Greedy rank updates combined with Riemannian descent methods for low-rank optimization

André Uschmajew
Hausdorff Center for Mathematics &
Institute for Numerical Simulation
University of Bonn
53115 Bonn, Germany
Email: uschmajew@ins.uni-bonn.de

Bart Vandereycken
Section de mathématiques
University of Geneva
1211 Geneva, Switzerland
Email: bart.vandereycken@unige.ch

***Abstract*—We present a rank-adaptive optimization strategy for finding low-rank solutions of matrix optimization problems involving a quadratic objective function. The algorithm combines a greedy outer iteration that increases the rank and a smooth Riemannian algorithm that further optimizes the cost function on a fixed-rank manifold. While such a strategy is not especially novel, we show that it can be interpreted as a perturbed gradient descent algorithms or as a simple warm-starting strategy of a projected gradient algorithm on the variety of matrices of bounded rank. In addition, our numerical experiments show that the strategy is very efficient for recovering full rank but highly ill-conditioned matrices that have small numerical rank.**

I. Introduction

Our goal is minimizing a smooth convex function $f\colon \mathbb{R}^{m\times n} \to \mathbb{R}$ where at least one of the solutions X_* can be well approximated by a matrix of low rank. Besides the obvious case $\operatorname{rank}(X_*) \ll \min(m,n)$, this also includes the situation where the singular values of a full rank matrix X_* decay sufficiently fast.

A typical application is the low-rank matrix completion problem (see, e.g., [3])

$$f(X) = \tfrac{1}{2}\|P_\Omega(X) - P_\Omega(X_*)\|_F^2$$

where P_Ω is the orthogonal projection onto a subset $\Omega \in [m]\times[n]$ of the entries of matrices in $\mathbb{R}^{m\times n}$. More generally, we can also consider the energy (semi-)norm minimization

$$\begin{aligned} f(X) &= \tfrac{1}{2}\langle X, \mathcal{A}(X)\rangle - \langle X, B\rangle \\ &= \|X - X_*\|_{\mathcal{A}}^2 + \text{constant} \end{aligned}$$

for linear systems where $\mathcal{A}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is a linear, symmetric and positive semidefinite operator, and $B = \mathcal{A}(X_*)$ is the known right-hand side. Applications of this kind are low-rank solvers of matrix equations, like the Lyapunov equation $\mathcal{A}(X) = AX + XA$; see [15] for an overview.

Denoting the set of matrices of constant rank k by

$$\mathcal{M}_k = \{X \in \mathbb{R}^{m\times n} \mid \operatorname{rank}(X) = k\},$$

a possible strategy for obtaining a rank-k approximation to X_* is solving

$$\min f(X) \qquad \text{s.t. } X \in \mathcal{M}_k \tag{1}$$

978-1-4673-7353-1/15/$31.00 ©2015 IEEE

by a local descent method. Since $\mathcal{M}_k$ is a real-analytic submanifold of $\mathbb{R}^{m\times n}$ of dimension $(m+n-k)k$ (see, e.g, [22, Ex. 1.7]), the above problem can be solved by well-established techniques from Riemannian optimization [1], [4], [11], [20]. These algorithms are closely related to those based on iterative hard thresholding [17] and can be seen as alternatives to block coordinate techniques like alternating direction methods based on the low-rank decomposition $X = UV^T$ with U and V tall matrices; see, e.g., [23].

Compared to algorithms based on convex relaxations, these non-convex methods can be very effective and efficient. However, they have a number of obvious practical and theoretical disadvantages: 1) the choice of the rank k has a large influence on the convergence and approximation quality, 2) the manifold $\mathcal{M}_k$ is not closed hence the theoretical analysis is complicated, 3) being a local method, there is no guarantee of recovery of X_*, and 4) they converge very slowly when recovering ill-conditioned matrices X_*.

In the rest of this paper, we will show that an outer iteration involving a greedy rank-one increase combined with an inner optimization on the set of matrices of rank at most k,

$$\mathcal{M}_{\le k} = \{X \in \mathbb{R}^{m\times n} \mid \operatorname{rank}(X) \le k\},$$

successfully addresses these problems. Regarded as standalone algorithms, both iterations are not new and in fact used in more elaborate formulations (see later). However, it is their combination that makes them successful empirically. The proposed algorithm is conceptually identical to Riemannian Pursuit from [16] for matrix completion but we present it in a more general way using recent results from [18] involving greedy rank-one updates. Parts of the presented results and numerical experiments already appeared in [19].

II. Line-search methods and tangent cone

Since a sequence of matrices in $\mathcal{M}_k$ may converge to a matrix of rank strictly smaller than k, the manifold $\mathcal{M}_k$ is not closed in $\mathbb{R}^{m\times n}$. This poses a serious difficulty if we want to show by standard arguments the convergence of a descent method applied to (1): even when assuming that f has bounded sublevel sets, we can neither guarantee that (1)

has a solution, nor that the sequence of iterates generated by our descent algorithm will have cluster points in $\mathcal{M}_k$.

A. Line search methods on $\mathcal{M}_{\leq k}$

The set $\mathcal{M}_{\leq k}$ is the closure of $\mathcal{M}_k$. Hence, the problem

$$\min f(X) \qquad \text{s.t. } X \in \mathcal{M}_{\leq k} \tag{2}$$

is formulated over a closed set and as such it does not suffer from the previous fundamental problems. For instance, the existence of a global minimum is guaranteed for continuous and coercive f. Furthermore, since in practice the (numerical) rank of X_* is unknown, problem (2) with a suitable upper bound for k seems more appropriate than (1) for which k needs to be known exactly.

Unfortunately, one cannot use Riemannian optimization techniques anymore to solve (2) since the set $\mathcal{M}_{\leq k}$ is not a smooth manifold. For example, the Riemannian gradient of f at $X \in \mathcal{M}_{\leq k}$ is only defined in points of full rank k. Hence, smooth line-search algorithms for (2), like projected steepest descent, formally cease to exist and become "non-smooth". In addition, one can show [20] that the (Riemannian) condition number of the second-order approximation of f restricted to $\mathcal{M}_k$ involves terms that are $O(1/\sigma_k(X))$ with $\sigma_k(X)$ the kth (or smallest nonzero) singular value of X. This suggests that Riemannian optimization algorithms on $\mathcal{M}_k$ may converge arbitrarily slow on $\mathcal{M}_{\leq k}$ when $\sigma_{\min}(X) \to 0$.

Instead of solving (2) directly with a Riemannian descent algorithm, we therefore follow [14] and consider general line-search methods on $\mathcal{M}_{\leq k}$:

$$X_{j+1} = P_{\leq k}(X_j + \alpha_j \Xi_j). \tag{3}$$

Here, Ξ_j is a descent direction in the tangent cone at X_j (see below), the step size α_j is determined by Armijo backtracking, and $P_{\leq k}$ is the metric projection onto $\mathcal{M}_{\leq k}$ in the Frobenius norm which can be computed by a truncated SVD of $X_j + \alpha_j \Xi_j$. Thanks to $P_{\leq k}$, iteration (3) generates $X_j \in \mathcal{M}_{\leq k}$ for all j. In addition, for sufficiently gradient-related tangential search directions Ξ_n, Cor. 2.11 and Thm. 3.9 in [14] guarantee that cluster points of (3) converge to a stationary point with some fixed but unknown asymptotic rate.

The crucial difference of (3) compared to, for example, Riemannian steepest descent that uses in X the Riemannian gradient as search direction

$$G_k(X) = \operatorname*{argmin}_{\Xi \in T_X \mathcal{M}_k} \| - \nabla f(X) - \Xi \|_F,$$

is that the search direction now has to be the steepest direction on the whole tangent cone,

$$G_{\leq k}(X) \in \operatorname*{argmin}_{\Xi \in T_X \mathcal{M}_{\leq k}} \| - \nabla f(X) - \Xi \|_F. \tag{4}$$

It is remarkable that the tangent cone of $\mathcal{M}_{\leq k}$ admits a concise expression such that the non-convex problem (4) has a closed-form solution which in turn leads to an efficient implementation of (3).

B. Tangent cone of $\mathcal{M}_{\leq k}$

Recall the definition of the tangent cone to a closed set $\mathcal{M} \subset \mathbb{R}^n$ at a point $x \in \mathcal{M}$ (see, e.g., [12]):

$$T_x\mathcal{M} = \{\xi \in \mathbb{R}^n \mid \exists\, (x_j) \subseteq \mathcal{M}, \exists\, (a_j) \subseteq \mathbb{R}_+ \text{ s.t. } x_j \to x,\ a_j \to +\infty,\ a_j (x_j - x) \to \xi\}. \tag{5}$$

To derive the tangent cone of $\mathcal{M}_{\leq k}$, we follow [4] and [14]. Let $X \in \mathcal{M}_{\leq k}$ have rank $s \leq k$, then we have for some $S \in \mathbb{R}^{s \times s}$ and orthonormal $U \in \mathbb{R}^{m \times s}$, $V \in \mathbb{R}^{n \times s}$ that

$$X = USV^T.$$

It is well known that $T_X\mathcal{M}_s$, that is, the tangent space at X as element of the smooth manifold $\mathcal{M}_s$, consists of all matrices of the form

$$\begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} V & V_\perp \end{bmatrix}^T \in T_X\mathcal{M}_s,$$

where $A \in \mathbb{R}^{s \times s}$, $B \in \mathbb{R}^{s \times (n-s)}$, and $C \in \mathbb{R}^{(m-s) \times s}$ are arbitrary. Obviously, $T_X\mathcal{M}_s$ has to be a subset of $T_X\mathcal{M}_{\leq k}$. However, when $s < k$ we can approach X by matrices of rank larger than s in the definition (5), which produces additional tangential vectors. Thm. 3.2 in [14] shows that

$$T_X\mathcal{M}_{\leq k} = T_X\mathcal{M}_s + \{\Xi_{k-s} \in (T_X\mathcal{M}_s)^\perp \mid \operatorname{rank}(\Xi_{k-s}) \leq k - s\},$$

with $^\perp$ the standard orthogonal complement. In other words, elements of the tangent cone at X satisfy

$$\begin{aligned} T_X\mathcal{M}_{\leq k} \ni \Xi &= \Xi_s + \Xi_{k-s} \\ &= \begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} V & V_\perp \end{bmatrix}^T, \end{aligned} \tag{6}$$

with A, B, and C as before, and $D \in \mathbb{R}^{(m-s) \times (n-s)}$ such that $\operatorname{rank}(D) \leq k - s$.

Let us now consider what (6) implies for the computation of (4), that is, the metric projection of the negative gradient on the tangent cone. When $\operatorname{rank}(X) = k$, $G_{\leq k}(X) = G_k(X)$ is just the orthogonal projection on $T_X\mathcal{M}_k$ which is given by

$$\begin{aligned} G_k(X) = {} & U\,U^T \nabla f(X) V\,V^T \\ & - U\,U^T \nabla f(X) - \nabla f(X) V\,V^T. \end{aligned} \tag{7}$$

If $\nabla f(X)$ and $(\nabla f(X))^T$ admit a fast matrix vector product (for example, when $\nabla f(X)$ is sparse or low-rank) then $G_k(X)$ can be efficiently implemented for large m, n. If $\operatorname{rank}(X) = s < k$, one first calculates the metric projection $G_s(X)$ of $-\nabla f(X)$ on $T_X\mathcal{M}_s$ (which is formally the same expression as (7)), and then a best rank $k - s$ approximation Ξ_{k-s} of $-\nabla f(X) - G_s(X)$ to obtain

$$G_{\leq k}(X) = G_s(X) + \Xi_{k-s}. \tag{8}$$

When computing matrix vector products with $-\nabla f(X) - G_s(X)$, one can again exploit structure in $\nabla f(X)$ and $G_s(X)$. This makes the computation of Ξ_{k-s} scalable using methods form large-scale and sparse SVD calculations or randomized low-rank techniques; see, e.g., [7], [10].

C. Practical considerations

The projected gradient scheme (3) reduces to Riemannian steepest descent as long as the iterates remain of maximal rank. However, seen as an algorithm on $\mathcal{M}_k$, the projections on $\mathcal{M}_k$ might only be well-defined for α_j very small. For example, $\alpha_j \to 0$ when the iterates (X_j) want to accumulate to a point with rank less than k. Fortunately, when the global minimizer X_* of f has a (numerical) rank larger or equal than k, numerical experience shows that Riemannian optimization algorithms for (1) never encounter such singular points. One can therefore, employ more sophisticated Riemannian algorithms for (1) like Newton's method or nonlinear CG. In addition, Cor. 3.4 in [14] guarantees that unless a global minimizer X_* of f has rank strictly smaller than k, any stationary point of (2) will be of maximal rank k. In our numerical experiments, we will therefore use the LRGeomCG method from [20] to solve (2).

III. Greedy rank updates

We now argue how the tangent cone of $\mathcal{M}_{\leq k}$ can be used to gradually increase the rank of X_j by wrapping (2) into an outer iteration. At the same time, this rank increase is guaranteed to decrease the objective f by a fixed amount in every step.

A. Warm starting projected gradient descent

Suppose an iterate X_j of the line-search method (3) has rank $s < k$, then any search direction Ξ_j of the form (6) with nonzero D will locally increase the rank for the next iterate by $\operatorname{rank}(D) \leq k - s$. This observation can be used to systematically increase the rank of X_j in a outer iteration: Given a locally optimal solution $X_j \in \mathcal{M}_{\leq k}$, warm start the line search on $\mathcal{M}_{k+\ell}$ in the point X_j for some $\ell > 0$. Observe that we cannot use a step of a smooth Riemannian optimization to accomplish this, but algorithm (3) is still a valid descent method since it stays formally the same. However, as soon as the iterate on $\mathcal{M}_{k+\ell}$ has maximal rank, one can again switch to a faster smooth Riemannian method. The result is the rank-adaptive algorithm, listed in Fig. 1.

Let us now verify how the first step of (3) looks like on $\mathcal{M}_{k+\ell}$. Since X_j is locally optimal, it will be a critical point of f on the smooth manifold $\mathcal{M}_s$ for some $s \leq k$, that is, $\nabla f(X_j) \in (T_{X_j}\mathcal{M}_s)^\perp$. We now embed X_j in $\mathcal{M}_{\leq k+\ell}$ and choose as search direction $\Xi_j = G_{\leq k+\ell}(X_j)$ the projection of the negative gradient on $T_{X_j}\mathcal{M}_{\leq k+\ell}$. By the considerations leading to (8), we compute Ξ_j as the sum of two terms. The first term (the projection onto $T_{X_j}\mathcal{M}_s$) is zero by optimality of X_j on $\mathcal{M}_s$. Hence, Ξ_n is just given by the best rank $k+\ell-s$ approximation of $-\nabla f(X_j)$. As a consequence, if Ξ_j is zero, then $\nabla f(X_j) = 0$ and we can terminate. Otherwise, the D matrix of Ξ_n is nonzero and because of orthogonality reasons, $P_{\leq k+\ell}$ is the identity operator and the next iterate $X_{j+1} = X_j + \alpha_j \Xi_j$ will have a rank of exactly $s + \operatorname{rank}(\Xi_j)$. Observe also that in this case the line search can be computed exactly for quadratic f.

Algorithm 1

$j \leftarrow 0$, $k \leftarrow 0$, $\ell \geq 1$, $X_j = 0$.
while not converged **do**
  $k \leftarrow k + \ell$ ▷ *Greedy rank ℓ update*
  $\Xi_j \leftarrow G_{\leq k}(X_j)$
  Compute exact step-size α_j by linear least-squares
  $X_{j+1} \leftarrow X_j + \alpha_j \Xi_j$
  $Y \leftarrow X_{j+1}$ ▷ *Perform inner optimization*
  while $\|G_{\leq k}(Y)\|_F \geq$ `tol` **do**
    Choose search-direction $\Phi \in T_Y \mathcal{M}_{\leq k}$
    Perform line-search for step size α_*
    $Y_+ \leftarrow P_{\leq k}(Y + \alpha_* \Phi)$
    $Y \leftarrow Y_+$
  end while
  $X_{j+1} \leftarrow Y$
  $j \leftarrow j + 1$
end while

Fig. 1. Greedy rank increase with line-search on $\mathcal{M}_{\leq k}$.

B. Perturbed steepest descent

Similar rank increasing schemes that use the best rank ℓ approximation of $-\nabla f(X_j)$ have been proposed before, for example, in [11], [16], and for tensors in [6]. By considering the truncated SVD, it is not hard to show that for $\ell \geq 1$ such search directions Ξ_j are gradient related,

$$\langle -\nabla f(X_j), \Xi_j \rangle \geq \frac{1}{\sqrt{\min(m,n)}} \|\nabla f(X_j)\|_F \|\Xi_j\|_F.$$

Hence, if these greedy rank updates are obtained from a suitable line search, they can be seen as a perturbed gradient descent step on f. In fact, [18] shows that for an exact line search and a quadratic f, the error behaves as

$$\|X_j - X_*\|_{\mathcal{A}}^2 \leq \left(1 - \frac{2\lambda_{\min}}{5\lambda_{\max}\min(m,n)}\right)^{2j} \|X_0 - X_*\|_{\mathcal{A}}^2.$$

Here $\lambda_{\min}$ is the smallest non-zero eigenvalue of $\mathcal{A}$ (which is one for P_Ω). We remark that only for strictly positive definite $\mathcal{A}$ this bound implies $X_j \to X_*$. In case of matrix completion, for example, more refined analyses are required.

Alg. 1 employs the greedy rank updates as an outer iteration. Since its inner iteration is a descent method, it can only improve the objective function. Hence Alg. 1 can be seen as an accelerated (perturbed) gradient scheme as introduced in [2] and so the convergence of the outer iteration still satisfies the bound from above. Observe also that this bound is valid for any point X_j, regardless whether it is locally optimal. Hence, we do not need $\nabla f(X_j) \in (T_{X_j}\mathcal{M}_s)^\perp$ like in the preceding discussion and we can safely terminate the inner optimization at a very crude approximation of a locally optimal X_j.

IV. Numerical experiments

We illustrate the performance of Alg. 1 and a few other algorithms for the matrix completion problem

$$\min \operatorname{rank}(X) \qquad \text{s.t.} \quad P_\Omega(X) = P_\Omega(A), \tag{9}$$

where $A \in \mathbb{R}^{m\times n}$ is an unknown low-rank matrix that is only known on a subset Ω of all its entries. We focus on the case where A has exponentially decaying singular values since most existing methods perform quite badly in this setting.

Under certain conditions (see [3] for more details), problem (9) has a unique solution X_*. By writing

$$\min f(X) = \tfrac{1}{2}\|P_\Omega(X) - P_\Omega(A)\|_F^2$$

we can solve (9) by Alg. 1.

For all experiments, the inner optimization on lines 8–13 in Alg. 1 is solved using LRGeomCG [20] with a strong Wolfe line-search. We remark that the required curvature condition (involving the derivative of the truncated SVD) can be computed cheaply using [5]. The reported relative errors for iterate X_j are computed as

$$\text{rel. error} = \|P_\Lambda(X_j) - P_\Lambda(A)\|_F / \|P_\Lambda(A)\|_F,$$

where training error uses $\Lambda = \Omega$, and testing error uses another randomly chosen set of indices $\Lambda = \Gamma$ with $|\Gamma| = |\Omega|$.

To compute Ξ_{k-s} for $G_{\leq k+\ell}(X_j)$, we use the randomized power algorithm from [7] which in our case requires $5(\ell+2)$ matrix vector products with $-\nabla f(X) - \Xi_s$.

A. Random matrix

Let $U, V \in \mathbb{R}^{n\times k_*}$ be random Gaussian matrices with $n = 1000$ and $k_* = 26$. Define the unknown matrix as

$$A = U\Sigma V^T,\ \Sigma = \operatorname{diag}(10^i), \qquad i = 0, -\tfrac{2}{5}, -\tfrac{4}{5}, \ldots, -\tfrac{50}{5}$$

and take Ω as $2.5 \cdot 2nk_*$ uniform random samples. This coincides with an oversampling of about 2.5 compared to the degrees of freedom in the rank 26 matrix A.

In Fig. 2, we see the training and testing error of Alg. 1 for $\ell = 1$. As stopping condition (line 8) we take a reduction of 10^{-5} of the relative gradient $\|G_{\leq k}(Y)\|_F / \|Y\|_F$. Observe that the matrix A is recovered up to a relative error of 10^{-12} and there is a good agreement between testing and training error. For each inner iteration, the relative residual is reduced by a factor of 10^{-5} in less than 30 iterations, showing that the warm-start using $G_{\leq k+\ell}(X_j)$ is very effective.

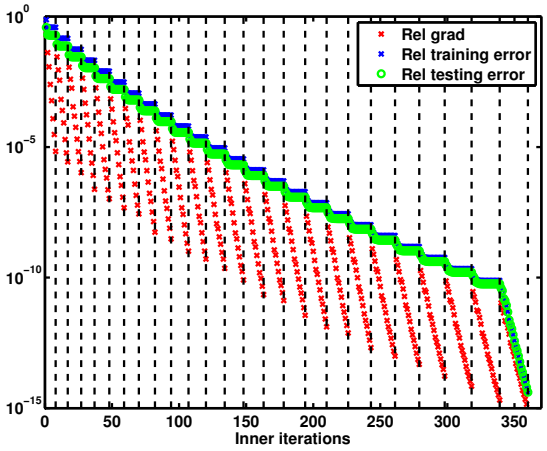
B. Bivariate function

As next example for A with exponentially decaying singular values, we discretize the bivariate function

$$h(x,y) = \frac{1}{1+(x-y)^2}, \qquad (x,y) \in [0,1]^2,$$

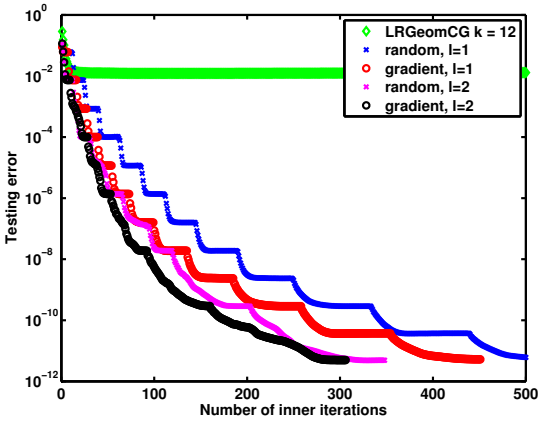
on a uniform grid with $m = n = 1000$ points between 0 and 1 for x and y. The numerical rank of A equals $k_* = 20$, so we construct Ω using $k_* = 20$ to obtain an oversampling rate of 2.5.

In Fig. 3, we see the testing error for $\ell = 1$ and $\ell = 2$ when we warm-start using the gradient (as explained in §III) in comparison to using random vectors orthogonal to X_j. In both cases, we run Alg. 1 until a maximum rank of 12 since for higher ranks the line-search for the sub-problems usually failed which makes the numerical results less interpretable. The table



Total time: 4.63 sec.

Fig. 2. Convergence of Alg. 1 with $\ell = 1$ for a random matrix with exponentially decaying singular values.



ℓ	strategy	its.	time (sec.)
1	random	532 (6.83)	15 (0.55)
1	gradient	453 (1.84)	13 (0.24)
2	random	388 (126)	11 (3.6)
2	gradient	296 (6.51)	8.1 (0.34)

Fig. 3. Convergence of Alg. 1 where Ξ_j is enlarged using a low-rank approximation of the gradient or using random vectors. The number between brackets is the standard deviation.

shows the mean and standard deviation of the number of inner iterations required to decrease the Frobenius norm of the projected gradient in every fixed-rank problem (including rank 12) to 10^{-4} for 10 random initializations. The corresponding error history for one realization is plotted in the curves above. In addition, we have also depicted the error of LRGeomCG when optimizing directly on a rank-12 manifold.

It is clear from the figure that the rank-adaptive strategy greatly improves the standard fixed-rank approach. Also, using a low-rank approximation of the gradient performs the best in terms of the number of iterations and total computational time. In addition, random vectors result in less predictable convergence compared to the gradient strategy.

C. Comparison with other greedy methods

The line search in steps 5–6 of Alg. 1 minimizes f along a one-dimensional affine subspace, that is, it solves

$$\min_{\alpha\in\mathbb{R}} f(X-\alpha\Xi) = \min_{Z\in\operatorname{span}(\Xi)} f(X+Z)$$

where $\Xi = G_{\leq k}(X)$ is the best rank ℓ approximation of $-\nabla f(X)$. Since f is a quadratic function, it is computationally feasible to compute

$$\min_{Z\in\mathcal{U}} f(X+Z)$$

with $\mathcal{U}$ some arbitrary subspace on $\mathbb{R}^{m\times n}\simeq\mathbb{R}^{mn}$. As long as $\mathcal{U}$ contains a gradient-related vector (like $G_{\leq k}(X)$), such an outer iteration will still be convergent by results from [2]. In addition, the resulting least-square problems can be computed efficiently by QR updating techniques [8].

For simplicity assume $\ell = 1$. Let $X = U\Sigma V^T$ and $\Xi = \sigma uv^T$ denote compact SVDs, then in addition to the choice

$$\mathcal{U}_{\text{1D}} := \operatorname{span}(\Xi) = \left\{suv^T : s\in\mathbb{R}\right\}$$

on which Alg. 1 and others are based, we can also consider

$$\mathcal{U}_{\text{full}} := \left\{\begin{bmatrix}U & u\end{bmatrix} S \begin{bmatrix}V & v\end{bmatrix}^T : S\in\mathbb{R}^{(k+1)\times(k+1)}\right\},$$
$$\mathcal{U}_{\text{diagonal}} := \left\{\begin{bmatrix}U & u\end{bmatrix} S \begin{bmatrix}V & v\end{bmatrix}^T : S_{i,j} = \delta_{i,j}\right\},$$

and variants thereof. This is for example the search space in [13], [21] after exploiting that X lies in $\mathcal{U}_{\text{full}}$ and $\mathcal{U}_{\text{diagonal}}$.

In Fig. 4, we have compared how these greedy strategies, which have a larger search space per rank-one update and are used without additional inner iteration, perform on the bivariate function from above. It is clear that, while they decrease the objective function linearly (as predicted), the rate of convergence is much slower compared to Alg. 1. In addition, there is sometimes a large discrepancy between testing and training error, showing that X_* is not approximated well.

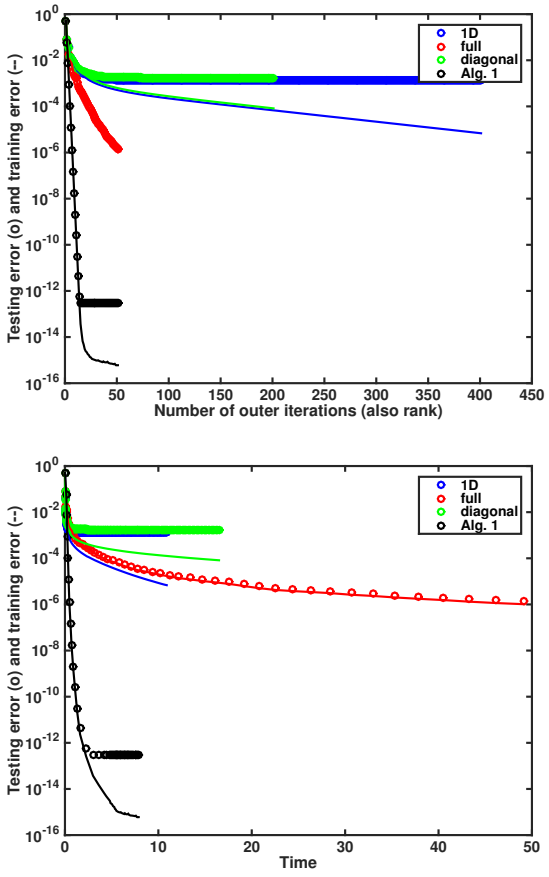


Fig. 4. Comparison of Alg. 1 with $\ell = 1$ to other greedy strategies without inner iteration.

References

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

[2] P.-A. Absil, and K. A. Gallivan, "Accelerated line-search and trust-region methods", *SIAM J. Numer. Anal.*, vol. 47, pp. 997–1018, 2009.

[3] E. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, pp. 717–772, 2009.

[4] T. P. Cason, P.-A. Absil, and P. Van Dooren, "Iterative methods for low rank approximation of graph similarity matrices," *Linear Algebra Appl.*, vol. 438, pp. 1863–1882, 2013.

[5] L. Dieci and T. Eirola, "On smooth decompositions of matrices", *SIAM J. Numer. Anal.*, vol. 20, pp. 800–819, 1999.

[6] S. V. Dolgov and D. V. Savostyanov, "Alternating minimal energy methods for linear systems in higher dimensions. Part I: SPD systems," *ArXiv e-print*, arXiv:1301.6068, 2013.

[7] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, pp. 217–288, 2011.

[8] S. Hammarling and C. Lucas, "Updating the QR factorization and the least squares problem", *Tech. Report, MIMS*, 2008.

[9] U. Helmke and M. A. Shayman, "Critical points of matrix least squares distance functions," Linear Algebra Appl., vol. 215, pp. 1–19, 1995.

[10] R. M. Larsen, "PROPACK—Software for large and sparse SVD calculations," `http://soi.stanford.edu/ rmunk/PROPACK`, 2004.

[11] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre, "Low-rank optimization with trace norm penalty," *SIAM J. Optim.*, vol. 23, pp. 2124–2149, 2013.

[12] R. T. Rockafellar and R. J.-B. Wets, "Variational Analysis," Springer-Verlag, Berlin, 1998.

[13] S. Shalev-Shwartz, A. Gonen, and O. Shamir, "Large-scale convex minimization with a low-rank constraint", *Proc. of ICML11*, 2011.

[14] R. Schneider and A. Uschmajew, "Convergence results for projected line-search methods on varieties of low-rank matrices via Łojasiewicz inequality," *ArXiv e-print*, arXiv:1402.5284, 2014 (to appear in SIAM J. Optim.).

[15] V. Simoncini,"Computational methods for linear matrix equations", *Tech. report. Università di Bologna*, 2014.

[16] M. Tan, I. W. Tsang, L. Wang, B. Vandereycken, and S. J. Pan, "Riemannian pursuit for big matrix recovery," *Proc. of ICML14*, 2014.

[17] J. Tanner and K. Wei, "Normalized iterative hard thresholding for matrix completion", *SIAM J. Scient. Comp.*, vol. 35, pp. S104–S125, 2013.

[18] A. Uschmajew, "Some results concerning rank-one truncated steepest descent directions in tensor spaces", *Proc. SAMPTA2015*, 2015.

[19] A. Uschmajew and B. Vandereycken, "Line-search methods and rank increase on low-rank matrix varieties", *Proc. NOLTA2014*, 2014.

[20] B. Vandereycken, "Low-rank matrix completion by Riemannian optimization," *SIAM J. Optim.*, vol. 23, pp. 1214–1236, 2013.

[21] Z. Wang, M. Lai, Z. Lu, and J. Ye, "Orthogonal rank-one matrix pursuit for low rank matrix completion" *http://arxiv.org/abs/1404.1377*, 2014.

[22] R. O. Wells, "Differential Analysis on Complex Manifolds," Springer, New York, 2008.

[23] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a non-linear successive over-relaxation algorithm," *Math. Progr. Comput.*, vol. 4, pp. 333–361, 2012.