ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

Rank-Constrainted Optimization: A Riemannian Manifold Approach*†

Guifang Zhou[1], Wen Huang[2], Kyle A. Gallivan[1], Paul Van Dooren[2], P.-A. Absil[2]

1- Florida State University - Department of Mathematics
1017 Academic Way, Tallahassee, FL 32306-4510, US

2- Université catholique de Louvain - ICTEAM Institute
Avenue G. Lemaître 4, 1348 Louvain-la-Neuve, Belgium

Abstract.
This paper presents an algorithm that solves optimization problems on a matrix manifold $\mathcal{M} \subseteq \mathbb{R}^{m\times n}$ with an additional rank inequality constraint. New geometric objects are defined to facilitate efficiently finding a suitable rank. The convergence properties of the algorithm are given and a weighted low-rank approximation problem is used to illustrate the efficiency and effectiveness of the algorithm.

1 Introduction

We consider low-rank optimization problems that combine a rank inequality constraint with a matrix manifold constraint:

$$\min_{X\in\mathcal{M}_{\leq k}} f(X), \tag{1}$$

where $k \leq \min(m,n), \mathcal{M}_{\leq k} = \{X \in \mathcal{M} | \text{rank}(X) \leq k\}$ and $\mathcal{M}$ is a submanifold of $\mathbb{R}^{m\times n}$. Typical choices for $\mathcal{M}$ are: the entire set $\mathbb{R}^{m\times n}$, a sphere, symmetric matrices, elliptopes, spectrahedrons.

Recently, optimization on manifolds has attracted significant attention. Attempts have been made to understand problem (1) by considering a related but simpler problem $\min_{X\in\mathbb{R}_k^{m\times n}} f(X)$, where $\mathbb{R}_k^{m\times n} = \{X \in \mathbb{R}^{m\times n} | \text{rank}(X) = k\}$, (see e.g., [1, 2]). Since $\mathbb{R}_k^{m\times n}$ is a submanifold of $\mathbb{R}^{m\times n}$ of dimension $(m+n-k)k$, the simpler problem can be solved using techniques from Riemannian optimization [3] applied to matrix manifolds. However, a disadvantage is that the manifold $\mathbb{R}_k^{m\times n}$ is not closed in $\mathbb{R}^{m\times n}$, which complicates considerably the convergence analysis and performance of an iteration.

Very recently a more global view of a projected line-search method on $\mathbb{R}_{\leq k}^{m\times n} = \{X \in \mathbb{R}^{m\times n} | \text{rank}(X) \leq k\}$ along with a convergence analysis has been developed in [4]. In [5], the results of [4] are exploited to propose an algorithm that successively increases the rank by a given constant. Its convergence result to critical

*This research was supported by the National Science Foundation under grant NSF-1262476.

†This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. This work was supported by grant FNRS PDR T.0173.13.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

points can be deduced from [4, Th. 3.9]; it relies on the assumption, often satisfied in practice, that the limit points have rank k. Under this assumption, a line-search method on $\mathbb{R}^{m\times n}_{\leq k}$ is ultimately the same as a line-search method on $\mathbb{R}^{m\times n}_{k}$.

In this paper, we develop an efficient algorithm for the optimization problems. The main contributions are as follows: First, we generalize the admissible set from $\mathbb{R}^{m\times n}_{\leq k}$ to $\mathcal{M}_{\leq k}$. Second, the proposed algorithm solves a rank inequality constrained problem while finding a suitable rank for approximation of the locally optimal solution. Third, the proposed algorithm successively increases or decreases the rank by an adaptively-chosen amount; therefore, the rank is updating appropriately, which avoids excessive increase of rank in order to save computations and storage. Finally, theoretical convergence results are given.

The rest of this paper is organized as follows. A tangent cone descent algorithm is given in Section 2. Some theoretical results are presented in Section 3 without proofs. Finally, a performance study is provided on the weighted low-rank problem [6] in Section 4.

2 A Tangent Cone Descent Algorithm

The following assumptions are made to hold throughout this paper: (i) The closure of $\mathcal{M}_r := \{X \in \mathcal{M} | \text{rank}(X) = r\}$ is a subset of or equal to $\mathcal{M}_{\leq r}$. (ii) $\mathcal{M}_r$ is a manifold. (iii) The extension of cost function f in (1) on $\mathcal{M}$ is well-defined and it is continuously differentiable.

The basic idea of the new algorithm is applying Riemannian optimization methods on a fixed-rank manifold $\mathcal{M}_r$ while efficiently and effectively updating the rank r. For each fixed-rank manifold $\mathcal{M}_r$, efficient and well-understood Riemannian optimization algorithms, see e.g., [7, 8, 9], can be used directly. The main issue is to move from one fixed-rank manifold to another efficiently and effectively.

A rank adjustment decision considers the following two functions: one is an extension of f on $\mathcal{M}$, i.e., $f_{\text{F}} : \mathcal{M} \to \mathbb{R} : X \mapsto f_{\text{F}}(X)$, such that $f = f_{\text{F}}\big|_{\mathcal{M}_{\leq k}}$, and the other is the restriction of f on a fixed-rank manifold $\mathcal{M}_r$, i.e., $f_r : \mathcal{M}_r \to \mathbb{R} : X \mapsto f_r(X)$, such that $f_r = f|_{\mathcal{M}_r}$. The rank is increased at $x \in \mathcal{M}_r$ if $r < k$ and the following two conditions hold simultaneously: Condition I (angle threshold ϵ_1): $\tan(\angle(\text{grad}f_{\text{F}}(X), \text{grad}f_r(X))) > \epsilon_1$; Condition II (difference threshold ϵ_2): $\|\text{grad}f_{\text{F}}(X) - \text{grad}f_r(X)\| \geq \epsilon_2$, where $\text{grad}f_{\text{F}}(X)$ and $\text{grad}f_r(X)$ denote the Riemannian gradients of f_{F} and f_r at a point $X \in \mathcal{M}_r$.

The parameters ϵ_1 and ϵ_2 are important for finding the exact/approximate solutions and controlling the computational efficiency of the method. A smaller ϵ_1 value makes it easier to increase rank per iteration. The smaller ϵ_2 is the stricter the accuracy of the approximate local minimizer required. In particular, convergence to critical points of (1) is obtained if ϵ_2 is set to 0.

Next, we discuss the details about updating the rank. The tangent cone to $\mathcal{M}_{\leq k}$ at a point $X \in \mathcal{M}_{\leq k}$, $\text{T}_X\mathcal{M}_{\leq k}$, is equal to the set of all smooth admissible directions for $\mathcal{M}_{\leq k}$ at X (see e.g., [10]). Note that $\text{T}_X\mathcal{M}_{\leq k}$ is a linear space

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

for $X \in \mathcal{M}_k$ but is not for any point $X \in \mathcal{M}_{\leq k-1}$. The tangent cone $\mathrm{T}_X\mathcal{M}_{\leq k}$ contains the tangent space $\mathrm{T}_X\mathcal{M}_r$ and also the curves approaching X by points of rank greater than r, but not beyond k, i.e., $\mathrm{T}_X\mathcal{M}_{\leq k} = \mathrm{T}_X\mathcal{M}_r + \{\eta_{k-r} \in \mathrm{N}_X\mathcal{M}_r \bigcap \mathrm{T}_X\mathcal{M} | \mathrm{rank}(\eta_{k-r}) \leq k-r\}$. The characterization of $\mathrm{T}_X\mathcal{M}_{\leq k}$ for $\mathcal{M} = \mathbb{R}^{m\times n}$ has been given in [4].

To obtain the next iteration on $\mathcal{M}_{\leq k} \subseteq \mathcal{M}$, a general line-search method is considered: $X_{n+1} = \tilde{R}_{X_n}(t_n\eta_{X_n,\tilde{r}})$, where t_n is a step size and $\tilde{R}$ is a rank-related retraction defined below in Definition 1.

Definition 1 (Rank-related retraction) *Let $X \in \mathcal{M}_r$. A mapping $\tilde{R}_X : \mathrm{T}_X\mathcal{M} \to \mathcal{M}$ is a* rank-related retraction *if, $\forall \eta_X \in \mathrm{T}_X\mathcal{M}$, (i) $\tilde{R}_X(0) = X$, (ii) $\exists \delta > 0$ such that $[0,\delta) \ni t \mapsto \tilde{R}_X(t\eta_X)$ is smooth and $\tilde{R}_X(t\eta_X) \in \mathcal{M}_{\tilde{r}}$ for all $t \in [0,\delta)$, where $\tilde{r}$ is the integer such that $r \leq \tilde{r}$, $\eta_X \in \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}}$, and $\eta_X \notin \mathrm{T}_X\mathcal{M}_{\tilde{r}-1}$, (iii) $\frac{d}{dt}\tilde{R}_X(t\eta_X)|_{t=0} = \eta_X$.*

It can be shown that a rank-related retraction always exists.

Note $\tilde{R}_X$ is not necessarily a retraction on $\mathcal{M}$ since it may not be smooth on the tangent bundle $\mathrm{T}\mathcal{M} := \bigcup_{X\in\mathcal{M}} \mathrm{T}_X\mathcal{M}$. The modified Riemannian optimization algorithm is sketched as Alg. 1.

Generally speaking, η^* in Step 7 in Alg. 1 is not unique and any one can be used to change to rank $\tilde{r}$. The key to efficiency of the algorithm is to choose $\tilde{r}$ correctly and then choose an appropriate η^*. In [4], a similar idea is developed for $\mathcal{M} = \mathbb{R}^{m\times n}$ and $\tilde{r} = k$. The authors define the search direction as the projection of the negative gradient onto the tangent cone. If $\mathcal{M} = \mathbb{R}^{m\times n}$ and $\tilde{r} = k$, the choice of η^* in Step 7 is equivalent to the definition in [4, Corollary 3.3].

3 Main Theoretical Results

Suppose the cost function f is continuously differentiable and the algorithm chosen in Step 2 of Alg. 1 has the property of global convergence to critical points of f_r, we give the following main results. The proofs are omitted due to space constraints (see [13]).

Theorem 1 (Global Result) *The sequence $\{X_n\}$ generated by Alg. 1 satisfies $\liminf_{n\to\infty} \|P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}(\mathrm{grad} f_{\mathrm{F}}(X_n))\| \leq \left(\sqrt{1+\frac{1}{\epsilon_1^2}}\right)\epsilon_2$.*

Theorem 2 (Local Result) *Let f_{F} be a C^2 function and X_* be a nondegenerate minimizer of f_{F} with rank r_*. Suppose $\epsilon_2 > 0$. Denote the sequence of iterates generated by Algorithm 1 by $\{X_n\}$.*

There exists a neighborhood of X_*, $\mathcal{U}_{X_*}$, such that if $\mathcal{D} = \{X \in \mathcal{M} | f(X) \leq f(X_0)\} \subset \mathcal{U}_{X_*}$; $\mathcal{D}$ is compact; $\hat{f} : \mathrm{T}\mathcal{M} \to \mathbb{R} : \xi \mapsto f_{\mathrm{F}} \circ \tilde{R}(\xi)$ is a radially L-C^1 function with sufficient large β_{RL} defined in [3, Definition 7.4.1] such that for any $X, Y \in \mathcal{D}$, $\|\tilde{R}_X^{-1}(Y)\| < \beta_{RL}$, then there exists $N > 0$ such that

$$\forall n > N \quad rank(X_n) = r \quad and \quad X_n \in \mathcal{U}_{X_*}.$$

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Algorithm 1 Modified Riemannian Optimization Algorithm

for $n = 0, 1, 2, \ldots$ **do**
  Apply a Riemannian algorithm (e.g. one of GenRTR [7], RBFGS [8, 9, 11], RTR-SR1 [9, 12]) to approximately optimize f over $\mathcal{M}_r$ with initial point X_n and stop at $\tilde{X}_n \in \mathcal{M}_r$, where either $\|\mathrm{grad} f_r(\tilde{X}_n)\| < \epsilon_3$ ($\mathtt{flag} \leftarrow 1$) or $\tilde{X}_n$ is close to $\mathcal{M}_{\leq r-1}$ ($\mathtt{flag} \leftarrow 0$);
  if $\mathtt{flag} = 1$ **then**
    if Both Conditions I and II are satisfied **then**
      Set $\tilde{r}$ and η^* to be r and $\mathrm{grad} f_r(\tilde{X}_n)$ respectively;
      while $\|\mathrm{grad} f_{\mathrm{F}}(\tilde{X}_n) - \eta^*\| > \frac{\epsilon_1}{2}\|\eta^*\|$ **do**
        Set $\tilde{r}$ to be $\tilde{r} + 1$ and $\eta^* \in \mathrm{argmin}_{\eta \in \mathrm{T}_X \mathcal{M}_{\leq \tilde{r}}} \|\mathrm{grad} f_{\mathrm{F}}(\tilde{X}_n) - \eta\|_F$;
      end while
      Obtain X_{n+1} by applying an Armijo-type line search algorithm along η^* using a rank-related retraction;
    else
      If ϵ_3 is small enough, stop. Otherwise, $\epsilon_3 \leftarrow \tau\epsilon_3$, where $\tau \in (0, 1)$;
    end if
  else $\{\mathtt{flag} = 0\}$
    If the rank has not been increased on any previous iteration, reduce the rank of $\tilde{X}_n$ based on truncation while keeping the function value decrease, update r, obtain next iterate X_{n+1};
    Else reduce the rank of $\tilde{X}_n$ such that the next iterate X_{n+1} satisfies $f(X_{n+1}) - f(X_i) \leq c(f(X_{i+1}) - f(X_i))$, where i is such that the latest rank increase was from X_i to X_{i+1}, $0 < c < 1$. Set r to be the rank of X_{n+1};
  end if
end for

4 Application

We illustrate the effectiveness of Alg. 1 for the weighted low-rank approximation problem:

$$\min_{X \in \mathcal{M}_{\leq k}} \|A - X\|_W^2, \quad f(X) = \|A - X\|_W^2 = \mathrm{vec}\{A - X\}^T W \mathrm{vec}\{A - X\} \quad (2)$$

where $\mathcal{M} = \mathbb{R}^{80\times 10}$, A is given, $W \in \mathbb{R}^{800\times 800}$ is a positive definite symmetric weighting matrix and $\mathrm{vec}\{A\}$ denotes the vectorized form of A, i.e., a vector constructed by stacking the consecutive columns of A in one vector.

The matrix A is generated by $A_1 A_2^T \in \mathbb{R}^{m\times n}$, where $A_1 \in \mathbb{R}^{80\times 5}$, $A_2 \in \mathbb{R}^{10\times 5}$. $W = U\Sigma U^T$, where $U \in \mathbb{R}^{800\times 800}$ is a random orthogonal matrix generated by Matlab's QR and RAND. The mn singular values of the weighting matrix are generated by Matlab function LOGSPACE with condition number 100 and multiplying, element-wise, by a uniform distribution matrix on the interval $[0.5, 1.5]$. Three values of k are considered, one less than the true rank, one equal to the

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

k	method	rank	f	Rel Err($\frac{\|A-X\|_W}{\|A\|_W}$)	time(sec)
k = 3	Alg. 1	3	8.657_{+01}	3.108_{-01}	6.162_{-01}
	DMM	3	8.657_{+01}	3.108_{-01}	3.032_{+00}
	SULS	3	8.657_{+01}	3.108_{-01}	3.967_{+00}
	APM	3	8.657_{+01}	3.108_{-01}	9.115_{+00}
k = 5	Alg. 1	5	3.293_{-22}	1.093_{-13}	5.195_{-01}
	DMM	5	1.257_{-20}	3.453_{-12}	3.052_{+00}
	SULS	5	3.056_{-12}	5.419_{-08}	1.067_{+00}
	APM	5	8.281_{-10}	8.930_{-07}	4.052_{+00}
k = 7	Alg. 1	5	3.652_{-17}	1.873_{-10}	4.434_{-01}
	DMM	5.02(99/100)	2.131_{-17}	3.740_{-11}	1.636_{+00}
	SULS	7(0/100)	2.236_{-12}	4.775_{-08}	2.075_{+00}
	APM	7(0/100)	2.349_{-10}	4.865_{-07}	6.895_{+00}

Table 1: Method comparisons. The number in the parenthesis indicates the fraction of experiments where the numerical rank (number of singular values greater than 10^{-8}) found by the algorithm equals the true rank. The subscript $\pm k$ indicates a scale of $10^{\pm k}$. Alg. 1 and SULS, are stopped when the norm of the final gradient on the fixed-rank manifold over the norm of initial full gradient is less than 10^{-8} while DMM and APM are stopped when the norm of final gradient over the norm of initial gradient is less than 10^{-7}.

true rank and one greater than the true rank. Four algorithms are compared: Alg. 1, DMM [14], SULS [4] and APM [6]. The initial point of Alg. 1 and SULS is a randomly generated rank-1 matrix and the initial points of DMM and APM are randomly generated n-by-$(n-k)$ and m-by-$(m-k)$ matrices respectively.

Results shown in Table 1 are the average of 100 runs for different data matrices R, weighting matrices W and initial points. The results show that Alg. 1 with $\epsilon_1 = \sqrt{3}, \epsilon_2 = 10^{-4}$ (GenRTR is used as the inner algorithm, when the sizes become larger, limited-memory RBFGS [9, 11], for example, may be a better choice) has significant advantages compared with the other methods. It achieves good accuracy in the approximation with less computational time. Furthermore, for Alg. 1, all Riemannian objects, their updates and X_n are based on an efficient three factor, U, D, V, representation, and the singular values are immediately available while, for the other three methods, an additional SVD is required.

References

[1] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre. Low-rank optimization with trace norm penalty. *SIAM Journal on Optimization*, 23(4):2124–2149, 2013.

[2] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimiza-*

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

tion, 20(5):2327–2351, 2010.

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

[4] R. Schneider and A. Uschmajew. Convergence results for projected line-search methods on varieties of low-rank matrices via Lojasiewicz inequality. *ArXiv e-prints*, February 2014.

[5] A. Uschmajew and B. Vandereycken. Line-search methods and rank increase on low-rank matrix varieties. In *Proceedings of the 2014 International Symposium on Nonlinear Theory and its Applications (NOLTA2014)*, 2014.

[6] W.-S. Lu, S.-C. Pei, and P.-H. Wang. Weighted low-rank approximation of general complex matrices and its application in the design of 2-d digital filters. *IEEE Transactions on Circuits and Systems I*, 44:650–655, 1997.

[7] P.-A. Absil, C. G. Baker, and Kyle A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.

[8] W. Ring and B. Wirth. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2):596–627, January 2012.

[9] W. Huang. *Optimization algorithms on Riemannian manifolds with applications*. PhD thesis, Florida State University, 2013.

[10] Donal B. O'Shea and Leslie C. Wilson. Limits of tangent spaces to real surfaces. *American Journal of Mathematics*, 126(5):pp. 951–980, 2004.

[11] Wen Huang, K. A. Gallivan, and P.-A. Absil. A Broyden class of quasi-Newton methods for Riemannian optimization. UCL-INMA-2014.01, U.C.Louvain, Submitted for publication, 2015.

[12] W. Huang, P.-A. Absil, and K. A. Gallivan. A Riemannian symmetric rank-one trust-region method. *Mathematical Programming*, February 2014. doi:10.1007/s10107-014-0765-1.

[13] Guifang Zhou. *Rank-contrained optimization: A Riemannian approach*. PhD thesis, Florida State University. in preparation.

[14] I. Brace and J. H. Manton. An improved BFGS-on-manifold algorithm for computing low-rank weighted approximations. In *Proceedings of 17th International Symposium on Mathematical Theory of Networks and Systems*, pages 1735–1738, 2006.