JSIAM Letters Vol.7 (2015) pp.13–16 ©2015 Japan Society for Industrial and Applied Mathematics

Joint singular value decomposition algorithm based on the Riemannian trust-region method

Hiroyuki Sato[1]

[1] Department of Management Science, Tokyo University of Science, 1-3 Kagurazaka, Shinjuku-ku, Tokyo 162-8601, Japan

E-mail *hsato@ms.kagu.tus.ac.jp*

Received October 22, 2014, Accepted November 27, 2014

Abstract

The joint singular value decomposition of multiple rectangular matrices is formulated as a Riemannian optimization problem on the product of two Stiefel manifolds. In this paper, the geometry of the objective function and the Riemannian manifold for this problem are studied to develop a Riemannian trust-region algorithm. The proposed algorithm globally and locally quadratically converges, and our numerical experiments demonstrate that it performs much better than the steepest descent method.

Keywords joint singular value decomposition, Riemannian optimization, trust-region method

Research Activity Group Algorithms for Matrix / Eigenvalue Problems and their Applications

1. Introduction

In the Tomasi-Kanade factorization method, scene geometry and camera motion are recovered from a sequence of camera images using the singular value decomposition (SVD) of a measurement matrix [1]. In [2], the problem was extended to several cameras installed in the same direction. Several measurement matrices were obtained from the cameras, and the SVD of the Tomasi-Kanade factorization was generalized to the joint singular value decomposition (JSVD) of these measurement matrices. However, the matrices do not generally have a common set of singular vectors, so they cannot be exactly decomposed into SVD forms by a common pair of orthogonal matrices. Thus, we must define the approximate JSVD problem as an optimization problem. Some other applications to blind source separation were also discussed in [3]. Furthermore, the JSVD is related to the Karhunen-Loève transform, which is a useful tool for image compression [4]. As such, JSVD is an important technique that is worth studying.

The SVD of a single matrix can also be written as an optimization problem. This problem contains orthonormality constraints on the variable matrices. Note that the Stiefel manifold $\mathrm{St}(p,n)$ with $p \le n$ is defined to be the set of all $n \times p$ orthonormal matrices. In [5], the problem was rewritten as an optimization problem on the product of two Stiefel manifolds, and several algorithms were developed. Optimization techniques on Riemannian manifolds are called Riemannian optimization. They have recently been intensively researched [6], and many general efficient algorithms have been developed.

In this paper, we consider the JSVD problem as a Riemannian optimization problem on the product of two Stiefel manifolds and develop the trust-region method for the problem.

2. Joint singular value decomposition

Let $A_1, A_2, \ldots, A_K$ be $m \times n$ real matrices with $m \ge n$. We assume that $p \le n$ throughout this paper. We consider the following JSVD problem as a Riemannian optimization problem.

Problem 1

$$\text{minimize } f(U,V) = -\sum_{l=1}^{K} \|\mathrm{diag}\left(U^T A_l V\right)\|_F^2,$$

$$\text{subject to } (U,V) \in \mathrm{St}(p,m) \times \mathrm{St}(p,n),$$

where $\|\cdot\|_F$ *denotes the Frobenius norm of the matrix and the Stiefel manifold* $\mathrm{St}(p,n)$ *is defined as*

$$\mathrm{St}(p,n) := \{Y \in \mathbb{R}^{n\times p} | Y^T Y = I_p\},$$

and where f *is defined on* $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$*, not in the whole* $\mathbb{R}^{m\times p} \times \mathbb{R}^{n\times p}$*.*

To derive optimization algorithms for Problem 1, we must study the geometry of the problem. We first review the geometry of $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$. See [5] for more details.

The tangent space of $\mathrm{St}(p,m) \times \mathrm{St}(p,n)$ at (U,V) is

$$\begin{aligned} &T_{(U,V)}(\mathrm{St}(p,m) \times \mathrm{St}(p,n)) \\ &\quad = \{(\xi,\eta) \in \mathbb{R}^{m\times p} \times \mathbb{R}^{n\times p} | \\ &\qquad \xi^T U + U^T \xi = \eta^T V + V^T \eta = 0\}. \end{aligned}$$

We can endow $\mathrm{St}(p,n)$ with the Riemannian metric

$$\langle \xi_1, \xi_2 \rangle_Y = \mathrm{tr}(\xi_1^T \xi_2), \quad \xi_1, \xi_2 \in T_Y \mathrm{St}(p,n), \tag{1}$$

where $T_Y \mathrm{St}(p,n)$ is the tangent space of $\mathrm{St}(p,n)$ at Y. The Riemannian metric (1) is induced from the natural inner product

$$\langle B, C \rangle = \mathrm{tr}(B^T C), \qquad B, C \in \mathbb{R}^{n\times p},$$

in the Euclidean space $\mathbb{R}^{n\times p}$. Using the induced metric (1), we can view $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ as a Riemannian manifold with the Riemannian metric

$$\langle(\xi_1,\eta_1),(\xi_2,\eta_2)\rangle_{(U,V)} = \mathrm{tr}(\xi_1^T\xi_2)+\mathrm{tr}(\eta_1^T\eta_2),$$
$$(\xi_1,\eta_1),(\xi_2,\eta_2)\in T_{(U,V)}(\mathrm{St}(p,m)\times\mathrm{St}(p,n)). \quad (2)$$

Under the Riemannian metric (2), the orthogonal projection map onto the tangent space at (U,V) acts on $(B,C)\in\mathbb{R}^{m\times p}\times\mathbb{R}^{n\times p}$ as

$$P_{(U,V)}(B,C) = \left(B-U\,\mathrm{sym}\left(U^TB\right), C-V\,\mathrm{sym}\left(V^TC\right)\right), \quad (3)$$

where $\mathrm{sym}(\cdot)$ denotes the symmetric part of the matrix in the parentheses; that is, $\mathrm{sym}(A)=\left(A+A^T\right)/2$.

We proceed to the geometry of the objective function f. Let $\bar{f}$ be a function with the same form as f defined in $\mathbb{R}^{m\times p}\times\mathbb{R}^{n\times p}$; that is,

$$\bar{f}(U,V) = -\sum_{l=1}^{K}\|\mathrm{diag}\left(U^TA_lV\right)\|_F^2,$$
$$(U,V)\in\mathbb{R}^{m\times p}\times\mathbb{R}^{n\times p}. \quad (4)$$

Note that (4) can be rewritten as

$$\bar{f}(U,V) = -\sum_{l=1}^{K}\mathrm{tr}\left(U^TA_lV\,\mathrm{diag}\left(U^TA_lV\right)\right).$$

The Euclidean gradient $\nabla\bar{f}$ and the Hessian $\nabla^2\bar{f}$ can then be computed as

$$\nabla\bar{f}(U,V) = -2\sum_{l=1}^{K}\left(A_lVD_l, A_l^TUD_l\right),$$
$$\nabla^2\bar{f}(U,V)[(\xi,\eta)] = -2\sum_{l=1}^{K}\left(A_l(\eta D_l+VD_l'), A_l^T(\xi D_l+UD_l')\right),$$

where $D_l := \mathrm{diag}(U^TA_lV)$, $D_l' := \mathrm{diag}(\xi^TA_lV + U^TA_l\eta)$, and $(\xi,\eta)\in\mathbb{R}^{m\times p}\times\mathbb{R}^{n\times p}$.

The gradient $\mathrm{grad}\,f$ of f on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ can be calculated by the projection map (3) on the Euclidean gradient $\nabla\bar{f}$. That is,

$$\mathrm{grad}\,f(U,V) = P_{(U,V)}(\nabla\bar{f}(U,V)) = -2\sum_{l=1}^{K}\left(A_lVD_l - U\,\mathrm{sym}(S_lD_l), A_l^TUD_l - V\,\mathrm{sym}(D_lS_l)\right), \quad (5)$$

where we have defined $S_l := U^TA_lV$.

The Hessian $\mathrm{Hess}\,f$ of f on $\mathrm{St}(p,m)\times\mathrm{St}(p,n)$ can also be expressed by using $\nabla^2\bar{f}$ and the projection map (3). That is,

$$\mathrm{Hess}\,f(U,V)[(\xi,\eta)] = P_{(U,V)}(\mathrm{D}(\mathrm{grad}\,f)(U,V)[(\xi,\eta)]). \quad (6)$$

Note that $\mathrm{grad}\,f(U,V) = P_{(U,V)}(\nabla\bar{f}(U,V))$. We regard the right-hand side of this relation as a product of two matrix functions of U and V, rather than a composition of a map and a function. Then, (6) can be written as

$$\begin{aligned}\mathrm{Hess}\,f(U,V)[(\xi,\eta)] &= P_{(U,V)}(\mathrm{D}(\mathrm{grad}\,f)(U,V)[(\xi,\eta)])\\ &= P_{(U,V)}(\mathrm{D}(P_{(U,V)}(\nabla\bar{f}))(U,V)[(\xi,\eta)])\\ &= P_{(U,V)}(\mathrm{D}P_{(U,V)}[(\xi,\eta)](\nabla\bar{f}(U,V)) + P_{(U,V)}(\mathrm{D}(\nabla\bar{f})(U,V)[(\xi,\eta)]))\\ &= P_{(U,V)}(\mathrm{D}P_{(U,V)}[(\xi,\eta)](\nabla\bar{f}(U,V)) + \nabla^2\bar{f}(U,V)[(\xi,\eta)]),\end{aligned} \quad (7)$$

where

$$\mathrm{D}P_{(U,V)}[(\xi,\eta)](B,C) = \left(-\xi\,\mathrm{sym}(U^TB) - U\,\mathrm{sym}(\xi^TB), -\eta\,\mathrm{sym}(V^TC) - V\,\mathrm{sym}(\eta^TC)\right). \quad (8)$$

We have used the relation $P_{(U,V)}^2 = P_{(U,V)}$ in the last equality of (7). Furthermore, if S_1 and S_2 are $p\times p$ symmetric matrices, we have

$$P_{(U,V)}(US_1,VS_2) = \left(US_1 - U\,\mathrm{sym}(U^TUS_1), VS_2 - V\,\mathrm{sym}(V^TVS_2)\right) = 0. \quad (9)$$

This relation can reduce the computational cost of $P_{(U,V)}(\mathrm{D}P_{(U,V)}[(\xi,\eta)](\nabla\bar{f}(U,V)))$ in (7). Indeed, it follows from (8) and (9) that

$$P_{(U,V)}(\mathrm{D}P_{(U,V)}[(\xi,\eta)](B,C)) = -P_{(U,V)}\left(\xi\,\mathrm{sym}(U^TB), \eta\,\mathrm{sym}(V^TC)\right), \quad (10)$$

for any $(B,C)\in\mathbb{R}^{m\times p}\times\mathbb{R}^{n\times p}$. Consequently, we can compute $\mathrm{Hess}\,f(U,V)[(\xi,\eta)]$ using (7) with (10).

3. Trust-region method for Problem 1

In this section, for simplicity, we let $M := \mathrm{St}(p,m)\times\mathrm{St}(p,n)$, $x := (U,V)\in M$, $\zeta := (\xi,\eta)\in T_xM$.

In [2], Problem 1 with $p=n$ is solved using a gradient flow of the objective function f. That is, they solved a differential equation

$$\dot{x} = -\,\mathrm{grad}\,f(x),$$

which is in fact a pair of equations

$$\begin{cases}\dot{U} = 2\sum_{l=1}^{K}(A_lVD_l - U\,\mathrm{sym}(S_lD_l)),\\ \dot{V} = 2\sum_{l=1}^{K}(A_l^TUD_l - V\,\mathrm{sym}(D_lS_l)).\end{cases}$$

Note that our resultant $\mathrm{grad}\,f$ (5) is more general than that in [2], because we assume that the integer $p\le n$ is arbitrary, whereas $p=n$ in [2].

In Riemannian optimization methods, the steepest descent method with a geodesic on M corresponds to solving a differential equation so that it follows a negative gradient flow. In the Riemannian steepest descent method, we often use a more efficient curve than a geodesic. In other words, we use a better map than the exponential map. Such a map is called a retraction [6]. It

is a map from the tangent bundle TM to M and defines an appropriate curve for the line search.

A reasonable choice for a retraction R^{St} on the Stiefel manifold $\mathrm{St}(p,n)$ is

$$R_Y^{\mathrm{St}}(\xi) = \mathrm{qf}(Y+\xi), \quad Y \in \mathrm{St}(p,n), \quad \xi \in T_Y\mathrm{St}(p,n),$$

where $\mathrm{qf}(\cdot)$ denotes the Q factor of the QR decomposition of the matrix in parentheses. If a full-rank matrix B has the QR decomposition $B = QR$, then $\mathrm{qf}(B) = Q$. A useful retraction R on $M = \mathrm{St}(p,m) \times \mathrm{St}(p,n)$ can then be defined by

$$R_{(U,V)}(\xi,\eta) = (\mathrm{qf}(U+\xi), \mathrm{qf}(V+\eta)), \quad (\xi,\eta) \in T_{(U,V)}M. \tag{11}$$

R on M then satisfies the following properties which appear in the definition of a general retraction.

(1) $R_x(0_x) = x$, where 0_x is the zero element of T_xM.

(2) With the identification $T_{0_x}T_xM \simeq T_xM$, we have $\mathrm{D}R_x(0_x) = \mathrm{id}_{T_xM}$, where $\mathrm{D}R_x(0_x)$ denotes the derivative of R_x at 0_x, and id_{T_xM} is the identity map on T_xM.

In the steepest descent method, we construct a sequence using the update formula

$$x_{k+1} = R_{x_k}(-\alpha_k \operatorname{grad} f(x_k)),$$

where $\alpha_k > 0$ is a step size satisfying some conditions (such as the Armijo or Wolfe conditions [6]). The steepest descent method globally converges, but very slowly, whereas the trust-region method generates a sequence that converges much faster. The results of our numerical experiments are shown in Section 4.

In the Riemannian trust-region method [6], we construct a quadratic model of the objective function within a so-called trust-region. A trust-region with a radius $\Delta > 0$ at $x \in M$ is defined as a ball in T_xM such that $\{\zeta \in T_xM | \|\zeta\|_x \le \Delta\}$. In this trust-region, we approximate the original objective function f using the Taylor expansion. That is,

$$\hat{m}_x(\zeta) = f(x) + \langle \operatorname{grad} f(x), \zeta\rangle_x + \frac{1}{2}\langle \operatorname{Hess} f(x)[\zeta], \zeta\rangle_x. \tag{12}$$

The trust-region sub-problem at $x \in M$ with a radius Δ is then formulated as to minimize $\hat{m}_x(\zeta)$ subject to $\zeta \in T_xM, \|\zeta\|_x \le \Delta$. After obtaining the solution ζ_* to this sub-problem, we compare the decrease in the objective function f and the model $\hat{m}_x$ attained by ζ_*, to evaluate whether ζ_* should be accepted, and whether the trust-region with the radius Δ is appropriate. This process is summarized in Algorithm 1.

In Algorithm 1, we can use the retraction R defined by (11). $\hat{m}_{x_k}$ is defined by (12), which can be computed by using the formulas for the gradient and Hessian of the objective function given in (5) and (7).

The trust-region sub-problem (Step 3) can be solved by means of the truncated conjugate gradient (CG) algorithm [6]. Because the trust-region sub-problem is the Euclidean optimization problem if we regard T_xM as Euclidean space, we can apply the standard linear CG algorithm. We must note that at iteration j of the inner truncated CG algorithm, if the updated vector ζ^{j+1} does not satisfy $\|\zeta^{j+1}\| \le \Delta$, we adjust the step length so that $\|\zeta^{j+1}\| = \Delta$.

Algorithm 1 Trust-Region Method for Problem 1

1: Choose parameters $\bar{\Delta} > 0, \Delta_0 \in (0,\bar{\Delta}), \rho' \in [0,\frac{1}{4})$, and an initial point $x_0 \in M$.
2: **for** $k = 0,1,2,\ldots$ **do**
3: Calculate $\zeta_k = \arg\min_{\zeta \in T_{x_k}M} \left\{\hat{m}_{x_k}(\zeta) | \langle\zeta,\zeta\rangle_{x_k} \le \Delta_k^2\right\}$.
4: Evaluate $\rho_k := \frac{f(R_{x_k}(0)) - f(R_{x_k}(\zeta_k))}{\hat{m}_{x_k}(0) - \hat{m}_{x_k}(\zeta_k)}$.
5: **if** $\rho_k < 1/4$ **then**
6: $\Delta_{k+1} = (1/4)\Delta_k$.
7: **else if** $\rho_k > 3/4$ and $\|\zeta_k\|_{x_k} = \Delta_k$ **then**
8: $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$.
9: **else**
10: $\Delta_{k+1} = \Delta_k$.
11: **end if**
12: **if** $\rho_k > \rho'$ **then**
13: $x_{k+1} = R_{x_k}(\zeta_k)$.
14: **else**
15: $x_{k+1} = x_k$.
16: **end if**
17: **end for**

In general, it can be shown that if the objective function is smooth and the Riemannian manifold is compact, then the Riemannian trust-region method with the inner truncated CG and appropriate parameters is globally and locally quadratically convergent [6]. Because our problem is for a smooth objective function on a compact Riemannian manifold, the proposed algorithm generates a globally and quadratically convergent sequence.

4. Numerical results

We implemented the proposed algorithm using Manopt, which is a MATLAB toolbox for optimization on manifolds [7], and compared the algorithm with several other methods.

4.1 Exact JSVD

In the first experiment, we considered the case in which the target matrices have an exact JSVD form. Let $m = 5, n = p = 3$, and $K = 2$. We prepared 5×3 matrices A_1 and A_2 in a similar manner to [8]. That is,

$$A_1 = U_{\mathrm{rand}} \operatorname{diag}(1,2,3) V_{\mathrm{rand}}^T,$$

$$A_2 = U_{\mathrm{rand}} \operatorname{diag}(3,2,1) V_{\mathrm{rand}}^T,$$

where $U_{\mathrm{rand}} \in \mathrm{St}(3,5)$ and $V_{\mathrm{rand}} \in O(3)(= \mathrm{St}(3,3))$ are randomly chosen matrices, and $O(n)$ denotes the orthogonal group of order n. We randomly chose an initial guess $(U_0, V_0) \in \mathrm{St}(3,5) \times O(3)$, and fixed it throughout this subsection. As mentioned in the previous section, the existing algorithm by Hori [2,8] corresponds to the steepest descent method. We compared our proposed trust-region method with the steepest descent and also (non-linear) conjugate gradient methods.

It is clear from Fig. 1 that the trust-region method (Algorithm 1) generated a sequence that quickly converged to an optimal solution, as expected.

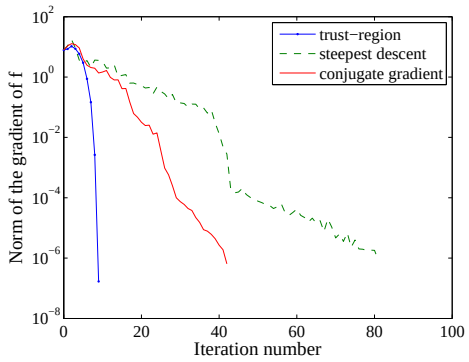

Fig. 1. Comparison between the trust-region, steepest descent, and non-linear conjugate gradient methods. The horizontal axis represents the iteration number k and the vertical axis represents the corresponding norm ($\|\mathrm{grad}\, f(x_k)\|_{x_k}$).

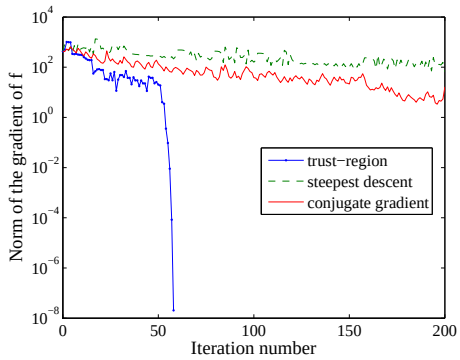

Fig. 2. Comparison between the three methods with $K = 100$ target matrices.

4.2 Approximate JSVD

We then ran some experiments for a more challenging and general case. In the context of recovering scene geometry, there are an $m \times n$ matrix A that correctly describes the situation and measurement matrices $A_1, A_2, \ldots, A_K$, where

$$A_l = A + N_l, \quad l = 1, 2, \ldots, K, \tag{13}$$

and N_l, $l = 1, 2, \ldots, K$ are noise matrices.

In our numerical experiment, we set $K = 100, m = 100, n = 50$, and $p = 50$, and used (13) to construct target matrices $A_1, A_2, \ldots, A_{100}$, where each element of N_l is an independent standard normal random variable.

In this case, we obtained a simple and reasonable initial guess for Problem 1 as follows. We first computed the average $\bar{A} = \sum_{l=1}^{K} A_l/K$ of $A_1, A_2, \ldots, A_K$. Then, we computed the standard SVD for the single matrix $\bar{A}$ to obtain $\bar{A} = \bar{U}\bar{\Sigma}\bar{V}^T$, and let U_0 and V_0 be the matrices with the leftmost p columns of $\bar{U}$ and $\bar{V}$, respectively. With (U_0, V_0) as an initial guess, we performed numerical experiments similar to those in the previous subsection.

The numerical results in Fig. 2 show that our algorithm converges quadratically. The computational time taken for 58 iterations of the trust-region method to obtain the final point of the graph was 11.12 seconds. For the achieved point $(U_{\mathrm{tr}}, V_{\mathrm{tr}})$, the norm of the gradient was $\|\mathrm{grad}\, f(U_{\mathrm{tr}}, V_{\mathrm{tr}})\|_{(U_{\mathrm{tr}}, V_{\mathrm{tr}})} = 2.047 \times 10^{-8}$. The point $(U_{\mathrm{st}}, V_{\mathrm{st}})$ obtained after 10000 iterations and 53.90 seconds of the steepest descent method satisfied $\|\mathrm{grad}\, f(U_{\mathrm{st}}, V_{\mathrm{st}})\|_{(U_{\mathrm{st}}, V_{\mathrm{st}})} = 0.7038$. On average, the computational times per iteration taken for the trust-region and the steepest descent methods are 0.1918 and 5.390×10^{-3} seconds, respectively. That is, when considering the computational time per iteration, the steepest descent method is faster than the trust-region method. However, since the convergence speed of the steepest descent method is much slower than that of the trust-region method, the trust-region method is much superior to the steepest descent method as a result.

Furthermore, the value of the objective function at the point $(U_{\mathrm{tr}}, V_{\mathrm{tr}})$ was $f(U_{\mathrm{tr}}, V_{\mathrm{tr}}) = -2.360 \times 10^4$, whereas $f(U_0, V_0) = -2.073 \times 10^4$ and $f(U_{\mathrm{st}}, V_{\mathrm{st}}) = -2.356 \times 10^4$. This implies that the proposed algorithm significantly decreased the value of the objective function.

5. Concluding remarks

We studied the geometry of the Riemannian optimization problem (Problem 1), which describes the JSVD for developing the Riemannian trust-region algorithm. Our numerical experiments showed that the proposed algorithm converges globally and quadratically, as indicated by the theory.

As stated in [8], optimization-based methods are expected to have an advantage because they can be flexibly used for on-line applications (i.e., for time-varying target matrices). This is because we can use the information on the optimal solution at time t to guess a reasonable initial point for the problem at the subsequent time t_+.

Acknowledgments

The author would like to thank the anonymous referee for providing him with valuable comments that helped him to brush up the paper. This work was supported by JSPS KAKENHI Grant Number 26887037.

References

[1] C. Tomasi and T. Kanade, Shape and motion from image streams under orthography: a factorization method, Int. J. Comput. Vision, **9** (1992), 137–154.

[2] G. Hori, Joint SVD and its application to factorization method, in: Proc. of 9th Int. Conf. Latent Variable Analysis and Signal Separation, pp. 563–570, 2010.

[3] M. Congedo and R. Phlypo, Approximate joint singular value decomposition of an asymmetric rectangular matrix set, IEEE Trans. Signal Process., **59** (2011), 415–424.

[4] B. Pesquet-Popescu, J.-C. Pesquet and A. P. Petropulu, Joint singular value decomposition–a new tool for separable representation of images, in: Proc. of 2001 IEEE Int. Conf. Image Processing, pp. 569–572, 2001.

[5] H. Sato and T. Iwai, A Riemannian optimization approach to the matrix singular value decomposition, SIAM J. Optim., **23** (2013), 188–212.

[6] P.-A. Absil, R. Mahony and R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, Princeton, 2008.

[7] N. Boumal, B. Mishra, P.-A. Absil and R. Sepulchre, Manopt, a Matlab toolbox for optimization on manifolds, J. Mach. Learn. Res., **15** (2014), 1455–1459.

[8] G. Hori, Comparison of two main approaches to joint SVD, in: Proc. of 8th Int. Conf. Independent Component Analysis and Signal Separation, pp. 42–49, 2009.