Multiplicative Nonholonomic/Newton -like Algorithm

Toshinao Akuzawa*

and

Noboru Murata

Brain Science Institute
RIKEN
2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan

October 19, 1999

Abstract

We construct new algorithms from scratch, which use the fourth order cumulant of stochastic variables for the cost function. The multiplicative updating rule here constructed is natural from the homogeneous nature of the Lie group and has numerous merits for the rigorous treatment of the dynamics. As one consequence, the second order convergence is shown. For the cost function, functions invariant under the componentwise scaling are choosen. By identifying points which can be transformed to each other by the scaling, we assume that the dynamics is in a coset space. In our method, a point can move toward any direction in this coset. Thus, no prewhitening is required.

1 Introduction

Suppose that N-dimensional stochastic variables $\{X_i|1 \leq i \leq N\}$ are observed. The independent component analysis (ICA) pursues a map $X \mapsto Y$, where each component of Y becomes mutually independent. In this letter we restrict ourselves to the linear independent component analysis. There we want to find a linear transformation $C : \mathbf{X} = (X_1, \cdots, X_N)' \mapsto \mathbf{Y} = (Y_1, \cdots, Y_N)' = C\mathbf{X}$ which minimizes some cost function that measures the independence. Hereafter we denote by the upper subscript $'$ the transposition and by $\dagger$ the complex conjugate.

*akuzawa@islab.brain.riken.go.jp

There can be many candidates for the cost function. For example the Kullback-Leibler information is a good measure for the independence. In this case the problem is translated to the minimization of $-\sum_{i=1}^{N}\int dy_i P_i(y_i)\ln P_i(y_i)$, where P_i is the probability density function of the i-th component. It is obvious that we must evaluate P_i's to find the optimal solution. A robust estimation of the probability density functions is not an easy task and if it is possible it may be computationally expensive.

An alternative idea is to make use of the cumulant of the fourth order, or the kurtosis[A.Hyvärinen,1997], which we will adopt in this letter. The fourth order cumulant vanishes for the normal distribution. So, this cost function is robust under the gaussian random noises. We will construct algorithms where a matrix, which specifies the linear transformation, is updated by the left-multiplication of a matrix $D = \mathrm{e}^{\Delta}$. This expression implies that D belongs to $GL(N,R)$ (more accurately, the component of $GL(N,R)$ connected to the unit element), which ensures the conservation of the rank. The specification of D by the coordinate Δ has many advantages since it has a compatibility with the homogeneous nature of the Lie group.

There are variations for the form of the cost function. We will show our definitions in the following two sections, which are choosen to possess invariance under componentwise scaling. This invariance is crucial for a rigorous treatment of the convergence properties. Moreover, this invariance allows us to identify points in $GL(N,R)$ which is transformed to each other by the scaling. Then we can legitimately restrict the dynamics to a coset space which is introduced by this identification.

Under these settings, we determine Δ by using the Newton method for the second order expansion of the cost function with respect to $\{\Delta_{ij}\}$. It is assumed that the diagonal elements of Δ are zeros, which does not impose any restrictions. That is, a point can move toward any direction in this coset by a left-multiplication of e^{Δ}. Thus it is not necesarry for our method to prewhiten the data. It is also not required that the optimal solution is the maximum or the minimum of the cost function. Indeed, the sole requirement is that the optimal point is a saddle point of the cost function since our method is in principle the Newton method. These are great advantages of our method.

Our strategy is as follows. As an initial condition we set C_0. For $t > 0$ ($t \in \mathbf{N}^+$), we introduce an $N \times N$ matrix Δ_t and denote C_t as $C_t = \mathrm{e}^{\Delta_t} C_{t-1}$. Next, we evaluate the cost function at C_t by using the expansion around C_{t-1} with respect to the elements of Δ_t up to the second order. Then Δ_t is choosen as a saddle point of this second order expansion. We iteratively follow these procedures until we obtain a satisfactory solution.

This letter is organized as follows. In Section 2 the main part of our algorithm is constructed, where the cost function is essentially identical to the sum of kurtoses. We adopt the square of the kurtoses for the cost function in Section 3. Explicit expressions for the optimal Δ (up to the second order) are obtained both in Sections 2 and 3. Section 4 is a short section where we show how each updating step is combined to obtain the optimal C. In Section 5 the convergence property of our algorithm is discussed. Section 6 contains conclusions and discussions.

2 Multiplicative update algorithm

2.1 Expansion of the cost function

Let us start by defining the cost function:

$$f(C,X) = \sum_i f_i(C,X) \ , \tag{2.1}$$

where f_i's are the fourth order moments of components divided by the square of their variances,

$$f_i(C,X) = \frac{E((CX)_i^4)}{E((CX)_i^2)^2} \ . \tag{2.2}$$

In this letter we denote by $E(A)$ the expectation of A. Obviously the cost function f coincides with the sum of kurtoses of all the components up to the constant. We set $D = \mathrm{e}^{\Delta}$ and expand $f(D,Y)$ in terms of the elements of Δ. For example expansions term by term are evaluated as follows:

$$\begin{aligned} E((DY)_i^4) &= E(Y_i^4) + 4\sum_p (\Delta_{ip} + (\frac{\Delta^2}{2})_{ip}) E(Y_i^3 Y_p) + 6\sum_{p,q} \Delta_{ip}\Delta_{iq} E(Y_i^2 Y_p Y_q) + O(\Delta^3) \\ E((DY)_i^2) &= E(Y_i^2) + 2\sum_p (\Delta_{ip} + (\frac{\Delta^2}{2})_{ip}) E(Y_i Y_p) + \sum_{p,q} \Delta_{ip}\Delta_{iq} E(Y_p Y_q) + O(\Delta^3) \ . \end{aligned} \tag{2.3}$$

Hereafter we denote by $O(\Delta^k)$ polynomials of matrix elements of Δ which does not contain terms with degrees less than k. For brevity's sake we introduce the following notations:

$$\sigma_i^{(k)} = |E(Y_i^k)|^{1/k} \ , \tag{2.4}$$

$$R_{pi}^{(k)} = \frac{E(Y_i^k Y_p)}{(\sigma_i^{(2)})^{k+1}} \ , \tag{2.5}$$

$$U_{pq}^{(k,i)} = \frac{E(Y_i^k Y_p Y_q)}{(\sigma_i^{(2)})^{k+2}} \ , \tag{2.6}$$

and

$$\kappa_i = (\sigma_i^{(4)})^4 / (\sigma_i^{(2)})^4 \ . \tag{2.7}$$

Using the quantities defined above we can show that the cost function is expanded as

$$\begin{aligned} f_i(D,Y) &= \left[\kappa_i + 4\big[(\Delta + \frac{\Delta^2}{2})R^{(3)}\big]_{ii} + 6\big[\Delta U^{(2,i)}\Delta'\big]_{ii} + O(\Delta^3)\right] \\ &\quad \times \left[1 - 4\big[(\Delta + \frac{\Delta^2}{2})R^{(1)}\big]_{ii} - 2\big[\Delta U^{(0,i)}\Delta'\big]_{ii} + 12\big[\Delta R^{(1)}\big]_{ii}^2 + O(\Delta^3)\right] \\ &= \kappa_i - 4\big[(\Delta + \frac{\Delta^2}{2})(\kappa_i R^{(1)} - R^{(3)})\big]_{ii} + 2\big[\Delta(3U^{(2,i)} - \kappa_i U^{(0,i)})\Delta'\big]_{ii} \\ &\quad + 12\kappa_i\big[\Delta R^{(1)}\big]_{ii}^2 - 16\big[\Delta R^{(1)}\big]_{ii}\big[\Delta R^{(3)}\big]_{ii} + O(\Delta^3) \end{aligned} \tag{2.8}$$

by straightforward calculations. Next, we evaluate partial derivatives of the cost function by the matrix elements of Δ. Partially differentiating (2.8), we get an expression,

$$
\begin{aligned}
&\frac{\partial f(\mathrm{e}^{\Delta}, Y)}{\partial \Delta_{kl}} = -4\big[K - R^{(3)}\big]_{lk} - 2\big[(K - R^{(3)})\Delta + \Delta(K - R^{(3)})\big]_{lk} \\
&+4\big[(3U^{(2,k)} - \kappa_k U^{(0,k)})\Delta'\big]_{lk} + 24K_{lk}\big[\Delta R^{(1)}\big]_{kk} - 16R^{(1)}_{lk}\big[\Delta R^{(3)}\big]_{kk} - 16R^{(3)}_{lk}\big[\Delta R^{(1)}\big]_{kk} \\
&+O(\Delta^2)\ ,
\end{aligned}
\tag{2.9}
$$

where K is an $N \times N$ matrix defined by

$$
K_{pq} = \kappa_q R^{(1)}_{pq}\ . \tag{2.10}
$$

We want to decide Δ for which the partial derivative by Δ_{kl} $(k \neq l)$ of the cost function vanish on condition that $\Delta_{ii} = 0$ for $1 \leq i \leq N$. We neglect $O(\Delta^3)$ terms in the cost function. Thus the right-hand side of (2.9) is regarded as a polynomial of $\{\Delta_{kl}\}$ of at most first order and it is always possible in principle to determine Δ which satifies the above condition. It is, at the same time, not easy to describe the problem in a form which is valid for arbitrary N. In the following subsection we will introduce a transparent and unified method for handling the partial derivatives of f. We leave this subsection by introducing $N \times N$ matrices

$$
V^{(i)} = 3U^{(2,i)} - \kappa_i U^{(0,i)} \tag{2.11}
$$

and

$$
Q = K - R^{(3)} \tag{2.12}
$$

for later convenience.

2.2 Expression by tensor product and determination of Δ

The expression (2.9) is quite complicated and not convenient for our purpose, " determine Δ, where all the partial derivatives vanish". Fortunately by mapping the relations between elements of $N \times N$ matrices to those of $N^2 \times N^2$ matrices, we can handle the problem transparently. Some preparations are needed. First, let us introduce a map cs:

$$
\begin{array}{rcl}
\mathrm{Mat}(N, F) & \to & F^{N^2} \\
A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & \multicolumn{3}{c}{\cdots\cdots\cdots\cdots\cdots} \\ \multicolumn{4}{c}{\cdots\cdots\cdots\cdots\cdots\cdots\cdots} \\ A_{N1} & \multicolumn{2}{c}{\cdots\cdots\cdots} & A_{NN} \end{pmatrix} & \mapsto & \mathrm{cs}(A) = (A_{11}\ A_{21}\ \cdots\ A_{N1}\ A_{12}\ A_{22}\ \cdots\ A_{NN})'\ ,
\end{array}
\tag{2.13}
$$

where F is an unspecified field. We also introduce two useful operators T and P. The "intertwiner" T is an $N^2 \times N^2$ matrix defined by

$$\mathrm{cs}(A') = T\mathrm{cs}(A) \text{ for } A \in \mathrm{Mat}(N, F) \ . \tag{2.14}$$

The projection operator P,

$$\begin{aligned} P &= \mathrm{diag}(p_1, \cdots, p_{N^2}) \ , \\ & \begin{cases} p_k = 1 & \text{for } \ k = N(i-1)+i, 1 \le i \le N \\ p_k = 0 & \text{otherwise} \ , \end{cases} \end{aligned} \tag{2.15}$$

is used to extract the "diagonal" elements of a matrix from its image by cs.

On this setting we can rewrite (2.9) as

$$\begin{aligned} \frac{\partial f(\mathrm{e}^{\Delta}, Y)}{\partial \Delta_{kl}} &= \Big[-4\mathrm{cs}(Q) - 2\big[I_N \otimes Q + T(I_N \otimes Q')T\big]\mathrm{cs}(\Delta) + 4\big\{\bigoplus_{i=1}^{N} V^{(i)}\big\}\mathrm{cs}(\Delta') \\ &\quad + \Big\{ 24(I_N \otimes K)P(I \otimes R^{(1)})' - 16(I_N \otimes R^{(1)})P(I \otimes R^{(3)})' \\ &\quad - 16(I_N \otimes R^{(3)})P(I \otimes R^{(1)})' \Big\} \mathrm{cs}(\Delta') \Big]_{l+N(k-1)} \ , \end{aligned} \tag{2.16}$$

where I_N is the $N \times N$ unit matrix and

$$\bigoplus_{i=1}^{N} V^{(i)} = \begin{pmatrix} V^{(1)} & 0 & & \cdots\cdots & 0 \\ 0 & V^{(2)} & 0 & & \cdots\cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots\cdots & & V^{(N-1)} & 0 \\ 0 & 0 & & \cdots\cdots & V^{(N)} \end{pmatrix} \ . \tag{2.17}$$

We make use of the following fact:
For $X \in \mathrm{Mat}(N, F)$

$$T(I_N \otimes X)T = X \otimes I_N \ . \tag{2.18}$$

See Appendix A for the proof of (2.18). Then (2.16) becomes

$$\frac{\partial f(\mathrm{e}^{\Delta}, Y)}{\partial \Delta_{kl}} = -4[\mathrm{cs}(Q)]_{l+N(k-1)} + \big[W\mathrm{cs}(\Delta)\big]_{l+N(k-1)} \ , \tag{2.19}$$

where

$$
\begin{aligned}
W &= -2\big(I_N \otimes Q + Q' \otimes I_N\big) + 4\{\bigoplus_{i=1}^{N} V^{(i)}\}T + \Big[24(I_N \otimes K)P(I \otimes R^{(1)})' \\
&\quad -16(I_N \otimes R^{(1)})P(I \otimes R^{(3)})' - 16(I_N \otimes R^{(3)})P(I \otimes R^{(1)})'\Big]T \ .
\end{aligned}
\tag{2.20}
$$

Now let us determine Δ. Remember that we are going along the spirit of the Newton method. Thus we want to find Δ which satisfies the conditions

$$
\frac{\partial f(\mathrm{e}^{\Delta}, Y)}{\partial \Delta_{kl}} = 0 + O(\Delta^2) \ \text{ for } 1 \le k, l \le N,\ k \ne l \tag{2.21}
$$

and

$$
\Delta_{kk} = 0 \ \text{ for } \ 1 \le k \le N \ . \tag{2.22}
$$

The conditions (2.22) make the problem rather complicated one. Fortunately, by using P we can combine the conditions (2.21) and (2.22) into a matrix equation :

$$
\Big[(I_{N^2} - P)W(I_{N^2} - P) + P\Big]\mathrm{cs}(\Delta) - 4(I_{N^2} - P)\mathrm{cs}(Q) = 0 \ . \tag{2.23}
$$

Immediately it follows that

$$
\mathrm{cs}(\Delta) = 4\Big[(I_{N^2} - P)W(I_{N^2} - P) + P\Big]^{-1}(I_{N^2} - P)\mathrm{cs}(Q) \ . \tag{2.24}
$$

Thus we have obtained Δ which specify a saddle point of the expansion of $f(C, Y)$ up to the second order. Note that quantities in the right-hand side of (2.24) are easily estimated ones from the observed data. So, an updating is determined by (2.24) without any ambiguities.

3 Case II: square of kurtosis

Obviously, points where kurtosis vanishes do not play any special role for the cost function f in Section 2. The optimal solution, however, contains components with zero kurtoses when the number of the sources is less than that of the observation channels. Thus, in this section we treat with a slightly different cost function, which is the sum,

$$
\boldsymbol{f}(C, X) = \sum_i \boldsymbol{f}_i(C, X) \ , \tag{3.1}
$$

of the square of the kurtoses,

$$\boldsymbol{f}_i(C,X) = \left[\frac{E((CX)_i^4)}{E((CX)_i^2)^2} - 3\right]^2 \ . \tag{3.2}$$

As in the last section, we want to know the saddle point $D = \mathrm{e}^{\Delta}$ of the expansion of $\boldsymbol{f}_i(D,Y)$ in terms of $\{\Delta_{ij}\}$ up to the second order. We do not describe details of the calculations in this section, which is carried out almost in the same way as in Section 2. First, the expansion of $\boldsymbol{f}_i(D,Y)$ is evaluated as

$$\begin{aligned} \boldsymbol{f}_i(D,Y) &= (\kappa_i-3)^2 - 8\big[(\Delta + \frac{\Delta^2}{2})(R^{(1)}\kappa_i - R^{(3)})\big]_{ii}(\kappa_i-3) \\ &\quad +4\big[\Delta(3U^{(2,i)} - \kappa_i U^{(0,i)})\Delta'\big]_{ii}(\kappa_i-3) + 16\big[\Delta(R^{(1)}\kappa_i - R^{(3)})\big]_{ii}^2 \\ &\quad +24(\kappa_i-3)\kappa_i\big[\Delta R^{(1)}\big]_{ii}^2 - 32(\kappa_i-3)\big[\Delta R^{(1)}\big]_{ii}\big[\Delta R^{(3)}\big]_{ii} + O(\Delta^3) \ . \end{aligned} \tag{3.3}$$

Next, we introduce $N\times N$ matrices $\boldsymbol{K}$, $\{\boldsymbol{V}^{(i)}|1\le i\le N\}$, $\boldsymbol{S}$, and $\boldsymbol{Q}$ defined respectively by

$$\boldsymbol{K}_{pq} = 2R^{(1)}_{pq}(\kappa_q - 3)\kappa_q \ , \tag{3.4}$$

$$\boldsymbol{V}^{(i)} = 2(\kappa_i-3)(3U^{(2,i)} - \kappa_i U^{(0,i)}) \ , \tag{3.5}$$

$$\tag{3.6}$$

$$\boldsymbol{S} = \mathrm{diag}(2(\kappa_i - 3)) \ , \tag{3.7}$$

and

$$\boldsymbol{Q}_{pq} = 2(\kappa_q-3)(R^{(1)}_{pq}\kappa_q - R^{(3)}_{pq}) \ . \tag{3.8}$$

We also rewrite Q in (2.12) by $\boldsymbol{q}$ in order to avoid confusions:

$$\boldsymbol{q}_{pq} = (R^{(1)}_{pq}\kappa_q - R^{(3)}_{pq}) \ . \tag{3.9}$$

Now we proceed to the expression by using the tensor product. We can show that the gradients of the cost function have the following expression:

$$\frac{\partial \boldsymbol{f}(\mathrm{e}^{\Delta},Y)}{\partial \Delta_{kl}} = -4[\mathrm{cs}(\boldsymbol{Q})]_{l+N(k-1)} + \big[\boldsymbol{W}\mathrm{cs}(\Delta)\big]_{l+N(k-1)} + O(\Delta^2) \ , \tag{3.10}$$

where

$$\begin{aligned}
\boldsymbol{W} \ &= \ -2\big(I_N \otimes \boldsymbol{Q} + \boldsymbol{Q}' \otimes I_N\big) + 4\{\bigoplus_{i=1}^{N} \boldsymbol{V}^{(i)}\}T + \Big[24(I_N \otimes \boldsymbol{K})P(I \otimes R^{(1)})' \\
&\quad +32(I_N \otimes \boldsymbol{q})P(I_N \otimes \boldsymbol{q})' - 16(I_N \otimes R^{(1)}\boldsymbol{S})P(I \otimes R^{(3)})' \\
&\quad -16(I_N \otimes R^{(3)}\boldsymbol{S})P(I \otimes R^{(1)})'\Big]T \ . \qquad (3.11)
\end{aligned}$$

This is a completely analogous expression to (2.19). Thus the coordinate Δ of the saddle point of the second order expansion is determined by

$$\mathrm{cs}(\Delta) = 4\Big[(I_{N^2} - P)\boldsymbol{W}(I_{N^2} - P) + P\Big]^{-1}(I_{N^2} - P)\mathrm{cs}(\boldsymbol{Q}) \ . \qquad (3.12)$$

In many cases obtained through the two cost functions in Section 2 and Section 3 are almost the same results. As implied at the beginning of this section, the main difference between these two lies in the points where the kurtosis of one of the components vanishes. These point indeed constitue saddle points of the cost function f, while it is impossible to capture them by the algorithm in Section 2. Thus, we must choose an appropriate method for individual problems having this differnce in mind.

4 Iteration of updating

Now we have obtained the updating rules. It is not necessary to tune the learning rate. Apparently, (2.23) and (3.12) look complicated. They are, however, easily implemented by the numerical tools like MatLab. (The source codes will be available from our Web-site.) Starting from C_0, C_i for positive i is determined by the left multiplication by e^{Δ_i}, where Δ is determined by setting $Y = C_{i-1}X$, i.e,

$$C_t = \mathrm{e}^{\Delta_t}\mathrm{e}^{\Delta_{t-1}}\mathrm{e}^{\Delta_{t-2}} \cdots \mathrm{e}^{\Delta_1}C_0 \ . \qquad (4.1)$$

If Δ becomes saficiently small, we can stop the iteration and exit the process.

5 Second order convergence

First, we will take over the notations in Section 2. The following discussion is, however, valid for the algorithm in Section 3 if we substitute the quantities f, W, and so on by their boldface counterparts. Let us start this section by introducing some additional notations. We set

$$G \in GL(N, R) \qquad (5.1)$$

and

$$K \in GL(1,R)^{\oplus N} \ . \tag{5.2}$$

We also define the coset space $K\backslash G$ by introducing the equivalence relation

$$g' g^{-1} \in K \Longleftrightarrow g \sim g' \tag{5.3}$$

to G. That is, $K\backslash G \cong \{Kg | g \in G\}$. Our method is understood as an orthodox adaptation of the Newton method to this coset space $K\backslash G$. Note that the cost function $F(\cdot) \stackrel{\text{def}}{=} f(\cdot, Y)$ on G satisfies the relation

$$F(g) = F(Kg) \ . \tag{5.4}$$

So F is naturally considered as a function on $K\backslash G$. That is the reason of our choice for the cost function. Thus, the second-order convergence immediately follows if the the correction to the error with respect to the coordinating resulting from the multiplicative nature is properly evaluated.

At time t, a point g on $K\backslash G$ is specified by the coordinate $X^{(t)}(g) \in \mathfrak{m}$ such that

$$\mathrm{e}^{X^{(t)}(g)} C_t \sim g \ , \tag{5.5}$$

where $\mathfrak{m}$ is the set of $N \times N$ matrices whose diagonal elements are zeros. Actually, this statement itself is not a thing of course, for which the proof will be given elsewhere. Define F_t, the representation of the cost function at t, by

$$F_t(X) = F(\mathrm{e}^X C_t) \ . \tag{5.6}$$

Here we introduce an $(N^2 - N) \times N^2$ matrix $\tilde{P}$ by drawing out the $i + N(i-1)$-th raws from the unit $N^2 \times N^2$ matrix where $i = N, N-1, \cdots, 2, 1$. We will denote by $H^{(t)}$ the Hessian,

$$H^{(t)}_{kl} = \frac{\partial^2 F_t(X)}{\partial(\tilde{P}\mathrm{cs}(X))_k \partial(\tilde{P}\mathrm{cs}(X))_l} \tag{5.7}$$

Note that if we set

$$h_t(X) = T\Big((I_{N^2} - P)W(I_{N^2} - P) + P\Big)\Big|_{C = \mathrm{e}^X C_t} \ , \tag{5.8}$$

the Hessian is written as

$$H^{(t)} = \tilde{P} h_t \tilde{P}' \ . \tag{5.9}$$

Suppose that at some neighborhood of the optimal solution g_*, $H^{(t)}(X)$ is Lipschitz continuous for some t:

$$||H^{(t)}(X) - H^{(t)}(X')|| \leq L||X - X'|| \ , \tag{5.10}$$

where $||A||$ is the norm of a matrix A as the Euclidian space,

$$||A||^2 = \mathrm{tr}(AA^\dagger) \ . \tag{5.11}$$

We set

$$\beta = ||H^{(t)}(X^t(g_*))^{-1}|| \ . \tag{5.12}$$

There exists a positive real number r, for which

$$||H^{(t)}(X^t(g))^{-1}|| < 2\beta \ \text{ for } \forall g \in B^{(t)}(g_*, r) \stackrel{\text{def}}{=} \left\{ g \middle| r > ||X^t(g) - X^t(g_*)|| \right\} \tag{5.13}$$

is satisfied. Then it is known that for all $g \in B(g_*, \min(r, (2\beta L)^{-1}))$,

$$||X^t(C_{t+1}) - X^t(g_*)|| \leq \beta L ||X^t(C_t) - X^t(g_*)||^2 \tag{5.14}$$

and

$$||X^t(C_{t+1}) - X^t(g_*)|| \leq \frac{1}{2} ||X^t(C_t) - X^t(g_*)|| \tag{5.15}$$

are fulfilled. Thus the second order convergence in this norm is shown. Unfortunately, this norm is not invariant and is unnatural. (A natural metric on $K\backslash G$ is one which is invariant under the parallel transformation, which is induced by the action of elements in $K\backslash G$ from the right-hand side.) But, it suffices in practice.

6 Discussions

6.1 Nonholonomy?

Our method is related to the nonholonomic method by Amari, Chen, and Chichocki[Amari *et al.*,1997]. In essence our method is a Newton approach to the same problem, the optimization without prewhitening. Let us set

$$\mathrm{e}^z = \mathrm{e}^x \mathrm{e}^y \tag{6.1}$$

for $x, y \in \mathfrak{gl}(N, R)$. Then it is obvious that z does not necessarily belongs to $\mathfrak{m}$ even if $x, y \in \mathfrak{m}$(, that is, z_{ii}'s do not always vanish when $x_{ii} = y_{ii} = 0$ for $1 \leq i \leq N$). This may be explained by using the concept of nonholonomy. The degree of freedom in each step, however, equals the dimension of the space $K\backslash G$ in our setting. The nonholonomic nature emerges when we go back to $G = GL(N, \boldsymbol{R})$ again.

There exist several studies[M.Takeuchi,1994, S.Helgason,1978, S.Helgason,1962, S.Helgason,1984, T.Akuzawa & M.Wadati,1998] which deal with cosets like $K\backslash G$ or the right coset G/K when K is a maximal compact subgroup of G. Unfortunately, what we are studying is the case where K is not a maximal compact subgroup of G. So, for example it is necessary to show whether the coordinate (5.5) is justified or not. As mentioned above, further studies including this justification will appear elsewhere.

6.2 Global convergence

We should carefully treat first few steps since this method also has a somewhat undesirable global convergent property inherent in the Newton method. Fortunately enough, there exist methods which can handle the earlier stage. For example, the nonholonomic gradient method[Amari *et al.*,1997] may be applicable. Another posiibility is to construct a nonholonomic fixed-point algorithm which uses the kernel method. These methods are suitable for capturing the optimal point which contains components with zero kurtoses. There we must, of course, use the method in Section 3. If it is not necessary to worry about these zero kurtosis components, there is little difference between the two methods described in Section 2 and Section 3.

6.3 Conclusions

We have constructed a new algorithm for finding a optimal point in a matrix space, where we have introduced a new multiplicative updating method. The algorithm is in essence the Newton method on a coset. So it converges quite rapidly and it can capture the saddle point. Since it does not require prewhitening, it is not necessary to worry about the error resulting from the prewhitening. Indeed, it is possible to increase the kurtosis slightly for data preprocessed by the FastICA[Hurri *et al.*,1998].

References

[A.Hyvärinen,1997] A.Hyvärinen (1997). A Fast Fixed-Point Algorithm for Independent Component Analysis. *Neural Computation*, *9*, 1483–1492.

[Amari *et al.*,1997] Amari, S., Chen, T.-P., & Cichocki, A. (1997). Non-holonomic Constraints in Learning Algorithms for Blind Source Separation. *preprint*.

[Hurri *et al.*,1998] Hurri, J., Gävert, H., Sälelä, J., & Hyvärinen, A. (1998). FastICA package for MATLAB. http://www.cis.hut.fi/projects/ica/fastica/.

[M.Takeuchi,1994] M.Takeuchi (1994). *Modern Spherical Functions*. Amer. Math. Soc.

[S.Helgason,1962] S.Helgason (1962). *Differential Geometry and Symmetric Spaces*. Academic Press.

[S.Helgason,1978] S.Helgason (1978). *Differential Geometry, Lie Groups and Symmetric Spaces*. New York: Academic Press.

[S.Helgason,1984] S.Helgason (1984). *Groups and Geometric Analysis*. Academic Press.

[T.Akuzawa & M.Wadati,1998] T.Akuzawa & M.Wadati (1998). Diffusions on symmetric spaces of type AIII and random matrix theories for rectangular matrices. *J.Phys.A*, *31*, 1713–1732.

appendix

A proof of (2.18)

For $B \in GL(N,F)$ and $1 \leq i,j \leq N$,

$$\begin{aligned}[][T(X\otimes Y)T\mathrm{cs}(B)]_{i+N(j-1)} &= [(X\otimes Y)T\mathrm{cs}(B)]_{j+N(i-1)} \\ &= X_{ip}Y_{jq}(B')_{qp} = (YB'X')_{ji}\ . \end{aligned} \tag{A.1}$$

On the other hand

$$[(Y\otimes X)\mathrm{cs}(B)]_{i+N(j-1)} \qquad = Y_{jp}X_{iq}B_{qp} = (YB'X')_{ji}\ . \tag{A.2}$$

This proves the statement since cs is bijective. $\square$