boldsymbol{p}^{(k)}\right) \leq m^{(k)}\left(\boldsymbol{p}_c^{(k)}\right)$.

IV. TR-ICA

Popular ICA algorithms are based on the gradient or the natural gradient method. Recently Newton or quasi-Newton method were applied to ICA [1], [12]. Trust region methods carry some useful properties such as super-linear convergence (between linear and quadratic convergence), local stability, and adjustable learning rate. To our best knowledge, trust-region methods have never been employed in ICA, yet. In this section, we develop TR-ICA algorithms using the dogleg method and the subspace method.

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 [11] 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 [12].

The quasi maximum likelihood ICA leads to the following empirical risk:

$$\mathcal{R} = -\log|\det \boldsymbol{W}| - \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{n}\log p_i(y_i(t)). \tag{12}$$

Trust-region methods update the size of the trust-region, depending on the objective value evaluated at the current estimate. Hence, we need to specify the probability density functions $p_i(\cdot)$. Here we consider two cases, each of which corresponds to the super- and sub-Gaussian sources.

Let us denote the super- and sub-Gaussian density function by p_i^+ and p_i^-, respectively. In the description of our algorithms, the parameter vector is $\boldsymbol{w} \in \mathbb{R}^{n^2} = \text{vec}\left(\boldsymbol{W}^T\right)$ where $\text{vec}(\cdot)$ is the *vec-function* which stacks the columns of the given matrix into one long vector.

We consider two exemplary score functions that were used in ICA

$$\begin{aligned}\frac{d\log p_i^+(y_i)}{dy_i} &= -\tanh(y_i),\\ \frac{d\log p_i^-(y_i)}{dy_i} &= -y_i^3,\end{aligned}$$

which lead to the objective functions, $f^+(\boldsymbol{w})$ and $f^-(\boldsymbol{w})$, that have the form

$$\begin{aligned}f^+(\boldsymbol{w}) &= -\log|\boldsymbol{W}| - \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{n}\log p_i^+(y_i(t))\\ &= -\log|\boldsymbol{W}| + \frac{1}{\alpha N}\sum_{t=1}^{N}\sum_{i=1}^{n}\log\cosh(\alpha y_i(t)),\\ f^-(\boldsymbol{w}) &= -\log|\boldsymbol{W}| - \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{n}\log p_i^-(y_i(t))\\ &= -\log|\boldsymbol{W}| + \frac{1}{\beta N}\sum_{t=1}^{N}\sum_{i=1}^{n}y_i^4(t),\end{aligned}$$

where α, β are positive normalizing constants that are chosen such that p_i^+ and p_i^- are eligible density functions.

A. Gradient Descent Learning: Backtracking

In contrast to using a constant learning rate in the gradient descent method, the backtracking method exploits the variable step size, which is summarized below.

(Backtracking line search)
Choose $\eta^{(0)}, \rho, c \in (0,1)$; set $\eta \leftarrow \eta^{(0)}$;
repeat
  until $f\left(\boldsymbol{w}^{(k)} + \alpha\boldsymbol{p}^{(k)}\right) \leq f^{(k)} + c\eta\left[\nabla\boldsymbol{f}^{(k)}\right]^T\boldsymbol{p}^{(k)}$
  $\eta \leftarrow \rho\eta$;
end (repeat)
Terminate with $\eta^{(k)} = \eta$

In our numerical experiments, we used $\eta^{(0)} = 1$, $\rho = 0.3$, $c = 0.3$.

B. Trust-Region Learning

We define the n-dimensional element-wise function $\psi(\boldsymbol{y}) \in \mathbb{R}^n$ by its ith 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. Then the objective function (corresponding the risk in the quasi maximum likelihood ICA) is written as

$$f(\boldsymbol{w}) = -\log|\det \boldsymbol{W}| + \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{n}\psi_i(y_i(t)), \tag{13}$$

where the statistical average is replaced by the time average over N data points.

Regardless of super- or sub-Gaussian, the gradient and the Hessian matrix of the objective function Eq. (13) are given by

$$\nabla\boldsymbol{f}(\boldsymbol{w}) = \text{vec}\left(-\boldsymbol{W}^{-1} + \frac{1}{N}\sum_{t=1}^{N}\boldsymbol{x}(t)(\psi'(\boldsymbol{y}(t)))^T\right) \tag{14}$$

$$\nabla^2\boldsymbol{f}(\boldsymbol{w}) = \boldsymbol{H} + \boldsymbol{D}, \tag{15}$$

where $\boldsymbol{D} \in \mathbb{R}^{n^2\times n^2}$ is a block-diagonal matrix which consists of n blocks, $\boldsymbol{D}_l \in \mathbb{R}^{n\times n}$, which have the form

$$\boldsymbol{D}_l = \frac{1}{N}\sum_{t=1}^{N}\psi_l''(y_l(t))\boldsymbol{x}(t)\boldsymbol{x}^T(t), \tag{16}$$

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

$$\vec{\boldsymbol{h}}_m \quad = \quad [\mathrm{vec}\,(\boldsymbol{a}_j \vec{\boldsymbol{a}}_i)]^T, \quad m = (i-1)n + j, \qquad (17)$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, n$. $\boldsymbol{a}_j$ and $\vec{\boldsymbol{a}}_i$ denote the jth column vector and the ith row vector of $\boldsymbol{A} = \boldsymbol{W}^{-1}$. More detailed results can be found in [12].

The TR-ICA algorithm with the dogleg method is summarized below. In the dogleg method, the symmetric matrix $\boldsymbol{B}^{(k)}$ in Eq. (6) is replaced by the Hessian matrix $\nabla^2 \boldsymbol{f}\left(\boldsymbol{w}^{(k)}\right)$ in Eq. (15).

(Dogleg TR-ICA Algorithm)

Given $\triangle > 0$, $\triangle^{(0)} \in (0, \triangle)$, and $\zeta \in [0, \frac{1}{4})$:

for $k = 0, 1, 2, \ldots$

  if $\nabla^2 \boldsymbol{f}(\boldsymbol{w}^{(k)})$ is positive definite, then

    if $\|\nabla^2 \boldsymbol{f}(\boldsymbol{w}^{(k)})^{-1} \nabla \boldsymbol{f}\| \leq \triangle$, then

      $\boldsymbol{p}^{(k)} = \nabla^2 \boldsymbol{f}(\boldsymbol{w}^{(k)})^{-1} \nabla \boldsymbol{f}$

    else, then

      $\boldsymbol{p}^{(k)}$ = intersec(Dogleg path, TR boundary)

  else, then

    $\boldsymbol{p}^{(k)} = -\frac{\nabla \boldsymbol{f}^T \nabla \boldsymbol{f}}{\nabla \boldsymbol{f}^T \boldsymbol{B} \nabla \boldsymbol{f}} \nabla \boldsymbol{f}$

  Evaluate $\rho^{(k)}$ from Eq. (9);

  if $\rho^{(k)} < \frac{1}{4}$, then $\triangle^{(k+1)} = \frac{1}{4}\|\boldsymbol{p}^{(k)}\|$

  else, then

    if $\rho^{(k)} > \frac{3}{4}$ and $\|\boldsymbol{p}^{(k)}\| = \triangle^{(k)}$, then

      $\triangle^{(k+1)} = \min(2\triangle^{(k)}, \triangle)$

    else, then $\triangle^{(k+1)} = \triangle^{(k)}$;

  if $\rho^{(k)} > \zeta$, then $\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} + \boldsymbol{p}^{(k)}$

  else, then $\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)}$

end (for)

C. Newton Method

The basic Newton step $\boldsymbol{p}^{(k)}$ is obtained by solving the following symmetric $n \times n$ linear system

$$\nabla^2 \boldsymbol{f}(\boldsymbol{w}^{(k)}) \boldsymbol{p}^{(k)} = -\nabla \boldsymbol{f}(\boldsymbol{w}^{(k)}). \qquad (18)$$

For local stability, the search direction $\boldsymbol{p}^{(k)}$ is required to be a descent direction, which is true if the Hessian $\nabla^2 \boldsymbol{f}(\boldsymbol{w}^{(k)})$ is positive definite. If the Hessian matrix is not positive definite, or is close to being singular, $\boldsymbol{p}^{(k)}$ may be an ascent direction or may be excessively long.

In order to guarantee descent direction in the case of nonconvex objective function, we use the modified Cholessky factorization [1] [6], which automatically finds a diagonal matrix $\boldsymbol{\Gamma}$ such that the matrix $\nabla^2 \boldsymbol{f}(\boldsymbol{w}^{(k)}) + \boldsymbol{\Gamma}$ is positive definite. The iteration rule is given by

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} + \alpha \boldsymbol{p}^{(k)}, \qquad (19)$$

where the step size α is determined by the backtracking line search.

[1]The matlab code of modified Cholessky factorization by Brian Borchers is available at http://www.nmt.edu/ borchers/ldlt.html.

V. Numerical Experiments

We used 3 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. Data3 is composed of DNA microarray data with 95 dimension and 4027 genes [8], [9] and we reduced 95 dimension to 15 dimension by PCA in our experiments. For binary data and sound data, three mixture signals were generated using the mixing matrix $\boldsymbol{A}$ given by

$$\boldsymbol{A} = \begin{pmatrix} -0.4667 & 2.0636 & -0.5136 \\ 0.0680 & 2.3982 & -0.1961 \\ -2.5108 & 0.3002 & 0.2247 \end{pmatrix}, \qquad (20)$$

where the condition number of $\boldsymbol{A}$ 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.

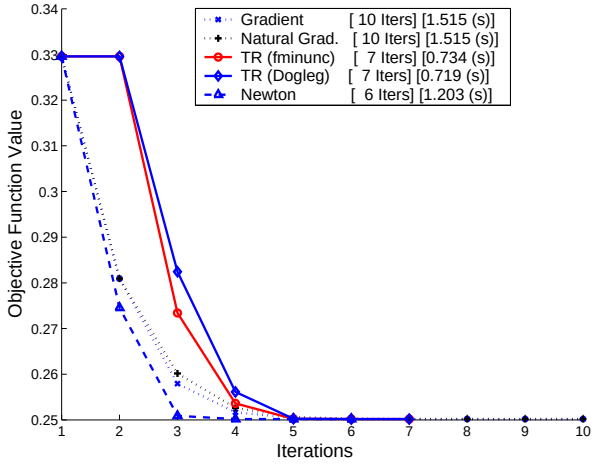

Fig. 3. Convergence comparison of several numerical optimization methods in the quasi maximum likelihood ICA for a set of binary data.

Fig. 3 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 Fig. 3, one can observe that the objective value after the first iteration, did not decrease in the TR-ICA algorithm. The reason being is the trust-region method sometimes need to control the size of the trust region without update. Once the size of the trust region is determined, the trust-region method

showed rapid convergence, compared to the gradient-based methods. The high CPU time in the gradient methods mainly came from the part of finding the optimal learning rate using the backtracking algorithm, in which at least several loops were required to find out the step length and to evaluate the objective value at each iteration.

For a set of microarray data (Data3), the convergence comparison is shown in Fig. 4. The natural gradient method achieved faster convergence than the gradient method in both iteration numbers and CPU time. Nevertheless, the trust-region method with the subspace method showed much faster convergence than the natural gradient method in iteration numbers as well as in CPU time. The dogleg method took less number of iterations but ate up more CPU time, compared to the gradient method. This resulted from that in the case of dogleg method, if the Hessian is not positive definite, then it throws the Hessian and uses gradient direction in trust region. The subspace method, however, can use the non-positive definite Hessian as stated in Sec. III.

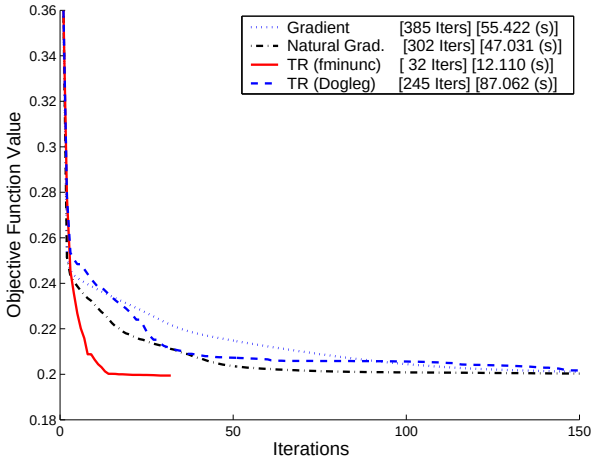

Fig. 4. Convergence comparison of several numerical optimization methods in the quasi maximum likelihood ICA for a set of DNA microarray data.

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

$$\text{PI} = \frac{1}{n(n-1)} \sum_{i=1}^{n} \frac{\sum_{j=1}^{n} |g_{ij}|}{\max_j |g_{ij}|} - 1, \qquad (21)$$

where g_{ij} is the (i,j)-element of the global system matrix $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{A}$. This measure is always between 0 and 1 and equal to zero if and only if there is a perfect match between $\boldsymbol{A}^{-1}$ and $\boldsymbol{W}$. Table. I 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.

VI. Conclusions

We have presented TR-ICA algorithms which employed the trust-region optimization scheme with the dogleg and the subspace method. Trust-region methods find a direction and a step simultaneously with the help of a quadratic model of the objective function, so do our TR-ICA algorithms.

TABLE I
Performance Index

Methods	Binary Data	Sound Signal
Gradient	2.120849e-003	6.173503e-003
Natural Grad.	2.119711e-003	7.293806e-003
TR fminunc	2.129067e-003	7.904882e-003
TR dogleg	2.120963e-003	7.906655e-003
Newton	2.120963e-003	7.906516e-003

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 (with the dogleg) algorithm ate up more CPU time due to its time complexity when the data dimensional grows, compared to the natural gradient ICA algorithm. But it required less CPU time, compared to the Newton method. The TR-ICA (with the subspace) showed the best convergence performance in terms of both iteration numbers and CPU time. In fact, our paper is the first application of the trust-region method to ICA. We are currently working on improving and speeding up TR-ICA algorithms.

Acknowledgment

This work was supported by Korea Ministry of Science and Technology under Brain Science and Engineering Research Program and under International Cooperative Research Program, by POSTECH BSRI Research Fund - 2004, by KOSEF 2000-2-20500-009-5, by ETRI, and by BK 21 in POSTECH.

References

[1] T. Akuzawa, "Extended quasi-Newton method for the ICA," in *Proc. ICA*, Helsinki, Finland, 2000, pp. 521–525.
[2] H. Attias and C. E. Schreiner, "Blind source separation and deconvolution: The dynamic component analysis algorithms," *Neural Computation*, vol. 10, pp. 1373–1424, 1998.
[3] J. F. Cardoso, "Infomax and maximum likelihood for source separation," *IEEE Signal Processing Letters*, vol. 4, no. 4, pp. 112–114, Apr. 1997.
[4] 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.
[5] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. John Wiley & Sons, Inc., 2002.
[6] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. New York: Academic Press, 1981.
[7] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley & Sons, Inc., 2001.
[8] H. Kim, S. Choi, and S. Bang, "Membership scoring via independent feature subspace analysis for grouping co-expressed genes," in *Proc. IJCNN*, Portland, Oregon, 2003.
[9] W. Liebermeister, "Linear modes of gene expression determined by independent component analysis," *Bioinformatics*, vol. 18, no. 1, pp. 51–60, 2002.
[10] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.
[11] 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.
[12] M. Zibulevsky, "Blind source separation with relative Newton method," in *Proc. ICA*, Nara, Japan, 2003, pp. 897–902.