RELATIVE TRUST-REGION LEARNING FOR ICA

Heeyoul Choi, Seungjin Choi

Department of Computer Science, POSTECH, Korea
{*hychoi,seungjin*}*@postech.ac.kr*

ABSTRACT

We present a new learning method, *relative trust-region learning*, where we incorporate the relative optimization technique [9] into the trust-region method. We apply this relative trust-region learning method to the problem of independent component analysis (ICA), which leads to the *relative TR-ICA* algorithm which turns out to be faster than Newton-type ICA algorithms as well as gradient-based ICA algorithms and to possess the equivariant property. Empirical comparisons with several existing ICA algorithms, confirm the fast convergence of the relative TR-ICA algorithm.

1. INTRODUCTION

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. 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. Given N data points, the model Eq. (1) is written as

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}, \tag{2}$$

where $\boldsymbol{X} = [\boldsymbol{x}(1), \ldots, \boldsymbol{x}(N)]$ and $\boldsymbol{S} = [\boldsymbol{s}(1), \ldots, \boldsymbol{s}(N)]$.

In general, ICA can be illustrated by a probability density matching problem [2]. The probability density matching is referred to as the Kullback matching when the Kullback-Leibler divergence is used as a measure of discrepancy between two probability distributions. The Kullback matching leads to the objective function that has the form

$$f(\boldsymbol{W}, \boldsymbol{X}) = -\log|\det \boldsymbol{W}| + \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{n}\psi_i(y_i(t)), \tag{3}$$

where $\boldsymbol{W} = \boldsymbol{A}^{-1}$, $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$, and $\psi_i(y_i(t)) = -\log p_i(y_i(t))$. The matrix $\boldsymbol{W}$ is referred to as a *demixing matrix* and the estimate of latent variable vector, $\boldsymbol{y}$, is restored up to the scaled and reordered version of the original hidden variable vector $\boldsymbol{s}$.

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 vec$(\cdot)$ is the *vec-function* which stacks the columns of the given matrix into one long vector. On the contrary, $\boldsymbol{W} = \text{mat}^T(\boldsymbol{w})$. For convenience, we abuse the function notation as follows

$$\begin{aligned} f(\boldsymbol{W}, \boldsymbol{X}) &= f(\boldsymbol{W}) = f(\boldsymbol{w}), \\ f(\boldsymbol{I}, \boldsymbol{Y}) &= f_r(\boldsymbol{W}) = f_r(\boldsymbol{w}), \end{aligned} \tag{4}$$

where the subscript r is used to emphasize that it involves the relative mode which will be described in detail later.

Popular ICA algorithms are based on the gradient or the natural gradient method [1]. 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. So, many other optimization methods have been applied to ICA.

In this paper, we present a relative trust-region learning method, where we incorporate the relative optimization into the trust-region method. We apply the relative trust-region learning method to the problem of ICA, which leads to the relative TR-ICA algorithm. The relative TR-ICA algorithm inherits various useful properties, such as the fast convergence, stability, and the equivariant property, from both conventional trust-region methods and the relative optimization. Moreover, we exploit a special structure of the Hessian matrix for memory-efficiency in the relative TR-ICA algorithm, so that the algorithm is useful, especially for the case of high-dimensional data. Several numerical examples confirm the high performance of our relative TR-ICA algorithm.

2. TRUST-REGION METHODS [8]

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. Let us consider an objective function $f(\boldsymbol{w}) : \mathbb{R}^{n^2} \rightarrow \mathbb{R}$ 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 $\boldsymbol{w}^{(k)}$. The search direction $\boldsymbol{p} \in \mathbb{R}^{n^2}$ is determined by solving the following subproblem:

$$\underset{\|\boldsymbol{p}\| \leq \triangle^{(k)}}{\arg\min}\ m^{(k)}(\boldsymbol{p}) = f^{(k)} + \left[\nabla \boldsymbol{f}^{(k)}\right]^T \boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T \boldsymbol{B}^{(k)}\boldsymbol{p}, \tag{5}$$

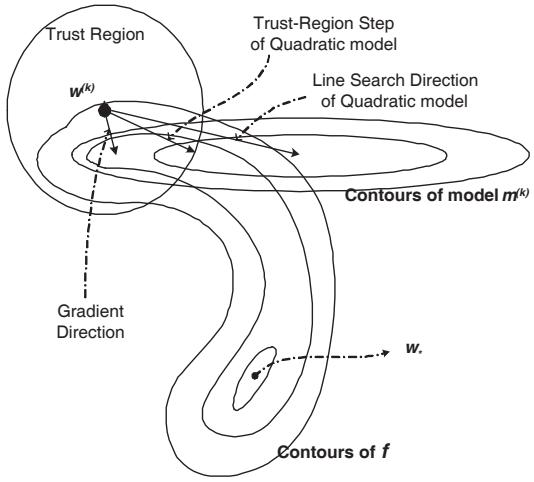

Fig. 1. An illustration of the trust-region method in determining a direction and a step size with the help of a quadratic model.

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

$$\begin{aligned} f^{(k)} &= f(\boldsymbol{w}^{(k)}), \\ \nabla \boldsymbol{f}^{(k)} &= \left.\frac{\partial f}{\partial \boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{w}^{(k)}}. \end{aligned} \tag{6}$$

The solution $\boldsymbol{p}_*^{(k)}$ of Eq. (5) is the minimizer of $m^{(k)}$ in the ball with its radius being $\triangle^{(k)}$. (See [5] for the details of Trust-Region method and TR-ICA).

3. RELATIVE OPTIMIZATION

This section describes the relative optimization method for ICA from the viewpoint of Lie group and equivariant property.

The serial updating where the parameters are learned in a multiplicative fashion, keeps the parameters in a group structure (especially Lie group). The relative gradient resulted from the idea of learning in Lie group, which was first investigated by Cardoso [3].

The general linear group of degree n, denoted by $GL(n)$, is a set of invertible (nonsingular) $n \times n$ matrices. The general linear group is an instance of Lie group. In the case of ICA, the parameter matrix $\boldsymbol{W}$ belongs to the general linear group $GL(n)$. A matrix $\boldsymbol{W}$ in $GL(n)$ gives rise to an invertible linear transformation $\Pi : \mathbb{R}^n \to \mathbb{R}^n$, defined by $\Pi(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x}$, and the matrix multiplication in the group corresponds to composition of linear transformations. Learning a demixing matrix $\boldsymbol{W}$ in ICA, can be carried out by a linear transformation of parameters, which leads to the following learning process

$$\boldsymbol{W}^{(k+1)} = \boldsymbol{E}^{(k)}\boldsymbol{W}^{(k)}, \tag{7}$$

where $\boldsymbol{E}^{(k)}$ is a linear transformation of parameters $\boldsymbol{W}^{(k)}$. The linear transformation $\boldsymbol{E}^{(k)}$ is computed such that an objective function (for instance, Eq. (3) in the case of ICA) is minimized on the Lie group. Moreover, a Lie group is a differentiable manifold obeying the group properties. Therefore, the multiplicative learning rule of $\boldsymbol{W}^{(k)}$ in Eq. (7) reflects a manifold. In fact, the natural gradient (which is identical to the relative gradient in ICA) was developed in the framework of learning in Riemannian manifold [1].

A family of adaptive ICA algorithms employs an update rule that has the form

$$\boldsymbol{W}^{(k+1)} = \boldsymbol{W}^{(k)} - \eta^{(k)} \widetilde{G}\left(\boldsymbol{X}, \boldsymbol{W}^{(k)}\right), \tag{8}$$

where $\widetilde{G}\left(\boldsymbol{X}, \boldsymbol{W}^{(k)}\right)$ is a matrix-valued function and $\eta^{(k)} > 0$ is a learning rate. Without loss of generality, the updating rule has the form

$$\boldsymbol{W}^{(k+1)} = \left(\boldsymbol{I} - \eta^{(k)} G\left(\boldsymbol{Y}^{(k)}\right)\right) \boldsymbol{W}^{(k)}. \tag{9}$$

If we denote the 'plugging' matrix $\left(\boldsymbol{I} - \eta^{(k)} G\left(\boldsymbol{Y}^{(k)}\right)\right)$ by $\boldsymbol{E}^{(k)}$, then the parameter matrix $\boldsymbol{W}^{(k+1)}$ can be decomposed into

$$\boldsymbol{W}^{(k+1)} = \boldsymbol{E}^{(k)}\boldsymbol{W}^{(k)} = \boldsymbol{E}^{(k)}\boldsymbol{E}^{(k-1)} \cdots \boldsymbol{E}^{(1)}\boldsymbol{E}^{(0)}\boldsymbol{W}^{(0)}, \tag{10}$$

where $\boldsymbol{W}^{(0)}$ is an initial value of $\boldsymbol{W}$. It follows from Eq. (10) that the serial update rule for $\boldsymbol{W}^{(k)}$ reflects a manifold. When the final convergence is achieved after c iterations, the stationary point $\boldsymbol{W}_*$ also consists of series of multiplications of matrices, $\boldsymbol{W}_* = \boldsymbol{E}_*\boldsymbol{W}^{(0)}$ where $\boldsymbol{E}_* = \boldsymbol{E}^{(c)}\boldsymbol{E}^{(c-1)} \cdots \boldsymbol{E}^{(1)}\boldsymbol{E}^{(0)}$. Even in the case of a different mixing matrix $\boldsymbol{A}'$ being involved, if we set the $\boldsymbol{W}^{(0)\prime}$ as an initial matrix of $\boldsymbol{W}$ such that $\boldsymbol{W}^{(0)}\boldsymbol{A} = \boldsymbol{W}^{(0)\prime}\boldsymbol{A}'$, then $\boldsymbol{W}_* = \boldsymbol{E}_*\boldsymbol{W}^{(0)\prime}$ and updating rules are identical to each other, which implies the uniform performance.

An estimator $\mathcal{A}$ for $\boldsymbol{A}$ is said to be equivariant if it satisfies

$$\mathcal{A}(\boldsymbol{M}\boldsymbol{X}) = \boldsymbol{M}\mathcal{A}(\boldsymbol{X}), \tag{11}$$

for any invertible $n \times n$ matrix $\boldsymbol{M}$. An important property induced by an equivariant estimator is the uniform performance which implies that the performance of an estimator does not depend on the mixing matrix $\boldsymbol{A}$ in ICA. Suppose that source signals are estimated as $\boldsymbol{y} = \hat{\boldsymbol{s}} = \boldsymbol{W}\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{x}$. Then, we have

$$\hat{\boldsymbol{s}} = (\mathcal{A}(\boldsymbol{X}))^{-1}\boldsymbol{x} = (\mathcal{A}(\boldsymbol{A}\boldsymbol{S}))^{-1}\boldsymbol{A}\boldsymbol{s} = \mathcal{A}(\boldsymbol{S})^{-1}\boldsymbol{s}. \tag{12}$$

Here, source signals estimated by an equivariant estimator $\mathcal{A}$ are given by $\hat{\boldsymbol{s}} = \mathcal{A}(\boldsymbol{S})^{-1}\boldsymbol{s}$, that is, they depend solely on original source signals $\boldsymbol{s}$. This equivariant property can be achieved by the 'serial update'.

The relative gradient involves the plugging matrix containing the first-order information (in the sense that the gradient is used). This can be generalized by computing the plugging matrix using other optimization methods (for instance, Newton method). The relative optimization [9] is summarized in Table 1.

Table 1. Relative Optimization Algorithm [9].

Start with an initial estimate $\boldsymbol{W}^{(0)}$;
repeat $k = 0, 1, 2, ...,$ until convergence
  $\boldsymbol{Y}^{(k)} = \boldsymbol{W}^{(k)}\boldsymbol{X}$;
  Starting with $\boldsymbol{V}^{(0)} = \boldsymbol{I}$ (identity matrix),
  Compute $\boldsymbol{V}^{(k)}$ which significantly decreases the objective function.
  Update $\boldsymbol{W}$ by $\boldsymbol{W}^{(k+1)} = \boldsymbol{V}^{(k)}\boldsymbol{W}^{(k)}$;
end (repeat)

4. A RELATIVE TRUST-REGION METHOD

Recently the relative gradient method was further elaborated in [9], which leads to the *relative Newton method* for ICA, and also trust-region method was applied to ICA, TR-ICA [5]. Here we illustrate how the relative optimization method is incorporated with the conventional trust-region methods, which is a main contribution of our paper. The method is described mainly for the case of ICA, which leads to the *relative TR-ICA* algorithm.

The relative trust-region method consists of two modes: (1) relative mode; (2) absolute mode. In the relative mode, we consider a modified quadratic model, $m_r^{(k)}$ where the subscript r represents the relative mode and find a direction $\boldsymbol{p}^{(k)}$ which solves the following subproblem modified from Eq. (5),

$$\underset{\|\boldsymbol{p}\| \leq \triangle^{(k)}}{\arg\min}\ m_r^{(k)}(\boldsymbol{p}) = f_r^{(k)} + \left[\nabla \boldsymbol{f}_r^{(k)}\right]^T \boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T \nabla^2 \boldsymbol{f}_r^{(k)} \boldsymbol{p}, \quad (13)$$

where $f_r(\boldsymbol{w}), \nabla \boldsymbol{f}_r(\boldsymbol{w})$ and $\nabla^2 \boldsymbol{f}_r(\boldsymbol{w})$ are calculated with $\boldsymbol{Y}^{(k)}$ and $\boldsymbol{I}$ instead of $\boldsymbol{X}$ and $\boldsymbol{W}^{(k)}$. For the ICA, we define an n-dimensional element-wise function $\psi(\boldsymbol{y}) \in \mathbb{R}^n$ with its ith element being $\psi_i(y_i) = -\log p_i(y_i)$ in Eq. (3). We denote the element-wise 1st- and 2nd-order derivatives of ψ by ψ' and ψ'', respectively. In relative mode, the function, the gradient and the Hessian of ICA are

$$f_r(\boldsymbol{w}) = \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{n}\psi_i(y_i(t)), \quad (14)$$

$$\nabla \boldsymbol{f}_r(\boldsymbol{w}) = \text{vec}\left(-\boldsymbol{I} + \frac{1}{N}\sum_{t=1}^{N}\boldsymbol{y}(t)\left[\psi'(\boldsymbol{y}(t))\right]^T\right), \quad (15)$$

$$\nabla^2 \boldsymbol{f}_r(\boldsymbol{w}) = \boldsymbol{H} + \boldsymbol{D}, \quad (16)$$

where $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ is the identity matrix and $\boldsymbol{D} \in \mathbb{R}^{n^2\times n^2}$ is defined in Eq. (18) and $\boldsymbol{H} \in \mathbb{R}^{n^2\times n^2}$ contains n^2 row vectors, $\vec{\boldsymbol{h}}_m$, defined in Eq. (17).

The Hessian of $f_r(\boldsymbol{w})$ has a special structure which permits a fast solution of the Newton system. Taking $\boldsymbol{V}^{(0)} = \boldsymbol{I}$ in Table 1 into account in the derivation, leads to

$$\vec{\boldsymbol{h}}_m = \left[\text{vec}\left(\boldsymbol{e}_j \boldsymbol{e}_i\right)\right]^T, \quad m = (i-1)n + j, \quad (17)$$

where $\boldsymbol{e}_j$ is the n-dimensional unit vector where only the jth element is 1 and other elements are zeros. If $\boldsymbol{Y}^{(k)}$ is close to the original source $\boldsymbol{S}$, then $\boldsymbol{D}_l$ becomes a diagonal matrix and the ith diagonal element of $\boldsymbol{D}_l$ is given by

$$[\boldsymbol{D}_l]_{ii} = \frac{1}{T}\sum_t \psi_l''(y_l(t)) y_i^2(t), \quad (18)$$

where $\boldsymbol{y}(t)$ is the current estimate of the source vector $\boldsymbol{s}(t)$.

A basic idea in modifying the quadratic model comes from the fact that the relative optimization considers $f(\boldsymbol{I}, \boldsymbol{Y})$ instead of $f(\boldsymbol{W}, \boldsymbol{X})$. The predicted reduction is also calculated in the relative mode, considering the modified quadratic model $m_r^{(k)}$. The actual reduction, however, should be calculated in the original objective function $f(\boldsymbol{w})$ instead of $f_r(\boldsymbol{w})$. Therefore, in the relative trust-region method, the agreement measure $\rho_r^{(k)}$ has the form

$$\rho_r^{(k)} = \frac{f\left(\boldsymbol{w}^{(k)}\right) - f\left(\boldsymbol{w}^{(k)} + \boldsymbol{p}^{(k)}\right)}{m_r^{(k)}(\boldsymbol{0}) - m_r^{(k)}\left(\boldsymbol{p}^{(k)}\right)}. \quad (19)$$

Table 2. Relative TR-ICA Algorithm.

Given $\hat{\triangle} > 0$, $\triangle^{(0)} \in (0, \hat{\triangle})$, and $\zeta \in [0, \frac{1}{4})$:
for $k = 0, 1, 2, \ldots$
(Relative Mode:)
$\boldsymbol{Y}^{(k)} = \boldsymbol{W}^{(k)}\boldsymbol{X}$
Obtain $\boldsymbol{p}^{(k)}$ by solving Eq. (13);
Calculate the predicted reduction in Eq. (19);
(Absolute Mode:)
Calculate the actual reduction in Eq. (19);

Evaluate $\rho_r^{(k)}$ in Eq. (19)
if $\rho_r^{(k)} < \frac{1}{4}$, then $\triangle^{(k+1)} = \frac{1}{4}\|\boldsymbol{p}^{(k)}\|$
else
if $\rho_r^{(k)} > \frac{3}{4}$ and $\|\boldsymbol{p}^{(k)}\| = \triangle^{(k)}$,
then $\triangle^{(k+1)} = \min(2\triangle^{(k)}, \hat{\triangle})$
else, then $\triangle^{(k+1)} = \triangle^{(k)}$;

if $\rho_r^{(k)} > \zeta$, then $\boldsymbol{W}^{(k+1)} = \text{mat}^T\left(\boldsymbol{p}^{(k)}\right)\boldsymbol{W}^{(k)}$
else, then $\boldsymbol{W}^{(k+1)} = \boldsymbol{W}^{(k)}$
end (for)

Then TR method determines the size of trust-region and finally parameters $\boldsymbol{W}$ are serially updated (like the relative optimization). The relative TR is summarized in Table 2.

In the relative mode, the direction $\boldsymbol{p}^{(k)}$ is computed by solving the modified subproblem (13) which is involved with the calculation of the Hessian matrix $\nabla^2 \boldsymbol{f}_r$ in order to compute $\left[\nabla^2 \boldsymbol{f}_r\right]^{-1}\nabla \boldsymbol{f}_r$ and $\boldsymbol{v}^T\nabla^2 \boldsymbol{f}_r$ where $\boldsymbol{v}$ is a gradient or other learning directions. In the case of high-dimensional data, the Hessian matrix takes a huge memory space. As in the fast relative Newton method [9], we use the modified Newton direction which is found at low computational complexity, in order to take care of $\left[\nabla^2 \boldsymbol{f}_r\right]^{-1}\nabla \boldsymbol{f}_r$. In addition, we show that the term $\boldsymbol{v}^T\nabla^2 \boldsymbol{f}_r$ in (13) can be easily computed, due to a special structure of $\boldsymbol{H}$ and $\boldsymbol{D}$ in Eq. (16).

5. NUMERICAL EXPERIMENTS

We used 2 different data sets for our numerical experiments. The first set of data contains the mixtures of two speech signals and one music signal, all of them were sampled at 8 kHz. The second data set is the USPS data which contains handwritten digits that are converted to 256-dimensional data ($16 \times 16 = 256$) of length 4000.

The optimization methods in ICA that we compared our relative TR-ICA with, include (1) the gradient; (2) the natural gradient; (3) TR-ICA (with the dogleg implementation); (4) the fast relative Newton [9]. In the gradient, the natural gradient and Newton's methods, an optimal learning rate was determined by the backtracking line search method (see [5] for the details).

5.1. Ill-conditioned Mixing Matrix

Relative optimization (or the natural gradient) was shown to have the equivariant property [4, 1, 6, 9] where the performance did not depend on the condition of a mixing matrix. In the experiment, 3-dimensional sound signals were artificially mixed using an ill-conditioned mixing matrix where its condition number is 525.44.

Fig. 2 shows the convergence behavior of various ICA algorithms for the case of 3-dimensional sound data. The relative algorithms (both Newton and trust-region) made convergence much faster than the gradient-based algorithms. Once again, the relative

TR-ICA algorithm was the fastest one among all algorithms that we tested.

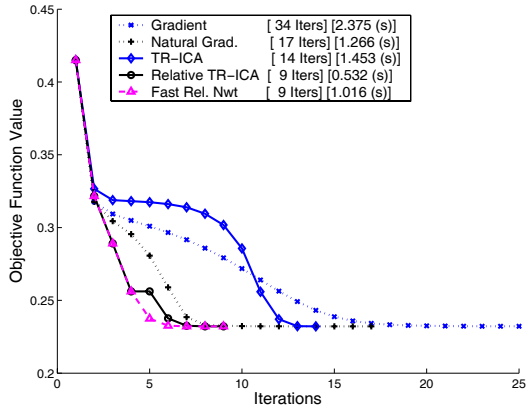

Fig. 2. Convergence comparison of several numerical optimization methods in the quasi maximum likelihood ICA for a set of 3-dimensional sound data (ill-conditioned mixing matrix).

5.2. High-Dimensional Data

For the case of USPS data set, the computation of Hessian matrix at each iteration is very expensive and the Hessian matrix is not positive definite at some iteration steps. Moreover, if dimension is very high (more than 30), then conventional trust-region methods cause a memory problem. However, the relative TR-ICA algorithm overcomes these limits by using a trick for memory-efficiency.

Table 3 shows that our proposed algorithm is faster than any other methods in high dimensional data set. In the numerical experiment, we chose the portion associated with the digit '2' and reduced the dimension by PCA, which produced 100-dimensional data of length 379. In such a case, FastICA [7] had difficulty in convergence because the number of data points were not enough. The small number of data points does not guarantee the objective function to be a contraction map which is a necessary condition for the fixed point algorithm. On the other hand, the natural gradient and the relative ICA algorithms including our proposed algorithm worked fine. In such high-dimensional data, our relative TR-ICA algorithm showed fastest convergence. For the measure of convergence, we specified the error to be inside the tolerance level in advance (here we use 10^{-5}). Moreover, in Table 3, even the iteration number of relative TR-ICA is much less than fast relative Newton's, as well as the CPU time. This means that the objective functions for high-dimensional real data have difficulty to be modelled with quadratic equation, so the recommended learning direction of trust-region method is better than Newton's.

6. CONCLUSION

We have introduced a relative trust-region method which jointly exploited the trust-region method and the relative optimization. In the relative trust-region method, a direction and a step size were searched with the help of a quadratic model (just like the trust-region method) and the parameters were serially updated (relative optimization). We have applied this relative trust-region method to the problem of ICA, which led to the relative TR-ICA algorithm which consisted of the relative mode and the absolute mode. The relative TR-ICA algorithm enjoyed fast convergence, compared to most of existing ICA algorithms and showed the equivariant property like the natural or the relative gradient. Moreover exploiting a special structure of the Hessian matrix in the relative mode led to a memory-efficient relative TR-ICA algorithm which could handle high-dimensional data. In experiment, the useful behavior and the high performance of the relative TR-ICA algorithm were observed.

Although we focused on ICA, the relative trust-region method could be applied to other machine learning problems.

Table 3. Convergence comparison of several numerical optimization methods in the quasi maximum likelihood ICA for digit '2' among USPS data (Dimension is reduced into 100 by PCA).

Methods	Iteration Number	Total Time
Natural Grad.	3415	731.375(s)
Relative TR	763	109.063(s)
Fast Rel. Newton	1147	269.407(s)

7. ACKNOWLEDGMENT

This work was supported by Korea Ministry of Science and Technology under Brain Science and Engineering Research Program and International Cooperative Research Program and by Systems Bio-Dynamics Research Center under NCRC program and BSRI Research Fund - 2004, KOSEF 2000-2-20500-009-5.

8. REFERENCES

[1] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[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, "Learning in manifolds: The case of source separation," in *Proc. SSAP*, Portland, Oregon, 1998.

[4] J. F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing*, vol. 44, no. 12, pp. 3017–3030, Dec. 1996.

[5] H. Choi, S. Kim, and S. Choi, "Trust-region learning for ICA," in *Proc. Int'l Joint Conf. Neural Networks*, Budapest, Hungary, 2004, pp. 41–46.

[6] S. Choi, A. Cichocki, and S. Amari, "Equivariant nonstationary source separation," *Neural Networks*, vol. 15, no. 1, pp. 121–130, 2002.

[7] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Trans. Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

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

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