

Multiplicative Algorithm for Orthogonal Groups and Independent Component Analysis

Toshinao AKUZAWA*

Brain Science Institute
RIKEN

2-1 Hirosawa, Wako, Saitama 351-0198, Japan

October 30, 2013

Abstract

The multiplicative Newton-like method developed by the author *et al.* is extended to the situation where the dynamics is restricted to the orthogonal group. A general framework is constructed without specifying the cost function. Though the restriction to the orthogonal groups makes the problem somewhat complicated, an explicit expression for the amount of individual jumps is obtained. This algorithm is exactly second-order-convergent. The global instability inherent in the Newton method is remedied by a Levenberg-Marquardt-type variation. The method thus constructed can readily be applied to the independent component analysis. Its remarkable performance is illustrated by a numerical simulation.

1 Overview

Many optimization problems take the form, “Find an optimal matrix under the constraints (1).. (2).. *etc.*” Some of these can be considered as optimizations on Lie groups. For groups, the fundamental manipulation is a multiplication whereas an addition is unnatural. In consideration of this fact, we have constructed a multiplicative Newton-like algorithm for maximizing the

*akuzawa@brain.riken.go.jp

kurtosis (a good barometer for the independence) in [T.Akuzawa & N.Murata,1999]. There the dynamics takes place on the coset $GL(1, \mathbb{R})^N \backslash GL(N, \mathbb{R})$. We can apply the techniques developed in [T.Akuzawa & N.Murata,1999] to many other optimization problems. The coset structure $GL(1, \mathbb{R})^N \backslash GL(N, \mathbb{R})$ is, however, characteristic of the independent component analysis(ICA). It is understood by the fact that the independence is nothing to do with the scaling. The redundancy resulting from the invariance of the model under the componentwise scaling must be eliminated for a rigorous discussion and this redundancy corresponds to $GL(1, \mathbb{R})^N$.

Another way to eliminate this redundancy is the prewhitening. The prewhitening is a linear transformation of the observed data which maps the covariance matrix to the unit matrix. If we deal with prewhitened data, we can legitimately narrow the sweeping range to the orthogonal group. The aim of this letter is the construction of a multiplicative algorithm for the orthogonal groups.

The framework is as follows. N -dimensional prewhitened random variables $\{X_i | 1 \leq i \leq N\}$ are available and it is anticipated that their origins are some unknown mutually independent components $\{Y_i^* | 1 \leq i \leq N\}$. The goal of the ICA is the map $\{X_i\} \mapsto \{Y_i^*\}$. We restrict ourselves to the linear independent component analysis. There we want to find a linear transformation $C^* : X = (X_1, \dots, X_N)' \mapsto Y^* = (Y_1^*, \dots, Y_N^*)' = C^* X$ which minimizes some cost function that measures the independence. Since we are assuming that the data is already prewhitened, the covariance matrix of X is the $N \times N$ unit matrix. If we do not take into account errors in the prewhitening, the optimal point C^* must belong to $O(N)$.

Giving up the analytical solution, we consider a sequence,

$$C(0), C(1), C(2), C(3), \dots, \quad (1.1)$$

which converges to the optimal solution C^* . The sequence $\{C(t)\}$ is generated by the left-multiplication of another sequence of orthogonal matrices $\{D(t)\}$. Each $D(t)$ is specified by the coordinate $\Delta(t)$ which satisfies $D(t) = e^{\Delta(t)}$. We assume that $\Delta(t)$ is an $N \times N$ skew-symmetric matrix, which implies that $D(t)$ belongs to the identity component of $O(N)$. In practice the procedure is as follows. As an initial condition we set $C(0)$. For $t > 0$ ($t \in \mathbb{N}^+$), we introduce $\Delta(t)$ and denote $C(t)$ as $C(t+1) = e^{\Delta(t)} C(t)$. Under these settings, we determine $\Delta(t)$ by using the Newton method with respect to the matrix elements of $\Delta(t)$. That is, we evaluate the cost function at $C(t+1)$ by expanding it around $C(t)$ in terms of the elements of $\Delta(t)$ up to the second order. Then $\Delta(t)$ is chosen as the (unique) critical 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 we will give a complete description of a new multiplicative updating method for the orthogonal groups. This section is the main part of this letter. Since our formulation does not depend on the details of the cost function the method can be useful for many problems other than the ICA. The performance of our method including the second-order-convergence is discussed in Section 3. Section 4 is a survey of possible applications of our method. The algorithm constructed in Section 2 is considered as a pure-Newton method on the orthogonal groups. To achieve the global convergence, we must modify the method. This is

accomplished in Section 5. Section 5 also includes a numerical examination of the performance of our method. Section 6 is a summary.

2 Multiplicative updating on $O(N)$

We assume that the cost function F takes the form,

$$F(Y) = \sum_{i=1}^N E(f_i(Y_i)) , \quad (2.1)$$

where each $f_i : \mathbb{R} \rightarrow \mathbb{R}$ is an unspecified function. Through this letter we denote by $E(\cdot)$ the expectation. We will determine the concrete procedures after the Newton manner. First, we introduce maps, R and $\{U_i(1 \leq i \leq N)\}$'s, from N -dimensional dataset to $N \times N$ matrices by

$$[R(Y)]_{ki} = E \left(\frac{\partial f_i(Y_i)}{\partial Y_i} Y_k \right) \quad (2.2)$$

and

$$[U_i(Y)]_{kl} = U_{ikl}(Y) = E \left(\frac{\partial^2 f_i(Y_i)}{\partial Y_i^2} Y_k Y_l \right) . \quad (2.3)$$

The goal is the construction of a sequence $\{Y(t)\}$ of the estimates of the independent components, which converges to the optimal point Y^* . Within the framework of the linear analysis, we consider that this sequence is derived from another sequence $\{C(t)\}$ of the linear transformation by the relation $Y(t) = C(t)X$, where X are the original data. Thus if we restate the problem, the task is to determine a sequence $\{C(t)\}$. We assume that for each $t \in \mathbb{N}^+$ the estimates of the independent components at time t and the estimates at time $t + 1$ are related by

$$Y(t+1) = D(t)Y(t) \quad (2.4)$$

or equivalently

$$C(t+1) = D(t)C(t) , \quad (2.5)$$

where $D(t)$ is some orthogonal matrix to be fixed. Our method is characterized by this left-multiplicative updating rule. As mentioned in the previous section, we assume that each $D(t)$ always belongs to the identity component of the orthogonal group $O(N)$. This assumption is reasonable, for example, if the original data X are already prewhitened in the case of the ICA. Anyway, under this restriction $D(t)$ is specified by an $N \times N$ anti-symmetric matrix $\Delta(t)$, which satisfies

$$\exp(\Delta(t)) = D(t) . \quad (2.6)$$

For brevity's sake we will omit the argument (t) and denote $Y(t+1)$ by Z . $F(Z)$ is expanded in terms of $\{\Delta_{ij}\}$ as

$$F(Z) = F(Y) + \text{tr}(\Delta R(Y)) + \text{tr}\left(\frac{\Delta^2}{2}R(Y)\right) + \frac{1}{2}\sum_{i,k,l}\Delta_{ik}\Delta_{il}U_{ikl}(Y) + O(\Delta^3). \quad (2.7)$$

Through the letter we denote by $O(\Delta^k)$ polynomials of matrix elements of Δ which does not contain terms with degrees less than k . Do not confuse this with the symbol for the orthogonal groups such as $O(N)$. As in the usual Newton method, we truncate the expansion (2.7) at the second order with respect to $\{\Delta_{ij}\}$. Then Δ in this step is determined as the coordinate of the critical point of this truncated expansion. The partial derivative of (2.7) is more convenient for the purpose. It reads

$$\frac{\partial F(Z)}{\partial \Delta_{kl}} = R_{lk} + \frac{1}{2}[\Delta R + R\Delta]_{lk} + \sum_p \Delta_{kp}U_{klp} + O(\Delta^2), \quad (2.8)$$

where we have omitted the argument Y for R and U . Now let us introduce a map cs (the column string) as in the previous article [T.Akuzawa & N.Murata,1999]:

$$\begin{aligned} \text{Mat}(N, \mathbb{F}) &\rightarrow \mathbb{F}^{N^2} \\ A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & \dots\dots\dots \\ \dots\dots\dots \\ A_{N1} & \dots\dots & A_{NN} \end{pmatrix} &\mapsto \text{cs}(A) = (A_{11} \ A_{21} \ \cdots \ A_{N1} \ A_{12} \ A_{22} \ \cdots \ A_{NN})', \end{aligned} \quad (2.9)$$

where $\text{Mat}(N, \mathbb{F})$ is $N \times N$ matrices on some unspecified field \mathbb{F} . We denote by the upper subscript $'$ the transposition and by \dagger the complex conjugate. For the orthogonal groups it is rather simple to move to the framework of the column string as compared to the case of $GL(1, \mathbb{R})^N \setminus GL(N, \mathbb{R})$: By neglecting $O(\Delta^2)$ terms, the right-hand-side of (2.8) is straightforwardly rewritten as

$$\begin{aligned} R_{lk} + \frac{1}{2}[\Delta R + R\Delta]_{lk} + \sum_p \Delta_{kp}U_{klp} \\ = \left[\text{cs}(R) + \frac{1}{2}(R' \otimes I_N + I_N \otimes R) \text{cs}(\Delta) + \left(\bigoplus_k U_k\right) T \text{cs}(\Delta) \right]_{l+(k-1)N}, \end{aligned} \quad (2.10)$$

where the symbol " \bigoplus " stands for the direct sum,

$$\bigoplus_{k=1}^N U_k = \begin{pmatrix} U_1 & 0 & \dots\dots & 0 \\ 0 & U_2 & 0 & \dots\dots \\ \dots\dots\dots \\ 0 & \dots\dots & U_{N-1} & 0 \\ 0 & 0 & \dots\dots & U_N \end{pmatrix}, \quad (2.11)$$

T is an $N^2 \times N^2$ matrix defined by

$$\text{cs}(A') = T \text{cs}(A) \text{ for } A \in \text{Mat}(N, \mathbb{F}) , \quad (2.12)$$

and I_N is the $N \times N$ unit matrix. We denote the tensor product by \otimes as usual. The “transposition” T is also considered as an intertwiner between two equivalent representations:

$$T(A \otimes B)T = B \otimes A . \quad (2.13)$$

The orthogonal group $O(N)$ has less degrees of freedom than the general linear group. The canonical basis of the Lie algebra, $\mathfrak{o}(N)$, of $O(N)$ is $N(N-1)/2$ anti-symmetric matrices. We will introduce some operators which enable us to move to the coordinates based on the canonical basis on $\mathfrak{o}(N)$. In the first place, we introduce an $N^2 \times N^2$ matrix H by

$$H = \sum_{i>j} H^{(i,j)} , \quad (2.14)$$

where $H^{(i,j)}$ is a $\pi/4$ rotation between the $j + N(i-1)$ -th component and the $i + N(j-1)$ -th component:

$$H_{kl}^{(i,j)} = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } k = j + N(i-1), \quad l = j + N(i-1) \\ -\frac{1}{\sqrt{2}} & \text{for } k = j + N(i-1), \quad l = i + N(j-1) \\ \frac{1}{\sqrt{2}} & \text{for } k = i + N(j-1), \quad l = j + N(i-1) \\ \frac{1}{\sqrt{2}} & \text{for } k = i + N(j-1), \quad l = i + N(j-1) \\ 0 & \text{otherwise.} \end{cases} \quad (2.15)$$

The projection operator P_D ,

$$P_D = \text{diag}(p_1, \dots, p_{N^2}) , \quad \begin{cases} p_k = 1 & \text{for } k = N(i-1) + i, 1 \leq i \leq N \\ p_k = 0 & \text{otherwise ,} \end{cases} \quad (2.16)$$

is used to extract the diagonal elements of a matrix from its image by cs . Then the coordinate transformation is realized by a multiplication of

$$H + P_D \quad (2.17)$$

to column string vectors. We need to introduce two more projection operators P_S and P_A defined by

$$P_S = \text{diag}(p_1, p_2, \dots, p_{N^2}) \quad (2.18)$$

$$P_A = \text{diag}(1 - p_1, 1 - p_2, \dots, 1 - p_{N^2}) , \quad (2.19)$$

where

$$p_k = \begin{cases} 1 & \text{if } \exists(i, j); \ j \leq i \text{ and } k = i + N(j - 1) \\ 0 & \text{otherwise.} \end{cases} \quad (2.20)$$

By the left-action of P_S and P_A to column string vectors rotated by $H + P_D$ we can extract, respectively, the symmetric components and the anti-symmetric components of the matrices. Then the conditions for the critical point of the second-order-expansion, which must be satisfied by Δ , are translated into the following two conditions. First, symmetric components of Δ must vanish. This condition is expressed as

$$[(H + P_D)\text{cs}(\Delta)]_{j+(i-1)N} = 0 \quad \text{for } i \leq j \quad \left(\Longleftrightarrow P_S(H + P_D)\text{cs}(\Delta) = 0 \right). \quad (2.21)$$

Secondly, for the anti-symmetric components the condition for the critical point is transformed to

$$[(H + P_D)\text{cs}(R) + (H + P_D)W\text{cs}(\Delta)]_{j+(i-1)N} = 0 \quad \text{for } i > j, \quad (2.22)$$

where we have set

$$W = \frac{1}{2} (R' \otimes I_N + I_N \otimes R) + \left(\bigoplus_k U_k \right) T. \quad (2.23)$$

The conditions (2.21) and (2.22) are combined into an equation,

$$P_A(H + P_D)\text{cs}(R) + \left[P_A(H + P_D)W(H + P_D)'P_A + P_S \right] (H + P_D)\text{cs}(\Delta) = 0. \quad (2.24)$$

Note that

$$P_A(H + P_D) = P_A H. \quad (2.25)$$

The optimal Δ is immediately obtained from (2.24):

$$\begin{aligned} \text{cs}(\Delta) &= -(H + P_D)' \left[P_A(H + P_D)W(H + P_D)'P_A + P_S \right]^{-1} P_A(H + P_D)\text{cs}(R) \\ &= -H' (P_A H W H' P_A + P_S)^{-1} P_A H \text{cs}(R). \end{aligned} \quad (2.26)$$

Thus we have obtained the explicit updating rule. By iterating the procedure in this section from a starting point sufficiently close to the optimal one, the sequences $\{C(t)\}$ and $\{Y(t)\}$ converge to the optimal solutions.

3 Performance (theoretical aspects)

The second-order-convergence is one of the main advantages of this method. Indeed, this algorithm is rigorously second-order-convergent. The proof can be given almost in the same way as in [T.Akuzawa & N.Murata,1999]. So we omit the proof in this letter.

Sometimes we have to deal with large matrices to apply the technique here constructed. Let us examine the situation. The $N^2 \times N^2$ matrix $P_A H W H' P_A + P_S$ is a direct sum of an $N(N-1)/2 \times N(N-1)/2$ matrix and an $N(N+1)/2 \times N(N+1)/2$ unit matrix. Within the $N(N-1)/2 \times N(N-1)/2$ block the number of non-zero off-diagonal elements is no more than $N(N-1)(N-2)$. So this is a very sparse matrix when N becomes large. Of course if N becomes extremely large, our method requires quite large memories. But due to the sparseness, it remains to be a practical tool for problems with considerably large N .

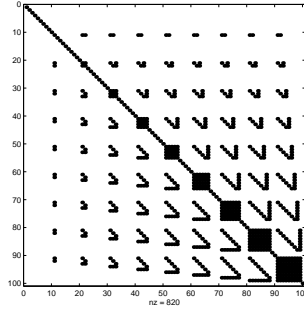


Figure 1: $N = 10$. The black dots denote non-zero elements of $P_A H W H' P_A + P_S$.

As is often the case with the Newton method, the global convergence is not assured by this algorithm. Fortunately it is possible to cure this fault. We will show the prescription to the global instability in Section 5.

4 Applications to ICA

So far we have not specified the cost function beyond the assumption that the cost function is a sum of the form (2.1). Many of the cost functions for the independent component analysis belong to this class.

4.1 Kullback-Leibler information

The Kullback-Leibler information,

$$\int \prod_{i=1}^N dy_i P(y) \left\{ \ln P(y) - \sum_{i=1}^N \ln P_i(y_i) \right\} , \quad (4.1)$$

is a good measure for the independence. Here P is the joint probability density function of $\{Y_i\}$ and P_i is the probability density function of the i -th component. We have already restricted ourselves to the case where the jacobian of the transformation equals one. Then the minimization of the Kullback-Leibler information is equivalent to the minimization of

$$- \int \prod_i dY_i P(Y) \sum_{i=1}^N \ln P_i(Y_i) = \sum_{i=1}^N E(-\ln P_i(Y_i)) . \quad (4.2)$$

Thus we can legitimately transform the Kullback-Leibler information to a cost function of the form (2.1), where we should set $\{f_i\}$'s as

$$f_i(\cdot) = -\ln P_i(\cdot) . \quad (4.3)$$

We must evaluate $\{P_i\}$'s, their derivatives, and so on to determine the optimal solution. A robust estimation of these quantities is possibly not an easy task[B.W.Silverman,1986, D.Cox,1985].

4.2 Cumulant of fourth order

The kurtosis of a random variable A is defined by

$$\kappa(A) = \frac{E(A^4)}{(E(A^2))^2} - 3 . \quad (4.4)$$

The kurtosis is related to the cumulant of the fourth order,

$$Cum^{(4)}(A) = E(A^4) - 3(E(A^2))^2 , \quad (4.5)$$

by

$$\kappa(A) = \frac{Cum^{(4)}(A)}{(E(A^2))^2} . \quad (4.6)$$

For prewhitened data the kurtosis equals the cumulant of the fourth order. As is well-known[A.Hyvärinen,1997, T.Akuzawa & N.Murata,1999], we can grab independent components in many cases by seeking the maximum of the absolute values of the kurtoses. Our method is applicable by setting

$$f_i = -\kappa^2 \quad (4.7)$$

for all i . If it is known a priori that all the sources $\{Y_i^*\}$ have positive kurtoses, we may use the kurtosis itself and set

$$f_i = -\kappa . \quad (4.8)$$

For these cost functions, R , $\{U_i\}$, and other quantities needed for determining each step are calculated easily from the observed data. Thus applying our method for this cost function is highly practical and reasonable choice.

5 Levenberg-Marquardt-type variation and performance in practice

The pure-Newton updating rule (2.26) has a poor global convergence property. This drawback is remedied by the Levenberg-Marquardt-type variation[W.H.Press *et al.*,1988]. First, We modify (2.26) as

$$\text{cs}(\Delta) = -H' (P_A H W H' P_A + P_S + \lambda I_{N^2})^{-1} P_A H \text{cs}(R) . \quad (5.1)$$

The initial value λ_0 for λ is fixed at some positive value. We also fix a real number $\alpha(> 1)$. (In the following example we set $\lambda_0 = 50$ and $\alpha = 10$.) Then the procedure at time t is as follows:

- i) Calculate Δ by (5.1).
- ii) If $F(e^{\Delta} Y(t))$ is larger than $F(Y(t))$, multiply λ by α and go back to i).
- iii) Otherwise, multiply λ by $1/\alpha$ and proceed to the next time step $t + 1$.

Other parts of the algorithm is completely the same as in the pure-Newton version in Section 2.

Let us examine the real performance of our method under this setting. For the cost function we choose the kurtosis as in Subsection 4.2. The source signals are three synthesizer-generated wav files(Fig.2). Pseudo-observed data are generated by mixing the source by a random matrix,

$$A = I_3 + S, \quad (5.2)$$

where each element of S is distributed uniformly on $(-1/2, 1/2)$. The residual crosstalk of the signals demixed by our method is 1.29% on average. It takes about 122 seconds (CPU time) for one hundred iteration of the same problem on our workstation. For reference, we have also solved the same demixing problem by the FastICA[Hurri *et al.*,1998]. In this case the residual crosstalk is 1.36% on average and it takes about 156 seconds for one hundred iteration on the same workstation. Since the author's knowledge about the FastICA package is limited, one should not take this result seriously. It can, however, be said that our method is quite good also in practice.

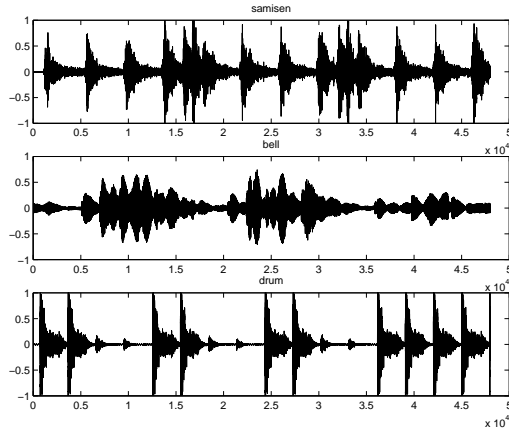


Figure 2: Sample data generated by a synthesizer (by courtesy of N.Murata).

6 Summary

We have constructed a new algorithm for finding a critical point of broad classes of cost functions on the orthogonal groups. This method is second-order-convergent since it is in essence the Newton method. The method here constructed is an extension (or a restriction) of the multiplicative updating method developed in our previous work[T.Akuzawa & N.Murata,1999]. The constraint for Δ from the nature of the orthogonal groups makes the problem a little complicated. We have, however, obtained a rigorous and explicit updating rule. We have also constructed a Levenberg-Marquardt-type variation, which is suitable for practical purpose. The global instability inherent in the Newton method is remedied in this version. Since our discussion does not depend on the detail of the cost function, this method is applicable to many concrete problems. The relatively mild assumption (2.1) on the form of the cost function, however, implies that our algorithm is especially suitable for the ICA. Its practical utility for the ICA have been illustrated here by a numerical simulation.

To summarize, our algorithm has numerous theoretical virtues such as the rigorous second order convergence, the explicit and strict formulation, and so on. It provides, also in practice, fast and powerful tools for the ICA and many other problems.

Acknowledgments

The author would like to thank Noboru Murata and Shun-ichi Amari for valuable discussions and comments.

References

- [A.Hyvärinen,1997] A.Hyvärinen (1997). A Fast Fixed-Point Algorithm for Independent Component Analysis. *Neural Computation*, 9, 1483–1492.
- [B.W.Silverman,1986] B.W.Silverman (1986). *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall.
- [D.Cox,1985] D.Cox, D. (1985). A Penalty Method for Nonparametric Estimation of the Logarithmic Derivative of a Density Function. *Ann.Inst.Statist.Math.*, 37, 271–288.
- [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/>.
- [T.Akuzawa & N.Murata,1999] T.Akuzawa & N.Murata (1999). Multiplicative Nonholonomic/Newton-like Algorithm. *preprint*
(available from <http://www.islab.brain.riken.go.jp/~akuzawa/>).
- [W.H.Press *et al.*,1988] W.H.Press, B.P.Flannery, S.A.Teukolsky, & W.T.Vetterling (1988). *Numerical Recipes in C*. Cambridge: Cambridge U.P.