Low-rank optimization with trace norm penalty

Bamdev Mishra, Gilles Meyer, Francis Bach, Rodolphe Sepulchre

SIAM J. OPTIM.
Vol. 23, No. 4, pp. 2124–2149

LOW-RANK OPTIMIZATION WITH TRACE NORM PENALTY*

B. MISHRA†, G. MEYER†, F. BACH‡, AND R. SEPULCHRE§

Abstract. The paper addresses the problem of low-rank trace norm minimization. We propose an algorithm that alternates between fixed-rank optimization and rank-one updates. The fixed-rank optimization is characterized by an efficient factorization that makes the trace norm differentiable in the search space and the computation of duality gap numerically tractable. The search space is nonlinear but is equipped with a Riemannian structure that leads to efficient computations. We present a second-order trust-region algorithm with a guaranteed quadratic rate of convergence. Overall, the proposed optimization scheme converges superlinearly to the global solution while maintaining complexity that is linear in the number of rows and columns of the matrix. To compute a set of solutions efficiently for a grid of regularization parameters we propose a predictor-corrector approach that outperforms the naive warm-restart approach on the fixed-rank quotient manifold. The performance of the proposed algorithm is illustrated on problems of low-rank matrix completion and multivariate linear regression.

Key words. trace norm, Riemannian optimization, trust region, regularization path, predictor-corrector on quotient manifold, matrix completion, multivariate linear regression

AMS subject classifications. 65K05, 90C30, 90C22, 90C25, 90C26, 90C27, 90C46, 58C05, 49M15

DOI. 10.1137/110859646

1. Introduction.

The present paper focuses on the convex program

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times m}} \quad f(\mathbf{X}) + \lambda\|\mathbf{X}\|_*, \tag{1.1}$$

where f is a smooth convex function, $\|\mathbf{X}\|_*$ is the *trace norm* (also known as nuclear norm) which is the sum of the singular values of $\mathbf{X}$ [16, 35, 13], and $\lambda > 0$ is the regularization parameter. Programs of this type have attracted much attention in recent years as efficient convex relaxations of intractable rank minimization problems [16]. The rank of the optimal solution $\mathbf{X}^*(\lambda)$ of (1.1) decreases to zero as the regularization parameter grows unbounded [3]. As a consequence, efficiently generating the regularization path $\{\mathbf{X}^*(\lambda_i)\}_{i=1,\dots,N}$, for a whole range of values of λ_i minimizers is a convenient proxy for obtaining suboptimal low-rank minimizers of f.

Motivated by machine learning and statistical large-scale regression problems [35, 43, 41, 25, 39, 32], we are interested in very low-rank solutions ($p < 10^2$) of very high-dimensional problems ($n > 10^6$). To this end, we propose an algorithm that guarantees second-order convergence to the solutions of (1.1) while ensuring a tight control on the data storage requirements (linear in n) and on the numerical complexity of each iteration.

*Received by the editors December 19, 2011; accepted for publication (in revised form) July 22, 2013; published electronically November 5, 2013. 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 State, Science Policy Office. The scientific responsibility rests with its authors. Bamdev Mishra is a research fellow of the Belgian National Fund for Scientific Research (FNRS).

http://www.siam.org/journals/siopt/23-4/85964.html

†Department of Electrical Engineering and Computer Science, University of Liège, 4000 Liège, Belgium (B.Mishra@ulg.ac.be, G.Meyer@ulg.ac.be).

‡INRIA - Sierra Project-Team Ecole Normale Supérieure Paris, France (Francis.Bach@inria.fr).

§University of Cambridge, Department of Engineering, Trumpington Street, Cambridge CB2 1PZ, UK (r.sepulchre@eng.cam.ac.uk).

The proposed algorithm is based on a low-rank factorization of the unknown matrix, similar to the singular value decomposition (SVD), $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$. Like in SVD, $\mathbf{U} \in \mathbb{R}^{n\times p}$ and $\mathbf{V} \in \mathbb{R}^{m\times p}$ are orthonormal matrices that span row and column spaces of $\mathbf{X}$. In contrast, the $p \times p$ scaling factor $\mathbf{B} = \mathbf{B}^T \succ 0$ is allowed to be nondiagonal which makes the factorization nonunique. Our algorithm alternates between fixed-rank optimization and rank-one updates. When the rank is fixed, the problem is no longer convex but the search space has a Riemannian structure. We use the framework of optimization on Riemannian quotient manifolds to propose a trust-region algorithm that generates low-cost (linear in n) iterates that converge superlinearly to a local minimum. Local minima are escaped by incrementing the rank until the global minimum in reached. The rank-one update is always selected to ensure a decrease of the cost.

Implementing the complete algorithm for a fixed value of the regularization parameter λ leads to a monotone convergence to the global minimum through a sequence of local minima of increasing ranks. Instead, we also modify λ along the way with a predictor-corrector method thereby transforming most local minima of (1.1) (for fixed λ and fixed rank) into global minima of (1.1) for different values of λ. The resulting procedure, thus, provides a full regularization path at a very efficient numerical cost.

Not surprisingly, the proposed approach has links with several earlier contributions in the literature. Primarily, the idea of interlacing fixed-rank optimization with rank-one updates has been used in semidefinite programming [12, 19]. It is here extended to a nonsymmetric framework using the Riemannian geometry recently developed in [8, 28, 30]. An improvement with respect to the earlier work [12, 19] is the use of a duality gap certificate to discriminate between local and global minima and its efficient computation thanks to the chosen parameterization.

Schemes that combine fixed-rank optimization and special rank-one updates have appeared recently in the particular context of matrix completion [21, 42]. The framework presented here is in the same spirit but in a more general setting and with a global convergence analysis. Most other fixed-rank algorithms [38, 21, 27, 36, 42, 28, 9, 40] for matrix completion are first-order schemes. It is more difficult to provide a tight comparison of the proposed algorithm to trace norm minimization algorithms that do not fix the rank a priori [13, 26, 43, 2]. It should be emphasized, however, that most trace norm minimization algorithms use a singular value thresholding (SVT) operation at each iteration. This is the most numerically demanding step for these algorithms. For the matrix completion application, it involves computing (potentially all) the singular values of a *low-rank + sparse* matrix [13]. In contrast, the proposed approach requires only dense linear algebra (linear in n) and rank-one updates using only dominant singular vectors and value of a sparse matrix. The main potential of the algorithm appears when computing the solution not for a single parameter λ but for a number of values of λ. We compute the entire regularization path with an efficient predictor-corrector strategy that convincingly outperforms the warm-restart strategy.

The paper is organized as follows. In section 2 the problem of fixed-rank optimization is related to the trace norm minimization problem. Section 3 proposes a Riemannian second-order geometry for the fixed-rank problem with a detailed numerical complexity analysis. An algorithm for (1.1) that alternates between fixed-rank optimization and rank-one updates is proposed in section 4. A novel predictor-corrector approach to generate the regularization path of (1.1) for a grid of values of λ is discussed in section 5. For the sake of illustration and empirical comparison with state-of-the-art algorithms we consider two particular applications, low-rank matrix completion [14] and multivariate linear regression [43], in section 6. In both cases, we

obtain iterative algorithms with a numerical complexity that is linear in the number of observations and with favorable convergence and precision properties.

2. Relationship between convex program and nonconvex formulation. Among the different factorizations that exist to represent low-rank matrices, we use the factorization [30, 8] that decomposes a rank-p matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ into

$$\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T, \tag{2.1}$$

where $\mathbf{U} \in \mathrm{St}(p,n)$, $\mathbf{V} \in \mathrm{St}(p,m)$, and $\mathbf{B} \in S_{++}(p)$. $\mathrm{St}(p,n)$ is the Stiefel manifold or the set of $n \times p$ matrices with orthonormal columns. $S_{++}(p)$ is the cone of $p \times p$ positive definite matrices. We stress that the scaling $\mathbf{B} = \mathbf{B}^T \succ 0$ is not required to be diagonal. The redundancy of this parameterization has nontrivial algorithmic implications (in section 3) but we believe that it is also the key to success of the approach. Refer to [21, 30] for earlier algorithms advocating matrix scaling and section 6.1 for a numerical illustration. With the factorization $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$, the trace norm is $\|\mathbf{X}\|_* = \mathrm{Trace}(\mathbf{B})$ which is now differentiable. For a fixed rank p, the optimization problem (1.1) is recast as

$$\begin{array}{ll} \min\limits_{\mathbf{U},\mathbf{B},\mathbf{V}} & f(\mathbf{U}\mathbf{B}\mathbf{V}^T) + \lambda \mathrm{Trace}(\mathbf{B}) \\ \text{subject to} & \mathbf{U} \in \mathrm{St}(p,n), \quad \mathbf{B} \in S_{++}(p), \quad \text{and} \quad \mathbf{V} \in \mathrm{St}(p,m). \end{array} \tag{2.2}$$

The search space of (2.2) is not Euclidean but the product space of two well-studied manifolds, namely, the Stiefel manifold $\mathrm{St}(p,n)$ [15] and the cone of positive definite matrices $S_{++}(p)$ [37, 7]. The column and row spaces of $\mathbf{X}$ are represented on $\mathrm{St}(p,n)$ and $\mathrm{St}(p,m)$ whereas the scaling factor is absorbed into $S_{++}(p)$. A proper metric on the space takes into account both rotational and scaling invariance.

2.1. First-order optimality conditions. In order to relate the fixed-rank problem (2.2) to the convex optimization problem (1.1) we look at the necessary and sufficient optimality conditions that govern the solutions. The first-order necessary and sufficient optimality condition for the convex program (1.1) is [3, 35]

$$\mathbf{0} \in \mathrm{Grad} f(\mathbf{X}) + \lambda \partial \|\mathbf{X}\|_*, \tag{2.3}$$

where $\mathrm{Grad} f(\mathbf{X})$ is the Euclidean gradient of f at $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\partial\|\mathbf{X}\|_*$ is the subdifferential of the trace norm [13, 10].

PROPOSITION 2.1. *The first-order necessary optimality conditions of* (2.2) *are*

$$\begin{aligned} \mathbf{S}\mathbf{V}\mathbf{B} - \mathbf{U}\mathrm{Sym}(\mathbf{U}^T\mathbf{S}\mathbf{V}\mathbf{B}) &= \mathbf{0}, \\ \mathrm{Sym}(\mathbf{U}^T\mathbf{S}\mathbf{V} + \lambda\mathbf{I}) &= \mathbf{0}, \\ \mathbf{S}^T\mathbf{U}\mathbf{B} - \mathbf{V}\mathrm{Sym}(\mathbf{V}^T\mathbf{S}^T\mathbf{U}\mathbf{B}) &= \mathbf{0}, \end{aligned} \tag{2.4}$$

where $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ (2.1), $\mathrm{Sym}(\mathbf{\Delta}) = (\mathbf{\Delta} + \mathbf{\Delta}^T)/2$ *for any square matrix* $\mathbf{\Delta}$, *and* $\mathbf{S} = \mathrm{Grad} f(\mathbf{U}\mathbf{B}\mathbf{V}^T)$. $\mathbf{S}$ *is referred to as the* dual variable *throughout the paper.*

Proof. The first-order optimality conditions are derived either by writing the Lagrangian of the problem (2.2) and looking at the *KKT conditions* or by deriving the Riemannian gradient of the function on the product space $\mathrm{St}(p,n)\times S_{++}(p)\times \mathrm{St}(p,m)$ with the metric (3.6) proposed in section 3. The proof is given in Appendix A.1. ∎

PROPOSITION 2.2. *A local minimum of* (2.2) $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ *is also the global optimum of* (1.1) *iff* $\|\mathbf{S}\|_{op} = \lambda$ *where* $\mathbf{S} = \mathrm{Grad} f(\mathbf{U}\mathbf{B}\mathbf{V}^T)$ *and* $\|\mathbf{S}\|_{op}$ *is the operator norm, i.e., the dominant singular value of* $\mathbf{S}$. *Moreover,* $\|\mathbf{S}\|_{op} \geq \lambda$ *and equality holds*

only at optimality. Consequently, a local minimum of (2.2) *is identified with the global minimum of* (1.1) *if* $\|\mathbf{S}\|_{op} - \lambda \leq \epsilon$*, where* ϵ *is a user-defined threshold.*

Proof. This is in fact rewriting the first-order optimality condition of (1.1) [13, 25]. The proof is given in Appendix A.2. ∎

2.2. Duality gap computation. Proposition 2.2 provides a criterion to check the global optimality of a solution of (2.2). Here however, it provides no guarantees on *closeness* to the global solution. A better way of certifying optimality for the optimization problem (1.1) is provided by the duality gap. The duality gap characterizes the difference of the obtained solution from the optimal solution and is always nonnegative [10].

PROPOSITION 2.3. *The Lagrangian dual formulation of* (1.1) *is*

$$\begin{array}{ll} \max\limits_{\mathbf{M}} & -f^*(\mathbf{M}) \\ \text{subject to} & \|\mathbf{M}\|_{op} \leq \lambda, \end{array} \tag{2.5}$$

where $\mathbf{M} \in \mathbb{R}^{n\times m}$ *is the dual variable,* $\|\mathbf{M}\|_{op}$ *is the largest singular value of* $\mathbf{M}$ *and is the dual norm of the trace norm.* f^* *is the Fenchel (convex) conjugate* [4, 10] *of* f*, defined as* $f^*(\mathbf{M}) = \sup_{\mathbf{X}\in\mathbb{R}^{n\times m}}[\mathrm{Trace}(\mathbf{M}^T\mathbf{X}) - f(\mathbf{X})]$.

Proof. The proof is given in Appendix A.4. ∎

When $\|\mathbf{M}\|_{op} \leq \lambda$, the expression of duality gap is

$$f(\mathbf{X}) + \lambda\|\mathbf{X}\|_* + f^*(\mathbf{M}), \tag{2.6}$$

where $\mathbf{M}$ is the *dual candidate*. A good choice for the dual candidate $\mathbf{M}$ is $\mathbf{S}$ (= $\mathrm{Grad}f(\mathbf{X})$) with appropriate scaling to satisfy the operator norm constraint: $\mathbf{M} = \min(1, \frac{\lambda}{\|\mathbf{S}\|_{op}})\mathbf{S}$ [4].

3. A Riemannian optimization approach for (2.2). In this section we propose an algorithm for the problem (2.2). In contrast to first-order optimization algorithms proposed earlier in [30, 29, 21], we develop a second-order (trust-region) algorithm that has a provably quadratic rate of convergence [1]. We rewrite (2.2) as

$$\begin{array}{ll} \min\limits_{\mathbf{U},\mathbf{B},\mathbf{V}} & \bar{\phi}(\mathbf{U},\mathbf{B},\mathbf{V}) \\ \text{subject to} & (\mathbf{U},\mathbf{B},\mathbf{V}) \in \mathrm{St}(p,n) \times S_{++}(p) \times \mathrm{St}(p,m), \end{array} \tag{3.1}$$

where $\bar{\phi}(\mathbf{U},\mathbf{B},\mathbf{V}) = f(\mathbf{U}\mathbf{B}\mathbf{V}^T) + \lambda\mathrm{Trace}(\mathbf{B})$. The function $\bar{\phi} : \mathrm{St}(p,n) \times S_{++}(p) \times \mathrm{St}(p,m) \to \mathbb{R}$ is introduced for notational convenience. An important observation for second-order algorithms [1] is that the local minima of the problem (3.1) are not isolated in the search space

$$\overline{\mathcal{M}}_p = \mathrm{St}(p,n) \times S_{++}(p) \times \mathrm{St}(p,m).$$

This is because the cost function is invariant under rotations, $\mathbf{U}\mathbf{B}\mathbf{V}^T = (\mathbf{U}\mathbf{O})(\mathbf{O}^T\mathbf{B}\mathbf{O})(\mathbf{V}\mathbf{O})^T$ for any $p \times p$ rotation matrix $\mathbf{O} \in \mathcal{O}(p)$. To remove the symmetry of the cost function, we identify all the points of the search space that belong to the equivalence class defined by

$$[(\mathbf{U},\mathbf{B},\mathbf{V})] = \{(\mathbf{U}\mathbf{O}, \mathbf{O}^T\mathbf{B}\mathbf{O}, \mathbf{V}\mathbf{O}) : \mathbf{O} \in \mathcal{O}(p)\}. \tag{3.2}$$

The set of all such equivalence classes is denoted by

$$\mathcal{M}_p = \overline{\mathcal{M}}_p / \mathcal{O}(p) \tag{3.3}$$

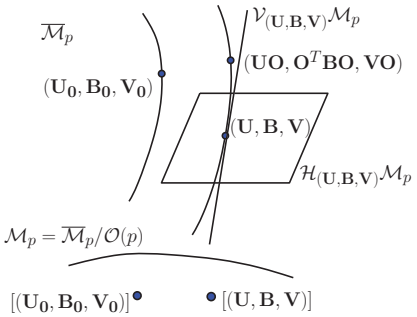


FIG. 3.1. *The quotient manifold representation of the search space.*

that has the structure of a smooth quotient manifold $\overline{\mathcal{M}}_p$ by $\mathcal{O}(p)$ [24, Theorem 9.16]. Note that $\mathcal{O}(p)$ takes away all the symmetry of the total space as the dimension of $\mathcal{M}_p$ is $(n+m-p)p$ which is equal to the dimension of the set of rank-p matrices. The dimension of the quotient manifold is obtained by subtracting the dimension of $\mathcal{O}(p)$ from the dimension of the product space $\overline{\mathcal{M}}_p$. Problem (3.1) is thus conceptually an *unconstrained* optimization problem on the quotient manifold $\mathcal{M}_p$ in which the minima are isolated. Computations are performed in the total space $\overline{\mathcal{M}}_p$, which is the product space of well-studied manifolds.

Tangent space of $\boldsymbol{\mathcal{M}_p}$. Tangent vectors at a point $x \in \mathcal{M}_p$ in the abstract quotient manifold have a matrix representation in the tangent space of the total space $\overline{\mathcal{M}}_p$ that respects the equivalence relationship (3.2). Figure 3.1 gives a graphical description of the search space and the matrix representations. Note that $\bar{x}$ belongs to $\overline{\mathcal{M}}_p$ and its equivalence class is represented by the element $x \in \mathcal{M}_p$ such that $x = [\bar{x}]$. Because the total space is a product space $\mathrm{St}(p,n) \times S_{++}(p) \times \mathrm{St}(p,m)$, its tangent space admits the decomposition at a point $\bar{x} = (\mathbf{U}, \mathbf{B}, \mathbf{V})$:

$$T_{\bar{x}}\overline{\mathcal{M}}_p = T_{\mathbf{U}}\mathrm{St}(p,n) \times T_{\mathbf{B}}S_{++}(p) \times T_{\mathbf{V}}\mathrm{St}(p,m)$$

and the following characterizations of $\mathrm{St}(p,n)$ and $S_{++}(p)$ are well known [15, 37]:

$$T_{\mathbf{U}}\mathrm{St}(p,n) = \{\mathbf{Z_U} - \mathbf{U}\mathrm{Sym}(\mathbf{U}^T\mathbf{Z_U}) : \mathbf{Z_U} \in \mathbb{R}^{n\times p}\},$$
$$T_{\mathbf{B}}S_{++}(p) = \{\mathbf{Z_B} = \mathbf{Z_B}^T : \mathbf{Z_B} \in \mathbb{R}^{p\times p}\},$$

where $\mathrm{Sym}(\cdot)$ extracts the symmetric part of a square matrix, i.e., $\mathrm{Sym}(\boldsymbol{\Delta}) = (\boldsymbol{\Delta} + \boldsymbol{\Delta}^T)/2$. Note that an arbitrary matrix $(\mathbf{Z_U}, \mathbf{Z_B}, \mathbf{Z_V}) \in \mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p} \times \mathbb{R}^{m\times p}$ is projected on the tangent space $T_{\bar{x}}\overline{\mathcal{M}}_p$ by the linear operation

(3.4)
$$\Psi_{\bar{x}}(\mathbf{Z_U}, \mathbf{Z_B}, \mathbf{Z_V}) = (\mathbf{Z_U} - \mathbf{U}\mathrm{Sym}(\mathbf{U}^T\mathbf{Z_U}), \mathrm{Sym}(\mathbf{Z_B}), \mathbf{Z_V} - \mathbf{V}\mathrm{Sym}(\mathbf{V}^T\mathbf{Z_V})),$$

where $\mathrm{Sym}(\mathbf{Z_B}) = (\mathbf{Z_B} + \mathbf{Z_B}^T)/2$. A matrix representation of the tangent space at $x \in \mathcal{M}_p$ relies on the decomposition of $T_{\bar{x}}\overline{\mathcal{M}}_p$ into its *vertical* and *horizontal* subspaces. The vertical space $\mathcal{V}_{\bar{x}}\overline{\mathcal{M}}_p$ is the subspace of $T_{\bar{x}}\overline{\mathcal{M}}_p$ that is tangent to the equivalence class $[\bar{x}]$,

$$\mathcal{V}_{\bar{x}}\overline{\mathcal{M}}_p = \{(\mathbf{U}\boldsymbol{\Omega}, \mathbf{B}\boldsymbol{\Omega} - \boldsymbol{\Omega}\mathbf{B}, \mathbf{V}\boldsymbol{\Omega}) : \boldsymbol{\Omega} \in S_{skew}(p)\}, \tag{3.5}$$

where $S_{skew}(p)$ is the set of skew symmetric matrices of size $p \times p$. The horizontal space $\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$ must be chosen such that $T_{\bar{x}}\overline{\mathcal{M}}_p = \mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p \oplus \mathcal{V}_{\bar{x}}\overline{\mathcal{M}}_p$. We choose $\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$

as the orthogonal complement of $\mathcal{V}_{\bar{x}}\overline{\mathcal{M}}_p$ for the metric

$$\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\eta}_{\bar{x}}) = \mathrm{Trace}(\bar{\xi}_{\mathbf{U}}^T \bar{\eta}_{\mathbf{U}}) + \mathrm{Trace}(\mathbf{B}^{-1}\bar{\xi}_{\mathbf{B}}\mathbf{B}^{-1}\bar{\eta}_{\mathbf{B}}) + \mathrm{Trace}(\bar{\xi}_{\mathbf{V}}^T \bar{\eta}_{\mathbf{V}}), \tag{3.6}$$

which picks the normal metric of the Stiefel manifold [15] and the natural metric of the positive definite cone [37, 7]. Here $\bar{\xi}_{\bar{x}}$ and $\bar{\eta}_{\bar{x}}$ are elements of $T_{\bar{x}}\mathcal{M}_p$. With this choice, a horizontal tangent vector $\bar{\zeta}_{\bar{x}}$ is any tangent vector belonging to the set

$$\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p = \{(\bar{\zeta}_{\mathbf{U}}, \bar{\zeta}_{\mathbf{B}}, \bar{\zeta}_{\mathbf{V}}) \in T_{\bar{x}}\overline{\mathcal{M}}_p : \bar{g}_{\bar{x}}((\bar{\zeta}_{\mathbf{U}}, \bar{\zeta}_{\mathbf{B}}, \bar{\zeta}_{\mathbf{V}}), (\mathbf{U}\boldsymbol{\Omega}, (\mathbf{B}\boldsymbol{\Omega} - \boldsymbol{\Omega}\mathbf{B}), \mathbf{V}\boldsymbol{\Omega})) = 0\} \tag{3.7}$$

for all $\boldsymbol{\Omega} \in S_{skew}(p)$. Another characterization of the horizontal space is $\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p = \{(\bar{\zeta}_{\mathbf{U}}, \bar{\zeta}_{\mathbf{B}}, \bar{\zeta}_{\mathbf{V}}) \in T_{\bar{x}}\overline{\mathcal{M}}_p : (\bar{\zeta}_{\mathbf{U}}^T\mathbf{U} + \mathbf{B}^{-1}\bar{\zeta}_{\mathbf{B}} - \bar{\zeta}_{\mathbf{B}}\mathbf{B}^{-1} + \bar{\zeta}_{\mathbf{V}}^T\mathbf{V})$ is symmetric$\}$. The horizontal space is invariant by the group action along the equivalence class. The horizontal space $\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$ in the total space $\overline{\mathcal{M}}_p$ provides a valid matrix representation of the abstract tangent space $T_x\mathcal{M}_p$ of the quotient manifold $\mathcal{M}_p$ [1, section 3.5.8]. The tangent vector $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$ is called the *horizontal lift* of $\xi_x \in T_x\mathcal{M}_p$ at $\bar{x}$. Starting from an arbitrary tangent vector $\bar{\eta}_{\bar{x}} \in T_{\bar{x}}\overline{\mathcal{M}}_p$ we construct its projection on the horizontal space, i.e.,

$$\Pi_{\bar{x}}(\bar{\eta}_{\bar{x}}) = (\bar{\eta}_{\mathbf{U}} - \mathbf{U}\boldsymbol{\Omega}, \bar{\eta}_{\mathbf{B}} - (\mathbf{B}\boldsymbol{\Omega} - \boldsymbol{\Omega}\mathbf{B}), \bar{\eta}_{\mathbf{V}} - \mathbf{V}\boldsymbol{\Omega}) \in \mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p, \tag{3.8}$$

by picking $\boldsymbol{\Omega} \in S_{skew}(p)$ such that it satisfies (3.7) and equivalently, it is the unique solution of the Lyapunov equation

$$\boldsymbol{\Omega}\mathbf{B}^2 + \mathbf{B}^2\boldsymbol{\Omega} = \mathbf{B}(\mathrm{Skew}(\mathbf{U}^T\bar{\eta}_{\mathbf{U}}) - 2\mathrm{Skew}(\mathbf{B}^{-1}\bar{\eta}_{\mathbf{B}}) + \mathrm{Skew}(\mathbf{V}^T\bar{\eta}_{\mathbf{V}}))\mathbf{B}, \tag{3.9}$$

where $\mathrm{Skew}(\cdot)$ extracts the skew-symmetric of a square matrix, i.e., $\mathrm{Skew}(\boldsymbol{\Delta}) = (\boldsymbol{\Delta} - \boldsymbol{\Delta}^T)/2$ and $(\bar{\eta}_{\mathbf{U}}, \bar{\eta}_{\mathbf{B}}, \bar{\eta}_{\mathbf{V}})$ is the matrix representation of $\bar{\eta}_{\bar{x}}$. The numerical complexity of solving the above Lyapunov equation is $O(p^3)$ [6].

The Riemannian submersion $(\mathcal{M}_p, g)$. The choice of the metric (3.6), which is invariant along the equivalence class $[\bar{x}]$ turns the quotient manifold $\mathcal{M}_p$ into a Riemannian submersion of $(\overline{\mathcal{M}}_p, \bar{g})$ [24, Theorem 9.16] and [1, section 3.6.2]. As shown in [1], this special construction allows for a convenient matrix representation of the Riemannian gradient [1, section 3.6.2] and the Riemannian Hessian [1, Proposition 5.3.3] on the abstract manifold $\mathcal{M}_p$. The Riemannian gradient of $\phi : \mathcal{M}_p \to \mathbb{R} : x \mapsto \phi(x) = \bar{\phi}(\bar{x})$ is uniquely represented by its horizontal lift in $\overline{\mathcal{M}}_p$ which has the matrix representation

$$\overline{\mathrm{grad}_x \phi} = \mathrm{grad}_{\bar{x}} \bar{\phi}. \tag{3.10}$$

It should be emphasized that $\mathrm{grad}_{\bar{x}}\bar{\phi}$ is in the the tangent space $T_{\bar{x}}\overline{\mathcal{M}}_p$. However, due to invariance of the cost along the equivalence class $[\bar{x}]$, $\mathrm{grad}_{\bar{x}}\bar{\phi}$ also belongs to the horizontal space $\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$ and hence, the equality in (3.10) [1, section 3.6.2]. The matrix expression of $\mathrm{grad}_{\bar{x}}\bar{\phi}$ in the total space $\overline{\mathcal{M}}_p$ at a point $\bar{x} = (\mathbf{U}, \mathbf{B}, \mathbf{V})$ is obtained from its definition: it is the unique element of $T_{\bar{x}}\overline{\mathcal{M}}_p$ that satisfies $\mathrm{D}\bar{\phi}[\bar{\eta}_{\bar{x}}] = \bar{g}_{\bar{x}}(\mathrm{grad}_{\bar{x}}\bar{\phi}, \bar{\eta}_{\bar{x}})$ for all $\bar{\eta}_{\bar{x}} \in T_{\bar{x}}\overline{\mathcal{M}}_p$. $\mathrm{D}\bar{\phi}[\bar{\eta}_{\bar{x}}]$ is the standard Euclidean directional derivative of $\bar{\phi}$ in the direction $\bar{\eta}_{\bar{x}}$, i.e.,

$$\bar{g}_{\bar{x}}(\mathrm{grad}_{\bar{x}}\bar{\phi}, \bar{\eta}_{\bar{x}}) = \mathrm{D}\bar{\phi}[\bar{\eta}_{\bar{x}}] = \lim_{t \downarrow 0_+} \frac{\bar{\phi}(\bar{x} + t\bar{\eta}_{\bar{x}}) - \bar{\phi}(\bar{x})}{t} \quad \forall \bar{\eta}_{\bar{x}} \in T_{\bar{x}}\overline{\mathcal{M}}_p.$$

This definition leads to the matrix representation

$$\begin{aligned} &\operatorname{grad}_{\mathbf{U}}\bar{\phi} = \operatorname{grad}_{\mathbf{U}}\bar{\phi}_{\mathrm{E}},\ \operatorname{grad}_{\mathbf{B}}\bar{\phi} = \mathbf{B}\left(\operatorname{grad}_{\mathbf{B}}\bar{\phi}_{\mathrm{E}}\right)\mathbf{B}, \\ &\operatorname{grad}_{\mathbf{V}}\bar{\phi} = \operatorname{grad}_{\mathbf{V}}\bar{\phi}_{\mathrm{E}}, \end{aligned} \tag{3.11}$$

where $\operatorname{grad}_{\bar{x}}\bar{\phi}_{\mathrm{E}} = \Psi_{\bar{x}}(\partial\bar{\phi}(\bar{x})/\partial\bar{x})$, i.e., the projection of the partial derivatives $\partial\bar{\phi}(\bar{x})/\partial\bar{x}$ on the tangent space $T_{\bar{x}}\overline{\mathcal{M}}_p$. Here $\partial\bar{\phi}(\bar{x})/\partial\bar{x}$ is the matrix representation of the partial derivatives of $\bar{\phi}$ with respect to $(\mathbf{U},\mathbf{B},\mathbf{V})$ in the Euclidean space $\mathbb{R}^{n\times r}\times\mathbb{R}^{r\times r}\times\mathbb{R}^{m\times r}$. And $\Psi_{\bar{x}}(\cdot)$ is the operator (3.4) that projects onto the tangent space $T_{\bar{x}}\overline{\mathcal{M}}_p$.

Likewise, the Riemannian connection $\nabla_{\nu_x}\eta_x$ on quotient manifold $\mathcal{M}_p$ is uniquely represented by its horizontal lift in $\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$ which is [1, Proposition 5.3.3]

$$\overline{\nabla_{\nu_x}\eta_x} = \Pi_{\bar{x}}(\overline{\nabla}_{\bar{\nu}_{\bar{x}}}\bar{\eta}_{\bar{x}}), \tag{3.12}$$

where ν_x and η_x are vector fields on the quotient manifold $\mathcal{M}_p$ and $\bar{\nu}_{\bar{x}}$ and $\bar{\eta}_{\bar{x}}$ are their horizontal lifts in $\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$. Once again, the Riemannian connection $\overline{\nabla}_{\bar{\nu}_{\bar{x}}}\bar{\eta}_{\bar{x}}$ on $\overline{\mathcal{M}}_p$ has well-known expression as a result of the individual Riemannian connection characterization on $\mathrm{St}(p,n)$ in [18, 1] and on $S_{++}(p)$ in [37, 7]. The Riemannian connection on the Stiefel manifold is derived in [18, Example 4.3.6] and on the positive definite cone is derived in [28, Appendix B]. Finally, the Riemannian connection on the total space is given by the product structure

$$\begin{aligned} \overline{\nabla}_{\bar{\nu}_{\bar{x}}}\bar{\eta}_{\bar{x}} = \Psi_{\bar{x}}(\mathrm{D}\bar{\eta}_{\bar{x}}[\bar{\nu}_{\bar{x}}]) - \ \Psi_{\bar{x}}(&\nu_{\mathbf{U}}\mathrm{Sym}(\mathbf{U}^T\eta_{\mathbf{U}}), \mathrm{Sym}(\nu_{\mathbf{B}}\mathbf{B}^{-1}\eta_{\mathbf{B}}), \nu_{\mathbf{V}}\mathrm{Sym}(\mathbf{V}^T\eta_{\mathbf{V}})). \end{aligned} \tag{3.13}$$

Here $\mathrm{D}\bar{\eta}_{\bar{x}}[\bar{\nu}_{\bar{x}}]$ is the standard Euclidean directional derivative of $\bar{\eta}_{\bar{x}}$ in the direction $\bar{\nu}_{\bar{x}}$. The Riemannian Hessian $\operatorname{Hess}_x\phi[\xi_x]$ of ϕ is the covariant derivative of the Riemannian gradient $\operatorname{grad}_x\phi$ in the direction $\xi_x \in T_x\mathcal{M}_p$. Its horizontal lift, from (3.12), in $\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$ has the matrix expression

$$\overline{\operatorname{Hess}_x\phi[\xi]} = \Pi_{\bar{x}}(\overline{\nabla}_{\bar{\xi}}\overline{\operatorname{grad}_x\phi}) \tag{3.14}$$

for any $\xi_x \in T_x\mathcal{M}_p$ and its horizontal lift $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$.

Trust-region subproblem and retraction on $\overline{\mathcal{M}}_p$. Trust-region algorithms on a quotient manifold with guaranteed quadratic rate convergence have been proposed in [1, Algorithm 10]. The convergence of the trust-region algorithm is quadratic because the assumptions [1, Theorem 7.4.11] are satisfied locally. The trust-region subproblem on the quotient manifold $\mathcal{M}$ is horizontally lifted to the horizontal space $\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$ and is formulated as

$$\begin{aligned} \min_{\bar{\xi}_{\bar{x}}\in\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p} \quad & \bar{\phi}(\bar{x}) + \bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \overline{\operatorname{grad}_x\phi}) + \tfrac{1}{2}\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \overline{\operatorname{Hess}_x\phi[\xi_x]}) \\ \text{subject to} \quad & \bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\xi}_{\bar{x}}) \leq \delta^2, \end{aligned} \tag{3.15}$$

where δ is the trust-region radius and $\operatorname{grad}_x\phi$ and $\operatorname{Hess}_x\phi[\xi_x]$ are the Riemannian gradient and Hessian on $\mathcal{M}_p$. In particular, we implement the Riemannian trust-region algorithm [1, Algorithm 10] using the generic solver GenRTR [5]. The subproblem (3.15) is solved using a *truncated-conjugate gradient* method with parameters set as in [1, Algorithm 11]. This leads to a direction $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$ that minimizes the quadratic model. To find a new iterate based on the obtained direction $\bar{\xi}_{\bar{x}}$, a mapping from the horizontal space $\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$ to the manifold $\overline{\mathcal{M}}_p$ is required. This mapping is more generally referred to as *retraction* which maps the vectors from the horizontal space

$\mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p$ onto $\overline{\mathcal{M}}_p$, $R_{\bar{x}} : \mathcal{H}_{\bar{x}}\overline{\mathcal{M}}_p \to \overline{\mathcal{M}}_p$ [1, Definition 4.1.1]. In the present case, a retraction of interest is [1, 30]

$$R_{\bar{x}}(\bar{\xi}) = (\mathrm{uf}(\mathbf{U} + \xi_{\mathbf{U}}), \mathbf{B}^{\frac{1}{2}}\mathrm{expm}(\mathbf{B}^{-\frac{1}{2}}\xi_{\mathbf{B}}\mathbf{B}^{-\frac{1}{2}})\mathbf{B}^{\frac{1}{2}}, \mathrm{uf}(\mathbf{V} + \xi_{\mathbf{V}})), \tag{3.16}$$

where $\mathrm{uf}(\cdot)$ extracts the orthogonal factor of the polar factorization, i.e., $\mathrm{uf}(\mathbf{A}) = \mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1/2}$ and $\mathrm{expm}(\cdot)$ is the *matrix exponential* operator. The retraction on the positive definite cone is the natural exponential mapping for the metric (3.6) [37]. The combination of these well-known retractions on the individual manifolds is also a valid retraction on the quotient manifold $\mathcal{M}_p$ by virtue of [1, Proposition 4.1.3].

Numerical complexity. The numerical complexity per iteration of the proposed trust-region algorithm for (3.1) depends on the computational cost of the following items.

- Objective function $\bar{\phi}(\bar{x})$: problem dependent.
- Metric $\bar{g}$ 3.6: $O(np^2 + mp^2 + p^3)$.
- Projection operator $\Psi_{\bar{x}}(\cdot)$ (3.4): $O(np^2 + mp^2)$.
- Projection operator $\Pi_{\bar{x}}(\cdot)$ (3.8): $O(np^2 + mp^2 + p^3)$:
  - Solving the Lyapunov equation for $\mathbf{\Omega}$ (3.9): $O(p^3)$.
- Retraction $R_{\bar{x}}(\cdot)$ (3.16): $O(np^2 + mp^2 + p^3)$.
- The Euclidean gradient $\mathrm{grad}_{\bar{x}}\bar{\phi}_{\mathrm{E}}$: problem dependent.
- $\overline{\nabla}_{\bar{\nu}}\bar{\eta}$ (3.13):
  - The Euclidean directional derivative $\mathrm{D}\bar{\eta}[\bar{\nu}]$: problem dependent;
  - $\nu_{\mathbf{U}}\mathrm{Sym}(\mathbf{U}^T\eta_{\mathbf{U}})$: $O(np^2)$;
  - $\mathrm{Sym}(\nu_{\mathbf{B}}\mathbf{B}^{-1}\eta_{\mathbf{B}})$: $O(p^3)$;
  - $\nu_{\mathbf{V}}\mathrm{Sym}(\mathbf{V}^T\eta_{\mathbf{V}})$: $O(mp^2)$.

As shown above all the manifold related operations have linear complexity in n and m. Other operations depend on the problem at hand and are computed in the search space $\overline{\mathcal{M}}_p$. With $p \ll \min\{n, m\}$ the computational burden on the algorithm considerably reduces.

4. An optimization scheme for (1.1). Starting with a rank-1 problem, we alternate a second-order local optimization algorithm on fixed-rank manifold with a first-order rank-one update. The scheme is shown in Table 4.1. The rank-one update decreases the cost with the updated iterate in $\overline{\mathcal{M}}_{p+1}$.

PROPOSITION 4.1. *If* $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ *is a stationary point of* (2.2) *then the rank-one update*

$$\mathbf{X}_+ = \mathbf{X} - \beta uv^T \tag{4.1}$$

TABLE 4.1
Algorithm to solve the trace norm minimization problem (1.1).

ALGORITHM TO SOLVE CONVEX PROBLEM (1.1).

0.
   - Initialize p to p_0, a guess rank.
   - Initialize the threshold ϵ for convergence criterion; refer to Proposition 2.2.
   - Initialize the iterates $\mathbf{U}_0 \in \mathrm{St}(p_0, n)$, $\mathbf{B}_0 \in S_{++}(p_0)$, and $\mathbf{V}_0 \in \mathrm{St}(p_0, m)$.
1. Solve the nonconvex problem (2.2) in the dimension p to obtain a local minimum $(\mathbf{U}, \mathbf{B}, \mathbf{V})$.
2. Compute σ_1 (the dominant singular value) of dual variable $\mathbf{S} = \mathrm{Grad}f(\mathbf{U}\mathbf{B}\mathbf{V}^T)$.
   - If $\sigma_1 - \lambda \leq \epsilon$ (or duality gap $\leq \epsilon$) due to Proposition 2.2, output $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ as the solution to problem (1.1) and stop.
   - Else, compute the update as shown in Proposition 4.1 and compute the new point $(\mathbf{U}_+, \mathbf{B}_+, \mathbf{V}_+)$ as described in (4.1). Set $p = p + 1$ and repeat step 1.

ensures a decrease in the objective function $f(\mathbf{X}) + \lambda\|\mathbf{X}\|_*$ *provided that* $\beta > 0$ *is sufficiently small and the descent directions* $u \in \mathbb{R}^n$ *and* $v \in \mathbb{R}^m$ *are the dominant left and right singular vectors of the dual variable* $\mathbf{S} = \mathrm{Grad} f(\mathbf{U}\mathbf{B}\mathbf{V}^T)$.

The maximum decrease in the objective function is obtained for $\beta = \frac{\sigma_1 - \lambda}{L_f}$, *where* σ_1 *is the dominant singular value of* $\mathbf{S}$ *and* L_f *is the Lipschitz constant for the first derivative of* f, *i.e.,* $\|\mathrm{Grad} f(\mathbf{X}) - \mathrm{Grad} f(\mathbf{Y})\|_F \leq L_f \|\mathbf{X} - \mathbf{Y}\|_F$ *for all* $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times m}$.

Proof. This is in fact a descent step as shown in [13, 25, 26] but now projected onto the rank-one dominant subspace. The proof is given in Appendix A.3. ∎

A representation of $\mathbf{X}_+ = \mathbf{X} - \beta u v^T$ on $\overline{\mathcal{M}}_{p+1}$ is obtained by performing the SVD of $\mathbf{X}_+$. Since $\mathbf{X}_+$ is a rank-one update of $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$, the SVD of $\mathbf{X}_+$ only requires $O(np^2 + mp^2 + p^3)$ operations [11]. Finally, to compute an Armijo optimal β we perform a *backtracking* line search starting from the value $\frac{\sigma_1 - \lambda}{L_f}$, where L_f is the Lipschitz constant for the first derivative of f [33]. The justification for this value is given in Appendix A.3. In many problem instances, it is easy to estimate L_f by randomly selecting two points, say $\mathbf{X}$ and $\mathbf{Y} \in \mathbb{R}^{n \times m}$, and computing $\|\mathrm{Grad} f(\mathbf{X}) - \mathrm{Grad} f(\mathbf{Y})\|_F / \|\mathbf{X} - \mathbf{Y}\|_F$ [33].

There is no theoretical guarantee that the algorithm in Table 4.1 stops at $p = p^*$ (the optimal rank). However, convergence to the global solution is guaranteed from the fact that the algorithm alternates between fixed-rank optimization and rank updates (unconstrained projected rank-1 gradient step) and both are descent iterates. Disregarding the fixed-rank step, the algorithm reduces to a gradient algorithm for a convex problem with classical global convergence guarantees. This theoretical certificate, however, does not capture the convergence properties of an algorithm that empirically always converges at a rank $p \ll \min(m, n)$ (most often at the optimal rank). One advantage of the scheme, in contrast to trace norm minimization algorithms proposed in [13, 39, 25, 26], is that it offers a tighter control over the rank at all intermediate iterates of the scheme. It should also be emphasized that the stopping criterion threshold of the nonconvex problem (2.2) and of the convex problem (1.1) are chosen separately. This means that rank increments can be made after a fixed number of iterations of the manifold optimization without waiting for the trust-region algorithm to converge to a local minimum.

5. Regularization path. In most applications, the optimal value of λ is unknown [26] which means that in fact problem (1.1) can be solved for a number of regularization parameters. In addition, even if the optimal λ is a priori known, a path of solutions corresponding to different values of λ provides interpretability to the intermediate iterates which are now global minima for different values of λ. This motivates us to compute the complete regularization path of (1.1) for a number of values λ, i.e., defined as $\mathbf{X}^*(\lambda_i) = \arg\min_{\mathbf{X} \in \mathbb{R}^{n \times m}} f(\mathbf{X}) + \lambda_i \|\mathbf{X}\|_*$, where $\mathbf{X}^*(\lambda_i)$ is the solution to the λ_i minimization problem.

A common approach is the *warm-restart* approach where the algorithm to solve the λ_{i+1} problem is initialized from $\mathbf{X}^*(\lambda_i)$ and so on [26]. However, the warm-restart approach does not use the fact that the regularization path is *smooth* especially when the values of λ are close to each other. An argument towards this is given later in the section. We propose a *predictor-corrector* scheme to compute the regularization path efficiently. We first take a *predictor* (estimator) step to predict the solution and then rectify the prediction by a *corrector* step. This scheme has been widely used in solving differential equations and regression problems [34]. We extend the *prediction* idea to the quotient manifold $\mathcal{M}_p$. The corrector step is carried out by initializing the algorithm in Table 4.1 from the predicted point. If $\mathbf{X}^*(\lambda_i) = \mathbf{U}_i \mathbf{B}_i \mathbf{V}_i{}^T$ is the fixed-

TABLE 5.1

Algorithm for computing the regularization path. If N is the number of values of λ and r is the number of rank increments then the scheme uses r warm restarts and $N - r$ predictor steps to compute the full path.

COMPUTING THE REGULARIZATION PATH.

0. Given $\{\lambda_i\}_{i=1,\dots,N}$ in decreasing order. Also given are the solutions $\mathbf{X}^*(\lambda_1)$ and $\mathbf{X}^*(\lambda_2)$ at λ_1 and λ_2, respectively, and their low-rank factorizations.
1. Predictor step:
   - If $\mathbf{X}^*(\lambda_{i-1})$ and $\mathbf{X}^*(\lambda_i)$ belong to the same quotient manifold $\mathcal{M}_p$ then construct a first-order approximation of the solution path at λ_i and estimate $\hat{\mathbf{X}}(\lambda_{i+1})$ as shown in (5.3).
   - Else $\hat{\mathbf{X}}(\lambda_{i+1}) = \mathbf{X}^*(\lambda_i)$.
2. Corrector step: Using the estimated solution of the λ_{i+1} − problem, initialize the algorithm described in Table 4.1 to compute the exact solution $\mathbf{X}^*(\lambda_{i+1})$.
3. Repeat steps 1 and 2 for all subsequent values of λ.

rank factorization (2.1) then the solution of the λ_{i+1} optimization problem is predicted (or estimated), i.e., $\hat{\mathbf{X}}(\lambda_{i+1}) = \hat{\mathbf{U}}_{i+1}\hat{\mathbf{B}}_{i+1}\hat{\mathbf{V}}_{i+1}^T$, by the two previous solutions $\mathbf{X}^*(\lambda_i)$ and $\mathbf{X}^*(\lambda_{i-1})$ at λ_i and λ_{i-1}, respectively, belonging to the same rank manifold $\mathcal{M}_p$. When $\mathbf{X}^*(\lambda_{i-1})$ and $\mathbf{X}^*(\lambda_i)$ belong to different rank manifolds we perform instead a warm restart to solve λ_{i+1} problem. The complete scheme is shown in Table 5.1 and has the following advantages.

- With a small number of rank increments we traverse the entire path.
- Potentially every iterate of the optimization scheme is now a global solution for a value of λ.
- The predictor-corrector approach outperforms the warm-restart approach in maximizing *prediction accuracy* with minimal extra computations.

In this section, we assume that the optimization problem (1.1) has a unique solution for all λ. A sufficient condition is that f is *strictly* convex, which can be enforced by regularizing f with the square Frobenius norm of $\mathbf{X}$.

In order to characterize smoothness of the regularization path we observe that the global solution $\mathbf{X}^*(\lambda) = \mathbf{U}\mathbf{B}\mathbf{V}^T$ is uniquely characterized by the nonlinear system of equations

$$\mathbf{S}\mathbf{V} = \lambda\mathbf{U},\ \mathbf{U}^T\mathbf{S}\mathbf{V} = \lambda\mathbf{I},\quad \text{and}\quad \mathbf{S}^T\mathbf{U} = \lambda\mathbf{V}$$

which is obtained from the optimality conditions (2.4) and Proposition 2.2. The smoothness of $\mathbf{X}^*(\lambda)$ with respect to λ follows the implicit function theorem [22]. A geometrical reasoning is followed by inspection of the dual formulation. Note that we employ the predictor-corrector step only when we are on the fixed-rank manifold which corresponds to a *face* of the dual operator norm set. From Proposition 2.3, the dual optimal solution is obtained by projection onto the dual set. Smoothness of the dual variable, say $\mathbf{M}^*(\lambda)$, with respect to λ follows from the smoothness of the projection operator [17]. Consequently, smoothness of the primal variable $\mathbf{X}^*(\lambda)$ follows from the smoothness assumption of f.

Predictor step on the quotient manifold $\mathcal{M}_p$. We first build a first-order approximation of the geodesic on $\mathcal{M}_p$ (and its representation on $\overline{\mathcal{M}}_p$), the curve of shortest length, connecting $\bar{x}_i = (\mathbf{U}_i, \mathbf{B}_i, \mathbf{V}_i)$ and $\bar{x}_{i-1} = (\mathbf{U}_{i-1}, \mathbf{B}_{i-1}, \mathbf{V}_{i-1})$ in $\overline{\mathcal{M}}_p$. The estimated solution $\hat{\mathbf{X}}(\lambda_{i+1})$ is then computed by extending the first-order approximation of the geodesic in the direction of $\bar{x}_i$. The approach is shown in Figure 5.1. In other words, we need to identify a vector $\xi_{x_i} \in T_{x_i}\mathcal{M}_p$ and its horizontal lift

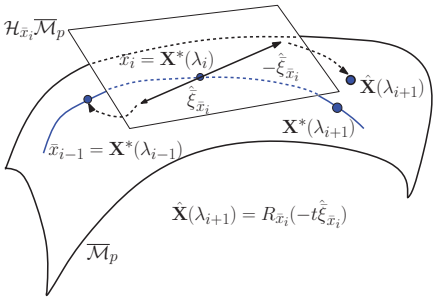


FIG. 5.1. *Tracing the path of solutions using the predictor-corrector approach. The blue line denotes the curve of optimal solutions.*

$\bar{\xi}_{\bar{x}_i} \in \mathcal{H}_{\bar{x}_i}\overline{\mathcal{M}}_p$ at $\bar{x}_i \in \overline{\mathcal{M}}_p$ defined as $\bar{\xi}_{\bar{x}_i} = \mathrm{Log}_{\bar{x}_i}(\bar{x}_{i-1})$ that maps $\bar{x}_{i-1}$ to a vector in the horizontal space $\mathcal{H}_{\bar{x}_i}\overline{\mathcal{M}}_p$. Log is called the *logarithmic* mapping (inverse of the exponential map) [24, 1]. Computing the logarithmic mapping might be numerically costly in general. For the case of interest there is no analytic expression for the logarithmic mapping. Instead a numerically efficient way is to use (locally around $\bar{x}_i$) an approximate inverse retraction $\hat{R}^{-1}_{\bar{x}_i}(\bar{x}_{i-1})$, where $\hat{R}^{-1}_{\bar{x}_i} : \overline{\mathcal{M}}_p \to \mathcal{E}$ to obtain a direction in the space $\mathcal{E}$ followed by *projection* onto the horizontal space $\mathcal{H}_{\bar{x}_i}\overline{\mathcal{M}}_p$. Note that $\mathcal{E} := \mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p} \times \mathbb{R}^{m\times p}$. The projection is accomplished using projection operators $\Psi_{\bar{x}_i} : \mathcal{E} \to T_{\bar{x}_i}\overline{\mathcal{M}}_p$ and $\Pi_{\bar{x}_i} : T_{\bar{x}_i}\overline{\mathcal{M}}_p \to \mathcal{H}_{\bar{x}_i}\overline{\mathcal{M}}_p$ defined in section 3. Hence, an estimate on $\bar{\xi}_{\bar{x}_i}$ is given as

$$\hat{\bar{\xi}}_{\bar{x}_i} = \Pi_{\bar{x}_i}(\Psi_{\bar{x}_i}(\hat{R}^{-1}_{\bar{x}_i}(\bar{x}_{i-1}))). \tag{5.1}$$

For the retraction of interest (3.16) the *Frobenius norm* error in the approximation of the logarithmic mapping is bounded as

$$\begin{aligned}\|\hat{\bar{\xi}}_{\bar{x}_i} - \bar{\xi}_{\bar{x}_i}\|_F &= \|\hat{\bar{\xi}}_{\bar{x}_i} - \hat{R}^{-1}_{\bar{x}_i}(\bar{x}_{i-1}) + \hat{R}^{-1}_{\bar{x}_i}(\bar{x}_{i-1}) - \bar{\xi}_{\bar{x}_i}\|_F \\ &\le \|\hat{\bar{\xi}}_{\bar{x}_i} - \hat{R}^{-1}_{\bar{x}_i}(\bar{x}_{i-1})\|_F + \|\hat{R}^{-1}_{\bar{x}_i}(\bar{x}_{i-1}) - \bar{\xi}_{\bar{x}_i}\|_F \\ &\le \min_{\bar{\zeta}_{\bar{x}_i} \in \mathcal{H}_{\bar{x}_i}\overline{\mathcal{M}}_p} \|\bar{\zeta}_{\bar{x}_i} - \hat{R}^{-1}_{\bar{x}_i}(\bar{x}_{i-1})\|_F + O(\|\bar{\xi}_{\bar{x}_i}\|_F^2) \\ &\quad \text{as} \quad \|\bar{\xi}_{\bar{x}_i}\| \to 0.\end{aligned}$$

The $O(\|\bar{\xi}_{\bar{x}_i}\|_F^2)$ approximation error comes from the fact that the retraction $R_{\bar{x}_i}$ (3.16) used is at least a first-order retraction. This approximation is exact if $\overline{\mathcal{M}}_p$ is the Euclidean space. The approximate inverse retraction $\hat{R}^{-1}_{\bar{x}_i}$, corresponding to the retraction $R_{\bar{x}_i}$ proposed in (3.16), is computed as

$$\hat{R}^{-1}_{\bar{x}_i}(\bar{x}_{i-1}) = (\mathbf{U}_{i-1} - \mathbf{U}_i,\ \mathbf{B}_i^{\frac{1}{2}}\mathrm{logm}(\mathbf{B}_i^{-\frac{1}{2}}\mathbf{B}_{i-1}\mathbf{B}_i^{-\frac{1}{2}})\mathbf{B}_i^{\frac{1}{2}},\ \mathbf{V}_{i-1} - \mathbf{V}_i), \tag{5.2}$$

where $\mathrm{logm}(\cdot)$ is the matrix logarithm operator. The predicted solution is then obtained by taking a step $t > 0$ and performing a backtracking line search in the direction $-\hat{\bar{\xi}}_{\bar{x}_i}$, i.e.,

$$\hat{\mathbf{X}}(\lambda_{i+1}) = R_{\bar{x}_i}(-t\hat{\bar{\xi}}_{\bar{x}_i}). \tag{5.3}$$

A good choice for the initial step-size t is $(\lambda_{i+1} - \lambda_i)/(\lambda_i - \lambda_{i-1})$. The motivation for the choice comes from the observation that it is optimal when the solution path is a straight line in the Euclidean space. The numerical complexity to perform the prediction step in the manifold $\overline{\mathcal{M}}_p$ is $O(np^2 + mp^2 + p^3)$.

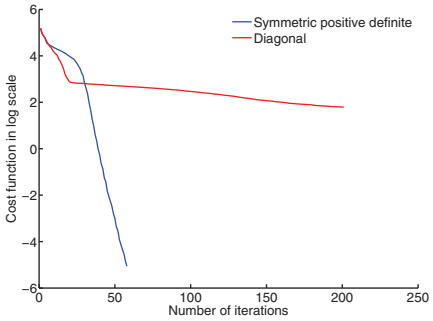


FIG. 6.1. *Convergence of a gradient descent algorithm is affected by making* $\mathbf{B}$ *diagonal.*

6. Numerical experiments. The overall optimization scheme with *descent-restart* and trust-regions algorithm is denoted as "Descent-restart + TR". We test the proposed optimization framework on the problems of low-rank matrix completion and multivariate linear regression where trace norm penalization has shown efficient recovery. Full regularization paths are constructed with optimality certificates. All simulations in this section have been performed in MATLAB on a 2.53 GHz Intel Core i5 machine with 4 GB of RAM. Our matrix completion implementation may be downloaded from http://www.montefiore.ulg.ac.be/~mishra/codes/traceNorm.html.

6.1. Diagonal versus matrix scaling. Before entering a detailed numerical experiment we illustrate here the empirical evidence that constraining $\mathbf{B}$ to be diagonal (as is the case with SVD) is detrimental to optimization. To this end, we consider the simplest implementation of a gradient descent algorithm for the matrix completion problem (see below). The plots shown Figure 6.1 compare the behavior of the same algorithm in the search space $\mathrm{St}(p,n) \times S_{++}(p) \times \mathrm{St}(p,m)$ and $\mathrm{St}(p,n) \times \mathrm{Diag}_+(p) \times \mathrm{St}(p,m)$ (SVD). $\mathrm{Diag}_+(p)$ is the set of diagonal matrices with positive entries. The empirical observation that convergence suffers from imposing diagonal structure on $\mathbf{B}$ is a generic observation that does not depend on the particular problem at hand. The problem here involves completing a 200×200 of rank 5 from 40% of observed entries. λ is fixed at 10^{-10}.

6.2. Low-rank matrix completion. The problem of matrix completion involves completing an $n \times m$ matrix when only a few entries of the matrix entries are known. Given an incomplete low-rank (but unknown rank) $n \times m$ real matrix $\widetilde{\mathbf{X}}$, a convex relaxation of the matrix completion problem is

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times m}} \quad \|\mathbf{W} \odot (\widetilde{\mathbf{X}} - \mathbf{X})\|_F^2 + \lambda \|\mathbf{X}\|_* \tag{6.1}$$

for $\mathbf{X} \in \mathbb{R}^{n \times m}$ and a regularization parameter $\lambda \in \mathbb{R}_+$. Here $\|\cdot\|_F$ denotes the Frobenius norm, matrix $\mathbf{W}$ is an $n \times m$ *weight* matrix with binary entries, and the operator $\odot$ denotes elementwise multiplication. If $\mathcal{W}$ is the set of known entries in $\widetilde{\mathbf{X}}$, then $\mathbf{W}_{ij} = 1$ if $(i,j) \in \mathcal{W}$ and $\mathbf{W}_{ij} = 0$ otherwise. The problem of matrix completion is known to be combinatorially hard. However, by solving the convex relaxation (6.1) a low-rank reconstruction is possible with a very high probability under Gaussian distribution of the observed entries [14, 21]. For an exact reconstruction, the lower bound on the number of known entries is typically of the order $O(nr + mr)$, where r is the optimal rank, $|\mathcal{W}| > \max(n,m) \gg r$. Consequently, it leads to a very sparse weight matrix $\mathbf{W}$, which plays a very crucial role for efficient algorithmic

implementations. For our case, we assume that the lower bound on the number of entries is met and we seek a solution to the optimization problem (6.1). Customizing the terminology for the present problem, the convex function is $f(\mathbf{X}) = \|\mathbf{W} \odot (\widetilde{\mathbf{X}} - \mathbf{X})\|_F^2$. Using the factorization $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ of (2.1), the rank-p objective function is $\bar{\phi}(\mathbf{U}, \mathbf{B}, \mathbf{V}) = \|\mathbf{W} \odot (\widetilde{\mathbf{X}} - \mathbf{U}\mathbf{B}\mathbf{V}^T)\|_F^2 + \lambda \mathrm{Trace}(\mathbf{B})$, where $(\mathbf{U}, \mathbf{B}, \mathbf{V}) \in \overline{\mathcal{M}}_p$. The dual variable $\mathbf{S} = 2(\mathbf{W} \odot (\mathbf{U}\mathbf{B}\mathbf{V}^T - \widetilde{\mathbf{X}}))$.

The matrix representation of the partial derivative of $\bar{\phi}$ in $\mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p} \times \mathbb{R}^{m\times p}$ is

$$\partial\bar{\phi}(\bar{x})/\partial\bar{x} = (\mathbf{S}\mathbf{V}\mathbf{B}, \mathbf{U}^T\mathbf{S}\mathbf{V} + \lambda\mathbf{I}, \mathbf{S}^T\mathbf{U}\mathbf{B}).$$

Similarly, the Euclidean directional derivative of $\partial\bar{\phi}(\bar{x})/\partial\bar{x}$ along $(\mathbf{Z_U}, \mathbf{Z_B}, \mathbf{Z_V}) \in T_{\bar{x}}\overline{\mathcal{M}}_p$ is $(\mathbf{S}\mathbf{V}\mathbf{Z_B} + \mathbf{S}\mathbf{Z_V}\mathbf{B} + \mathbf{S}_*\mathbf{V}\mathbf{B}, \mathbf{Z_U}^T\mathbf{S}\mathbf{V} + \mathbf{U}\mathbf{S}\mathbf{Z_V} + \mathbf{U}^T\mathbf{S}_*\mathbf{V}, \mathbf{S}^T\mathbf{U}\mathbf{Z_B} + \mathbf{S}^T\mathbf{Z_U}\mathbf{B} + \mathbf{S}_*^T\mathbf{U}\mathbf{B})$, where the auxiliary variable $\mathbf{S}_* = \mathrm{D}_{(\mathbf{U},\mathbf{B},\mathbf{V})}\mathbf{S}[(\mathbf{Z_U}, \mathbf{Z_B}, \mathbf{Z_V})] = 2(\mathbf{W} \odot (\mathbf{Z_U}\mathbf{B}\mathbf{V}^T + \mathbf{U}\mathbf{Z_B}\mathbf{V}^T + \mathbf{U}\mathbf{B}\mathbf{Z_V}^T))$ is the directional derivative of the dual variable $\mathbf{S}$ along $(\mathbf{Z_U}, \mathbf{Z_B}, \mathbf{Z_V})$.

The Riemannian gradient and Hessian are computed using formulas (3.11) and (3.14). Note that since $\mathbf{W}$ is sparse, $\mathbf{S}$ and $\mathbf{S}_*$ are sparse too. As a consequence, the numerical complexity per iteration for the trust-region algorithm is of order $O(|\mathcal{W}|p + np^2 + mp^2 + p^3)$, where $|\mathcal{W}|$ is the number of known entries. In addition computation of dominant singular value and vectors for rank-one updating (4.1) costs $O(|\mathcal{W}|)$ [23], potentially allowing us to handle large datasets.

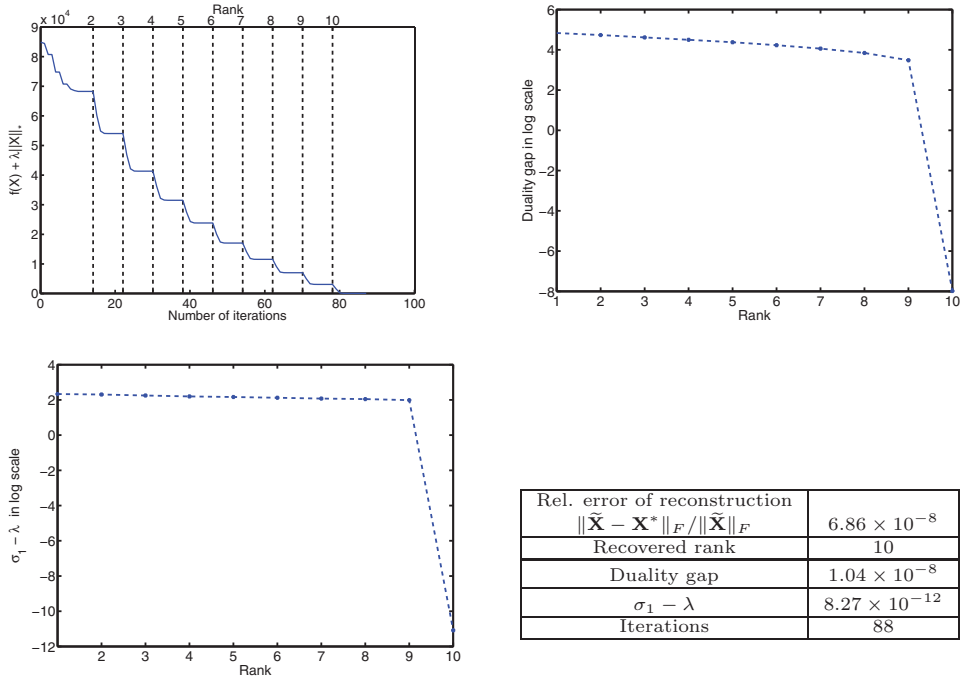
Fenchel dual and duality gap for matrix completion. From Proposition 2.3, after a routine calculation, the Fenchel conjugate of f has the expression $f^*(\mathbf{M}) = \mathrm{Trace}(\mathbf{M}^T\mathbf{M})/4 + \mathrm{Trace}(\mathbf{M}^T(\mathbf{W} \odot \widetilde{\mathbf{X}}))$, where the domain of f^* is the nonzero support of $\mathbf{W}$. Therefore, the duality gap for a duality gap candidate $\mathbf{M} = \min(1, \frac{\lambda}{\sigma_f})\mathrm{Grad}f(\mathbf{X})$ is

$$f(\mathbf{X}) + \lambda\|\mathbf{X}\|_* + \mathrm{Trace}(\mathbf{M}^T\mathbf{M})/4 + \mathrm{Trace}(\mathbf{M}^T(\mathbf{W} \odot \widetilde{\mathbf{X}})),$$

where σ_f is the dominant singular value of $\mathrm{Grad}f(\mathbf{X})$.

Simulations. Next we provide some benchmark simulations for the low-rank matrix completion problem. For each example, an $n \times m$ random matrix of rank p is generated as in [13]. Two matrices $\mathbf{A} \in \mathbb{R}^{n\times p}$ and $\mathbf{B} \in \mathbb{R}^{m\times p}$ are generated according to a Gaussian distribution with zero mean and unit standard deviation. The matrix product $\mathbf{A}\mathbf{B}^T$ then gives a random matrix of rank p. A fraction of the entries are randomly removed with uniform probability. The dimensions of $n \times m$ matrices of rank p is $(n+m-p)p$. The oversampling (OS) ratio determines the number of entries that are known. An OS $= 6$ means that $6(n + m - p)p$ randomly and uniformly selected entries are known a priori out of nm entries.

Example 1. Fixed $\boldsymbol{\lambda}$. A 100×100 random matrix of rank 10 is generated as mentioned above. 20% (OS $= 4.2$) of the entries are randomly removed with uniform probability. To reconstruct the original matrix we run the optimization scheme proposed in Table 4.1 along with the trust-region algorithm to solve the nonconvex problem. For illustration purposes λ is fixed at 10^{-5}. We also assume that we do not have any a priori knowledge of the optimal rank and, thus, start from rank 1. The trust-region algorithm stops when the relative or absolute variation of the cost function is below 10^{-10}. The rank-incrementing strategy stops when the relative duality gap is less than 10^{-5}, i.e., $\frac{f(\mathbf{X})+\lambda\|\mathbf{X}\|_*+f^*(\mathbf{M})}{|f^*(\mathbf{M})|} \leq 10^{-5}$. Convergence plots of the



Rel. error of reconstruction $\|\widetilde{\mathbf{X}} - \mathbf{X}^*\|_F / \|\widetilde{\mathbf{X}}\|_F$	6.86×10^{-8}
Recovered rank	10
Duality gap	1.04×10^{-8}
$\sigma_1 - \lambda$	8.27×10^{-12}
Iterations	88

FIG. 6.2. *Matrix completion by trace norm minimization algorithm with* $\lambda = 10^{-5}$. *Upper left: Rank incremental strategy with descent directions. Upper right: Optimality certificate of the solution with duality gap. Lower left: Convergence to the global solution according to Proposition* 2.2. *Lower right: Recovery of the original low-rank matrix.*

TABLE 6.1
Efficacy of trace norm penalization to reconstruct low-rank matrices by solving (6.1).

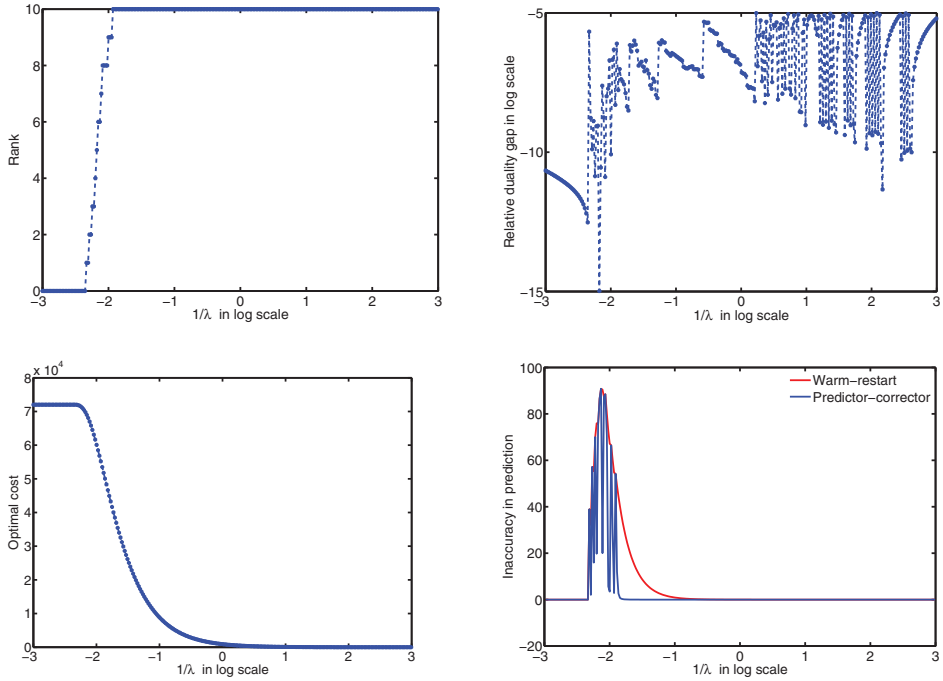
λ	10	10^{-2}	10^{-5}	10^{-8}
Rel. reconstruction error	6.33×10^{-2}	7.42×10^{-5}	7.11×10^{-8}	6.89×10^{-11}
Recovered rank	10	10	10	10
Iterations	113	120	119	123
Time in seconds	2.7	2.8	2.9	2.9

scheme are shown in Figure 6.2. A good way to characterize matrix reconstruction at $\mathbf{X}$ is to look at the relative error of reconstruction, defined as

$$\text{Rel. error of reconstruction} = \|\widetilde{\mathbf{X}} - \mathbf{X}\|_F / \|\widetilde{\mathbf{X}}\|_F.$$

Next, to understand low-rank matrix reconstruction by trace norm minimization we repeat the experiment for a number of values of λ all initialized from the same starting point and report the relative reconstruction error in Table 6.1 averaged over five runs. This, indeed, confirms that matrix reconstruction is possible by solving the trace norm minimization problem (6.1).

Example 2. Regularization path for matrix completion. In order to compute the entire regularization path, we employ the predictor-corrector approach described in Table 5.1 to find solutions for a grid of λ values. For the purpose of illustration, a geometric sequence of λ values is created with the maximum value



# λ values	270
# iterations	766
Time	38.60 seconds

FIG. 6.3. *Computation of entire regularization path using Descent-restart + TR with a predictor-corrector approach. Upper left: Recovery of solutions of all ranks. Upper right: Optimality certificate for the regularization path. Lower left: Path traced by the algorithm. Lower right: Better prediction by the algorithm in Table* 5.1 *than a pure warm-restart approach. Table: Number of iterations per value of* λ *is* < 3.

fixed at $\lambda_1 = 10^3$, the minimum value is set at $\lambda_N = 10^{-3}$, and a reduction factor $\gamma = 0.95$ such that $\lambda_{i+1} = \gamma\lambda_i$. We consider the example proposed previously. The algorithm for a $\lambda_i \in \{\lambda_1, \ldots, \lambda_N\}$ stops when the relative duality gap is less than 10^{-5}. Various plots are shown in Figure 6.3. Figure 6.3 also demonstrates the advantage of the scheme in Table 5.1 with respect to a warm-restart approach. We compare both approaches on the basis of

$$\text{Inaccuracy in prediction} = \bar{\phi}(\hat{\mathbf{X}}(\lambda_i)) - \bar{\phi}(\mathbf{X}^*(\lambda_i)), \tag{6.2}$$

where $\mathbf{X}^*(\lambda_i)$ is the global minimum at λ_i and $\hat{\mathbf{X}}(\lambda_i)$ is the prediction. A lower inaccuracy means better prediction. It should be emphasized that in Figure 6.2 most of the points on the curve of the objective function have no other utility than being intermediate iterates towards the global solution of the algorithm. In contrast all the points of the curve of optimal cost values in Figure 6.3 are now global minima for different values of λ.

Example 3. Competing methods for matrix completion. In this section, we analyze the following state-of-the-art algorithms for low-rank matrix completion, namely,

1. SVT algorithm by Cai, Candès, and Shen [13],
2. FPCA algorithm by Ma, Goldfarb, and Chen [25],

3. SOFT-IMPUTE (SOFT-I) algorithm by Mazumder, Hastie, and Tibshirani [26],
4. APG and APGL algorithms by Toh and Yun [39].

While FPCA, SOFT-I, and APGL solve (6.1), the iterates of SVT converge towards a solution of the optimization problem that minimizes $\tau\|\mathbf{X}\|_* + \frac{1}{2}\|\mathbf{X}\|_F^2$ subject to the constraint that the entries of $\mathbf{X}$ belonging to the set $\mathcal{W}$ agree with that of $\widetilde{\mathbf{X}}$, i.e., $\mathbf{W} \odot \mathbf{X} = \mathbf{W} \odot \widetilde{\mathbf{X}}$. τ is a regularization parameter for SVT.

For our simulation studies we use the MATLAB codes supplied on the authors' webpages for SVT, FPCA, and APGL. Due to simplicity of the SOFT-I algorithm we use our own MATLAB implementation. The numerically expensive step in all these algorithms is the computation of the SVT operation. To reduce the computational burden FPCA uses a linear time approximate SVD. Likewise, implementations of SVT, SOFT-I, and APGL exploit the low-rank + sparse structure of the iterates to optimize the thresholding operation [23].

The basic algorithm FPCA by Ma, Goldfarb, and Chen [25] is a fixed-point algorithm with a proven bound on the iterations for ϵ-accuracy. To accelerate the convergence they use the technique of *continuation* that involves approximately solving a sequence of parameters leading to the target λ. The SVT burden step is carried out by a linear time approximate SVD. The basic algorithm APG of Toh et al. is a proximal method [33] and gives a much stronger bound $O(1/\sqrt{\epsilon})$ on the number of iterations for ϵ-accuracy. To accelerate the scheme, the authors propose three additional heuristics: continuation, truncation (hard thresholding of ranks by projecting onto fixed-rank matrices), and a line search technique for estimating the Lipschitz constant. The accelerated version is called APGL. The basic algorithm SOFT-I iteratively replaces the missing elements with those given by an approximate SVD thresholding at each iteration. Accelerated versions involve postprocessing like continuation and truncation. It should be emphasized that the performance of SOFT-I greatly varies with the singular values computation at each iteration. For our simulations we compute 20 dominant singular values at each iteration of SOFT-I.

Convergence behavior with varying $\boldsymbol{\lambda}$. In this section we analyze the algorithms FPCA, SOFT-I, and Descent-restart + TR in terms of their ability to solve (6.1) for a fixed value of λ. For this simulation, we use FPCA, SOFT-I, and APGL without any acceleration techniques like continuation and truncation. SVT is not used for this test since it optimizes a different cost function. We plot the objective function $f(\mathbf{X}) + \lambda\|\mathbf{X}\|_*$ against the number of iterations for a number of λ values. A 100×100 random matrix of rank 5 is generated under standard assumptions with OS ratio $= 4$ (61% of entries are removed uniformly). The algorithms Descent-restart + TR, FPCA, SOFT-I, and APG are initialized from the same point. The algorithms are stopped when either the variation or relative variation of $f(\mathbf{X}) + \lambda\|\mathbf{X}\|_*$ is less than 10^{-10}. The maximum number of iterations is set at 500. The rank incrementing procedure of our algorithm is stopped when the relative duality gap is less than 10^{-5}.

The plots are shown in Figure 6.4. The convergence behavior of FPCA is greatly affected by λ. It has a slow convergence for a small λ while for a larger λ, the algorithm fluctuates. SOFT-I has a better convergence in all three cases; however, the convergence suffers when a more accurate solution is sought. The performance of APG is robust to the change in values of λ. For moderate accuracy it outperforms all other algorithms. However, when a higher accuracy is sought it takes a large number of iterations. Descent-restart + TR, on the other hand, outperforms others in all the cases here with minimal number of iterations.

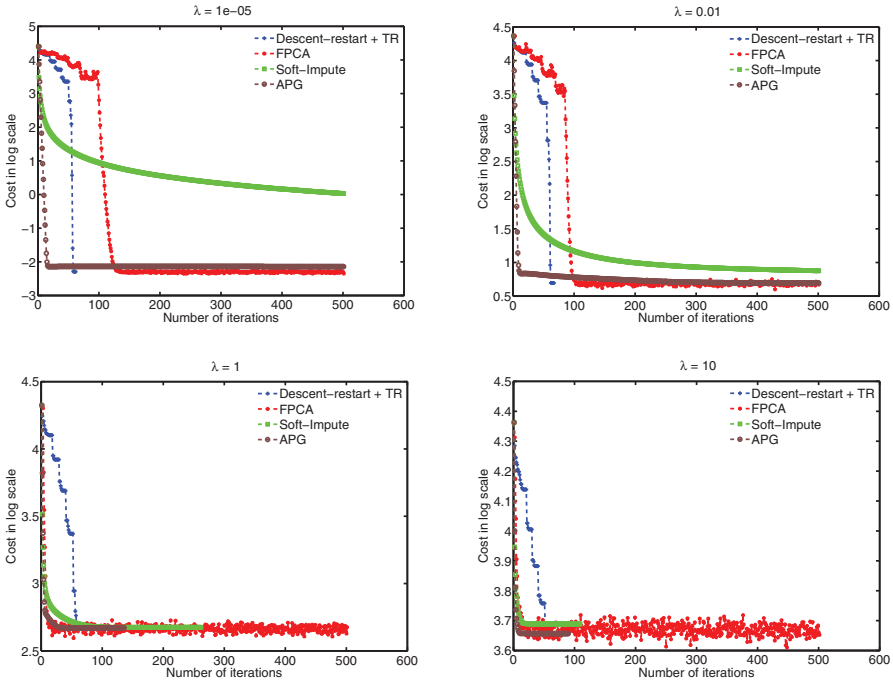


FIG. 6.4. *Convergence behavior of different algorithms for different values of* λ. *The algorithms compared here do not use any acceleration heuristics.*

Convergence test. To understand the convergence behavior of different algorithms involving different optimization problems, we look at the evolution of the training error [13, 26] defined as

$$\text{Training error} = \|\mathbf{W} \odot (\widetilde{\mathbf{X}} - \mathbf{X})\|_F^2, \tag{6.3}$$

with iterations. We generate a 150×300 random matrix of rank 10 under standard assumptions with OS $= 5$. The algorithms Descent-restart + TR, FPCA, and SOFT-I are initialized similarly. We fix $\lambda = 10^{-5}$ as it gives a good reconstruction for comparing algorithms. For SVT we use the default values of τ and step size as suggested in [13]. The algorithms are stopped when the variation or relative variation of the training error (6.3) is less than 10^{-10}. The maximum number of iterations is set at 500. The rank incrementing procedure of our algorithm is stopped when the relative duality gap is below 10^{-5}.

In Figure 6.5 APG has a fast convergence but the performance slows down later. Consequently, it exceeds the maximum limit of iterations. Similarly, SOFT-I converges to a different solution but has a faster convergence in the initial phase (for iterations less than 60). FPCA and Descent-restart + TR converge faster at a later stage of their iterations. Descent-restart + TR initially sweeps through ranks until arriving at the optimal rank where the convergence is accelerated owing to the trust-region algorithm.

Scaling test. To analyze the scalability of these algorithms to larger problems we perform a test where we vary the problem size n from 200 to 2200. The reason for choosing a moderate value of n is that large-scale implementations of SVT, FPCA, and SOFT-I are unavailable from authors' webpages. For each n, we generate a random

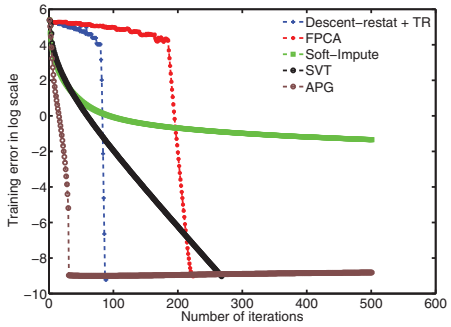
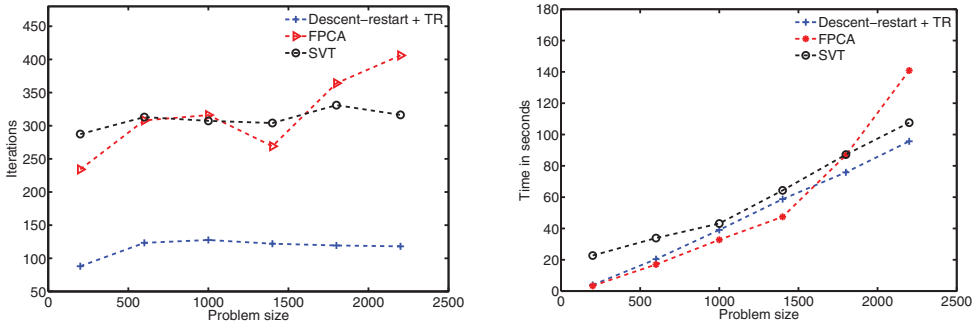


FIG. 6.5. *Convergence behavior of different algorithms for minimizing the training error* (6.3).



n	200	600	1000	1400	1800	2200
$\lvert\mathcal{W}\rvert$	18409	66676	120000	176184	234380	294134
f	0.46	0.19	0.12	0.09	0.07	0.06
OS	4.7	5.6	6.0	6.3	6.5	6.7

FIG. 6.6. *Analysis of the algorithms on randomly generated datasets of rank* 10 *with varying fractions of missing entries. SVT, FPCA, and Descent-restart + TR have similar performances but Descent-restart + TR usually outperforms the others.*

matrix of size $n \times n$ of rank 10 under standard assumptions with different OS ratios. The initializations are chosen as in the earlier example, i.e., $\lambda = 10^{-5}$. We note the time and number of iterations taken by the algorithms until the stopping criterion is satisfied or when the number of iterations exceed 500. The stopping criterion is the same as the one used before for comparison, when the absolute variation or relative variation of the training error (6.3) is less than 10^{-10}. Results averaged over five runs are shown in Figure 6.6. We have not shown the plots for SOFT-I and APG as in all the cases either they did not converge in 500 iterations or took much more time than the nearest competitor.

Below we have shown two more case studies where we intend to show the numerical scalability of our algorithm to large-scale instances. The first one involves comparisons with fixed-rank optimization algorithms. The second case is a large-scale comparison with APGL (the accelerated version of APG). We consider the problem of completing a 50000×50000 matrix $\widetilde{\mathbf{X}}$ of rank 5. The OS ratio is 8 implying that 0.16% (3.99×10^6) of entries are randomly and uniformly revealed. The maximum number of iterations is fixed at 500.

Fixed-rank comparison. Because our algorithm uses a fixed-rank approach for the fixed-rank subproblem, it is also meaningful to compare its performance with other

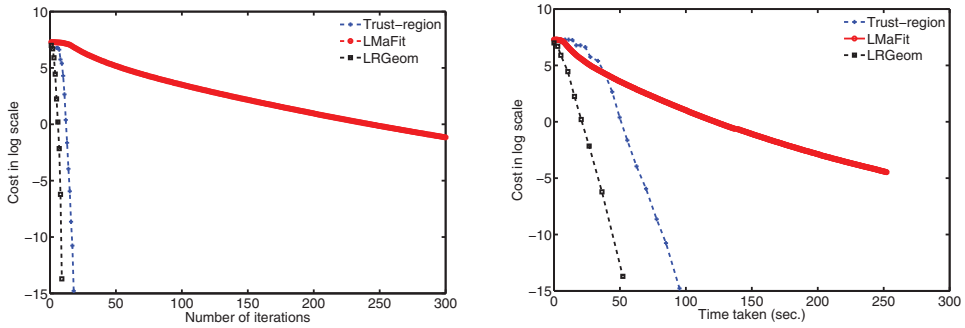


FIG. 6.7. *Rank* 5 *completion of* 50000×50000 *matrix with* OS $= 8$. *All the algorithms are initialized by taking* 5 *dominant SVDs of sparse* $\widetilde{\mathbf{X}}$ *as proposed in* [20]. *Algorithms are stopped when the objective function falls below a threshold,* $\|\mathbf{W} \odot (\widetilde{\mathbf{X}} - \mathbf{X})\|_F^2 \leq 10^{-10}$. *The proposed trust-region scheme is competitive with LMaFit for large scale problems. Although LMaFit has a smaller time complexity per iteration, its convergence seems to suffer for large-scale problems. With respect to LRGeom, the performance is poorer although both have a similar asymptotic rate of convergence.*

fixed-rank optimization algorithms. However, a rigorous comparison with other algorithms is beyond the scope of the present paper. We refer to a recent paper [31] that deals with this question in a broader framework. Here we compare with the two state-of-the-art algorithms LMaFit [42] and LRGeom (trust-region implementation) [40]. LMaFit is an alternating minimization scheme with a different factorization for a fixed-rank matrix. We use the fixed-rank implementation of LMaFit. It is a tuned version of the Gauss–Seidel nonlinear scheme and has a smaller time complexity per iteration. LRGeom is based on the embedded geometry of fixed-rank matrices. This viewpoint allows us to simplify notions of moving on the search space. We use their trust-region implementation. The geometry leads to an efficient guess of the optimal step size in a search direction. Figure 6.7 shows a competitive performance of our trust-region scheme with respect to LMaFit. Asymptotically, both our trust-region scheme and LRGeom perform similarly with LRGeom performing better in the initial phase.

Comparison with APGL. APG has a better iteration complexity than other optimization algorithms. However, scalability of APG by itself to larger dimensional problems is an issue. The principal bottleneck is that the ranks of the intermediate iterates seem to be uncontrolled and only asymptotically, a low-rank solution is expected. To circumvent this issue, an accelerated version of APG called APGL is also proposed in [39]. APGL is APG with three additional heuristics: *continuation* (a sequence of parameters leading to the target λ), *truncation* (hard thresholding of ranks by projecting onto fixed-rank matrices), and a line-search technique for estimating the Lipschitz constant L_f for the first derivative of the cost function. We compare our algorithm with APGL. The algorithms are stopped when either absolute variation or relative variation of the objective function is less than 10^{-10}. For our algorithm, the trust-region algorithm is also terminated with the same criterion. In addition, the rank-one updating is stopped when the relative duality gap is below 10^{-5}.

For a fixed $\lambda = \bar{\lambda}$, APGL proceeds through a sequence of values for λ such that $\lambda_k = \max\{0.7\lambda_{k-1}, \bar{\lambda}\}$, where k is the iteration count of the algorithm. Initially λ_0 is set to $2\|\mathbf{W} \odot \widetilde{\mathbf{X}}\|_{op}$. We also follow a similar approach and create a sequence of values. A decreasing sequence is generated leading to $\bar{\lambda}$ by using the recursive rule, $\lambda_i = \lambda_{i-1}/2$ when $\lambda_{i-1} > 1$ and $\lambda_i = \lambda_{i-1}/100$ otherwise until $\lambda_{i-1} < \bar{\lambda}$. Initial λ_0 is set to $\|\mathbf{W} \odot \widetilde{\mathbf{X}}\|_{op}$. For $\lambda_i \neq \bar{\lambda}$ we also relax the stopping criterion for the trust region

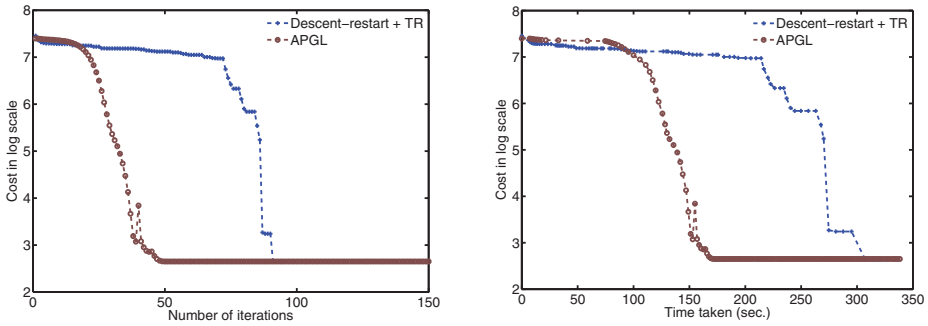


FIG. 6.8. *A large-scale instance of rank* 5 *completion of* 50000×50000 *matrix with* OS $= 8$. $\lambda = 2\|\mathbf{W} \odot \widetilde{\mathbf{X}}\|_{op}/10^5$ *as suggested in* [39]. *The proposed framework is competent for very low ranks and when a high accuracy is sought. However, we spend considerable time in just traversing through ranks before arriving at the optimal rank.*

to 10^{-5} as well as stopping the rank-one increment when the relative duality gap is below 1 as we are only interested in an accurate solution for $\lambda = \bar{\lambda}$.

In Figure 6.8 we compete favorably with APGL in large-scale problems for very low ranks and when a higher accuracy is required. However, as the rank increases, APGL performs better. This is not surprising as our algorithm traverses all ranks, one by one before arriving at the optimal rank. In the process we spend a considerable effort in traversing ranks. This approach is most effective only when computing in the full regularization path. Also for moderate accuracy, APGL performs extremely well. However, the better performance of APGL significantly relies on heuristics like continuation and truncation. The truncation heuristic allows the APGL algorithm to approximate an iterate by a low- and fixed-rank iterate. On the other hand, we strictly move in the low-rank space. Exploiting this leads to an efficient way for computing the full regularization path using a predictor-corrector strategy of section 5.

Comments on matrix completion algorithms. We summarize our observations in the following points.

- The convergence rate of SOFT-I is greatly dependent on the computation of singular values. For large-scale problems this is a bottleneck and the performance is greatly affected. However, in our experiments, it performs quite well within a reasonable accuracy as seen in Figures 6.4 and 6.5.
- SVT, in general, performs well on random examples. However, the choice of step size and regularization parameter τ affect the convergence speed of the algorithm [25, 26].
- FPCA has a superior numerical complexity per iteration owing to an approximate SVD [25], but the performance suffers as the regularization parameter λ is increased as shown in Figure 6.4.
- APG has a better iteration complexity than the others and is well suited when a moderate accuracy is required (Figures 6.4 and 6.5). As the ranks of the intermediate iterates are not necessarily low, scalability to large dimensions is an issue. Its accelerated version APGL does not suffer from this problem and performs very well for large dimensions.
- In all our simulation studies on random examples, Descent-restart + TR has shown a favorable performance on different benchmarks. In particular our framework is well suited when the optimal solution is low rank and when one needs to compute the regularization path. The Riemannian geometry

of the set of fixed-rank matrices allows us to make a first-order prediction of the regularization path, thereby employing an efficient predictor-corrector strategy.

6.3. Multivariate linear regression. Given matrices $\mathbf{Y} \in \mathbb{R}^{n\times k}$ (response space) and $\mathbf{X} \in \mathbb{R}^{n\times q}$ (input data space), we seek to learn a weight/coefficient matrix $\mathbf{W} \in \mathbb{R}^{q\times k}$ that minimizes the *loss* between $\mathbf{Y}$ and $\mathbf{XW}$ [43]. Here n is the number of observations, q is the number of predictors, and k is the number of responses. One popular approach to the multivariate linear regression problem is by minimizing a *quadratic loss* function. Note that in various applications, *responses* are related and may therefore, be represented with many fewer coefficients. From an optimization point of view this corresponds to finding a low-rank coefficient matrix. The papers [43, 2], thus, motivate the use of trace norm regularization in the following optimization problem formulation, defined as

$$\min_{\mathbf{W}\in\mathbb{R}^{q\times k}} \quad \|\mathbf{Y}-\mathbf{XW}\|_F^2 + \lambda\|\mathbf{W}\|_*.$$

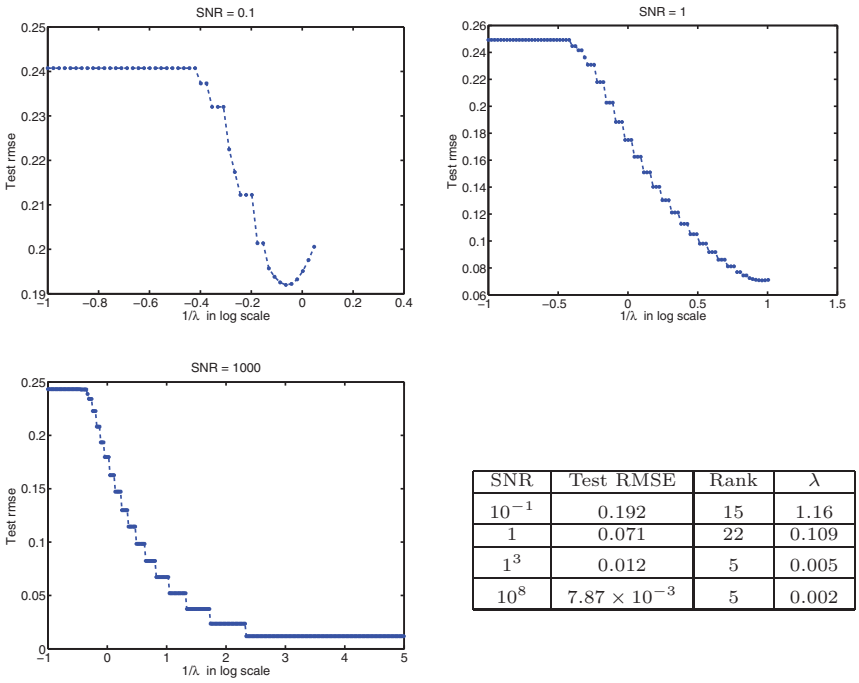
(The optimization variable is $\mathbf{W}$.) Although the focus here is on the quadratic loss function, the framework can be directly applied to other smooth loss functions. Other than the difference in the dual variable $\mathbf{S}$ and $\mathbf{S}_*$, the rest of the computation of the gradient and its directional derivative in the Euclidean space is similar to that of the low-rank matrix completion case. The matrix representations of the auxiliary variables are $\mathbf{S} = 2(\mathbf{X}^T\mathbf{XW} - \mathbf{X}^T\mathbf{Y})$ and $\mathbf{S}_* = \mathrm{D}_{(\mathbf{U},\mathbf{B},\mathbf{V})}\mathbf{S}[\mathbf{Z}] = 2(\mathbf{X}^T\mathbf{X}(\mathbf{Z}_\mathbf{U}\mathbf{BV}^T + \mathbf{UZ}_\mathbf{B}\mathbf{V}^T + \mathbf{UBZ}_\mathbf{V}^T))$, where the rank of $\mathbf{W}$ is p and $\mathbf{W} = \mathbf{UBV}^T$.

The numerical complexity per iteration is dominated by the numerical cost to compute $\bar{\phi}(\mathbf{U},\mathbf{B},\mathbf{V})$, $\mathbf{S}$, and terms like $\mathbf{SVB}$. The cost of computing $\bar{\phi}$ is of $O(nqp + nkp + kp^2 + nk)$ and $\mathbf{SVB}$ is $O(q^2p + qkp + kp^2)$. And that of full matrix $\mathbf{S}$ is $O(q^2p + qkp + kp^2)$. From a *cubic* numerical complexity of $O(q^2k)$ per iteration (using the full matrix $\mathbf{W}$) the low-rank factorization reduces the numerical complexity to $O(q^2p + qkp)$ which is *quadratic*. Note that the numerical complexity per iteration is linear in n.

Fenchel dual and duality gap computation. As an extension for some functions f of type $f(\mathbf{W}) = \psi(\mathcal{A}(\mathbf{W}))$, where $\mathcal{A}$ is a linear operator, computing the Fenchel conjugate of the function ψ may be easier than that of f. When $\|\mathcal{A}^*(\mathbf{M})\|_{op} \leq \lambda$ the duality gap, using calculations similar to Proposition 2.3, is $f(\mathbf{W}) + \lambda\|\mathbf{W}\|_* + \psi^*(\mathbf{M})$, where $\mathcal{A}^*$ is the adjoint operator of $\mathcal{A}$ and ψ^* is the Fenchel conjugate of the transformed function ψ. A good choice of $\mathbf{M}$ is again $\min\{1, \frac{\lambda}{\sigma_\psi}\}\mathrm{Grad}\psi$, where σ_ψ is the dominant singular value of $\mathcal{A}^*(\mathrm{Grad}\psi)$ [4].

For the multivariate linear regression problem we have $\mathcal{A}(\mathbf{W}) = \mathbf{XW}$ which suggests the choice $f(\mathbf{W}) = \psi(\mathbf{XW})$. Note that the domains of f and ψ are different. Finally, the duality gap is $f(\mathbf{W}) + \lambda\|\mathbf{W}\|_* + \psi^*(\mathbf{M})$, where the dual candidate $\mathbf{M} = 2\min(1, \frac{\lambda}{\sigma_\psi})(\mathbf{XW} - \mathbf{Y})$ and σ_ψ is the dominant singular value of $\mathcal{A}^*(\mathrm{Grad}\psi) = \mathbf{X}^T\mathrm{Grad}\psi = 2\mathbf{X}^T(\mathbf{XW} - \mathbf{Y})$. As we use a low-rank factorization of $\mathbf{W}$, i.e., $\mathbf{W} = \mathbf{UBV}^T$, the numerical complexity of finding the duality gap is dominated by the numerical cost of computing $\psi^*(\mathbf{M})$ which is also of the order of the cost of computing $\bar{\phi}(\mathbf{U},\mathbf{B},\mathbf{V})$. The numerical complexity of computing $\mathbf{M}$ is $O(nqp + nkp + kp^2)$ and of $\psi^*(\mathbf{M})$ is $O(nk)$.

Regularization path for multivariate linear regression. An input data matrix $\mathbf{X}$ of size 5000×120 is randomly generated according to a Gaussian distribution with zero mean and unit standard deviation. The response matrix $\mathbf{Y}$ is computed as



SNR	Test RMSE	Rank	λ
10^{-1}	0.192	15	1.16
1	0.071	22	0.109
1^3	0.012	5	0.005
10^8	7.87×10^{-3}	5	0.002

FIG. 6.9. *Regularization path for multivariate linear regression with various SNR values. Results are averaged over* 5 *random* 70/30 *splits.*

$\mathbf{XW}_*$, where $\mathbf{W}_*$ is a randomly generated coefficient matrix of a rank 5 matrix with size 120×100. We randomly split the observations as well as responses into *training* and *testing* datasets in the ratio 70/30 resulting in $\mathbf{Y}_{\text{train}}/\mathbf{Y}_{\text{test}}$ and $\mathbf{X}_{\text{train}}/\mathbf{X}_{\text{test}}$. A Gaussian white noise of zero mean and variance σ^2_{noise} is added to the training response matrix $\mathbf{Y}_{\text{train}}$ resulting in $\mathbf{Y}_{\text{noise}}$. We learn the coefficient matrix $\mathbf{W}$ by minimizing the *scaled* cost function, i.e.,

$$\min_{\mathbf{W}\in\mathbb{R}^{q\times k}} \quad \frac{1}{nk}\|\mathbf{Y}_{\text{noise}} - \mathbf{X}_{\text{train}}\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_*,$$

where λ is a regularization parameter. We validate the learning by computing the root mean square error (RMSE) defined as

$$\text{Test RMSE} = \sqrt{\frac{1}{n_{\text{test}}k}\|\mathbf{Y}_{\text{test}} - \mathbf{X}_{\text{test}}\mathbf{W}\|_F^2}$$

where n_{test} is the number of test observations. Similarly, the signal to noise ratio (SNR) is defined as $\sqrt{\frac{\|\mathbf{Y}_{\text{train}}\|_F^2}{\sigma^2_{\text{noise}}}}$.

We compute the entire regularization path for four different SNR values. The maximum value of λ is fixed at 10 and the minimum value is set at 10^{-5} with the reduction factor $\gamma = 0.95$ (270 values of λ in total). Apart from this we also put the restriction that we only fit ranks less than 30. The solution to an optimization problem for a value of λ is claimed to have been obtained when either the duality gap is less than 10^{-2} or the relative duality gap is below 10^{-2} or $\sigma_1 - \lambda$ is less than 10^{-2}. Similarly, the trust-region algorithm stops when relative or absolute variation of the cost function is less than 10^{-10}. The results are summarized in Figure 6.9.

7. Conclusion. Three main ideas have been presented in this paper. First, we have given a framework to solve a general trace norm minimization problem (1.1) with a sequence of increasing but fixed-rank nonconvex problems (2.2). We have analyzed the convergence criterion and duality gap which are used to monitor convergence to the solution of the original problem. The duality gap expression was shown numerically tractable even for large problems thanks to the specific choice of the low-rank parameterization. We have also given a way of incrementing the rank while simultaneously ensuring a decrease of the cost function. This may be termed as a *descent-restart* approach. The second contribution of the paper is to present a second-order trust-region algorithm for a general rank-p (fixed-rank) optimization in the quotient search space $\mathrm{St}(p,n) \times S_{++}(p) \times \mathrm{St}(p,m)/\mathcal{O}(p)$ equipped with the natural metric $\bar{g}$ (3.6). The search space with the metric $\bar{g}$ has the structure of a Riemannian submersion [1]. We have used the manifold-optimization techniques [1] to derive the required geometric objects in order to devise a second-order algorithm. With a proper parameter tuning the proposed trust-region algorithm guarantees a quadratic rate of convergence. The third contribution of the paper is to develop a predictor-corrector algorithm on the quotient manifold where the predictor step is along the first-order approximation of the geodesic. The corrector step is achieved by initializing the descent-restart approach from the predicted point. The resulting performance is superior to the warm-restart approach.

These ideas have been applied to the problems of low-rank matrix completion and multivariate linear regression leading to encouraging numerical results.

Appendix A. Proofs.

A.1. Derivation of first-order optimality conditions of (2.4). We derive the gradient $\mathrm{grad}_{\bar{x}}\bar{\phi}$ in the total space $\overline{\mathcal{M}}_p$ with the metric (3.6) using (3.11) at $\bar{x} = (\mathbf{U}, \mathbf{B}, \mathbf{V})$. First, we compute the partial derivative $\partial\bar{\phi}(\bar{x})/\partial\bar{x}$ of $\bar{\phi}$ in the Euclidean space $\mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p} \times \mathbb{R}^{m\times p}$ which has the matrix representation $(\mathbf{SVB}, \mathbf{U}^T\mathbf{SV} + \lambda\mathbf{I}, \mathbf{S}^T\mathbf{UB})$, where $\mathbf{S} = \mathrm{Grad} f(\mathbf{UBV}^T)$. Note that this development follows from the chain rule of computing partial derivatives. Finally, from (3.11)

$$\mathrm{grad}_{\mathbf{U}}\bar{\phi} = (\mathbf{SVB} - \mathbf{U}\mathrm{Sym}(\mathbf{U}^T\mathbf{SVB}),\ \mathrm{grad}_{\mathbf{B}}\bar{\phi} = \mathbf{B}\left(\mathrm{Sym}(\mathbf{U}^T\mathbf{SV}) + \lambda\mathbf{I}\right)\mathbf{B},$$
$$\mathrm{grad}_{\mathbf{V}}\bar{\phi} = \mathbf{S}^T\mathbf{UB} - \mathbf{V}\mathrm{Sym}(\mathbf{V}^T\mathbf{S}^T\mathbf{UB}).$$

The conditions (2.4) are obtained by equating $\|\mathrm{grad}_{\bar{x}}\bar{\phi}\|_{g_{\bar{x}}}$ to 0.

A.2. Proof of Proposition (2.2). From the characterization of the subdifferential of the trace norm [35] we have the following:

(A.1)
$$\partial\|\mathbf{X}\|_* = \{\mathbf{UV}^T + \mathbf{W} : \quad \mathbf{W} \text{ and } \mathbf{X} \text{ have orthogonal column and row spaces,}\ \mathbf{W} \in \mathbb{R}^{n\times m},\ \text{and } \|\mathbf{W}\|_{op} \leq 1\},$$

where $\mathbf{X} = \mathbf{UBV}^T$. Since $\mathbf{X} = \mathbf{UBV}^T$ is also a stationary point for the problem (2.2), the conditions (2.4) are satisfied including $\mathrm{Sym}(\mathbf{U}^T\mathbf{SV}) + \lambda\mathbf{I} = \mathbf{0}$. From the properties of a matrix norm we have

$$\begin{aligned} & \lambda\mathbf{I} = -\mathrm{Sym}(\mathbf{U}^T\mathbf{SV}) \\ \Rightarrow \quad & \lambda = \|\mathrm{Sym}(\mathbf{U}^T\mathbf{SV})\|_{op} \leq \|\mathbf{U}^T\mathbf{SV}\|_{op} \leq \mathbf{S}\|_{op}. \end{aligned}$$

Equality holds iff $\mathbf{U}$ and $\mathbf{V}$ correspond to the dominant row and column subspace of $\mathbf{S}$, i.e., if $\mathbf{S} = -\lambda\mathbf{UV}^T + \mathbf{U}_\perp\mathbf{\Sigma}{\mathbf{V}_\perp}^T$, where $\mathbf{U}^T\mathbf{U}_\perp = \mathbf{0}$, $\mathbf{V}^T\mathbf{V}_\perp = \mathbf{0}$, $\mathbf{U}_\perp \in \mathrm{St}(n-p, n)$,

$\mathbf{V}_\perp \in \mathrm{St}(m-p,m)$, and $\mathbf{\Sigma}$ is a diagonal matrix with positive entries with $\|\mathbf{\Sigma}\|_{op} \leq \lambda$. Note that this also means that $\mathbf{S} \in \lambda\partial\|\mathbf{X}\|_*$ such that $\mathbf{W} = \mathbf{U}_\perp\mathbf{\Sigma}{\mathbf{V}_\perp}^T$ which satisfies (A.1) and the global optimality condition (2.3). This proves Proposition (2.2).

A.3. Proof of Proposition (4.1). Since $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ is a stationary point for the problem (2.2) and not the global optimum of (1.1) by virtue of Proposition 2.2 we have $\|\mathbf{S}\|_{op} > \lambda$ (strict inequality). We assume that f is smooth and hence, the first derivative of f is Lipschitz continuous with the Lipschitz constant L_f, i.e., $\|\mathrm{Grad}f(\mathbf{X}) - \mathrm{Grad}f(\mathbf{Y})\|_F \leq L_f\|\mathbf{X}-\mathbf{Y}\|_F$ for any $\mathbf{X},\mathbf{Y} \in \mathbb{R}^{n\times m}$ [33]. Therefore, the update (4.1), $\mathbf{X}_+ = \mathbf{X} - \beta uv^T$ results in the following inequalities:

(A.2)

$$\begin{aligned} f(\mathbf{X}_+) &\leq f(\mathbf{X}) + \langle \mathrm{Grad}f(\mathbf{X}), \mathbf{X}_+ - \mathbf{X}\rangle + \tfrac{L_f}{2}\|\mathbf{X}_+ - \mathbf{X}\|_F^2 \text{ (from [33])} \\ &= f(\mathbf{X}) - \beta\sigma_1 + \tfrac{L_f}{2}\beta^2; \\ &\text{also} \\ \|\mathbf{X}_+\|_* &\leq \|\mathbf{X}\|_* + \beta \text{ (from triangle inequality of matrix norm)} \\ \Rightarrow \quad f(\mathbf{X}_+) + \lambda\|\mathbf{X}_+\|_* &\leq f(\mathbf{X}) + \lambda\|\mathbf{X}\|_* - \beta(\sigma_1 - \lambda - \tfrac{L_f}{2}\beta) \end{aligned}$$

for $\beta > 0$ and σ_1 being equal to $\|\mathbf{S}\|_{op} = \|\mathrm{Grad}f(\mathbf{X})\|_{op}$. The maximum decrease in the cost function is obtained by maximizing $\beta(\sigma_1 - \lambda - \frac{L_f}{2}\beta)$ with respect to β which gives $\beta_{\max} = \frac{\sigma_1-\lambda}{L_f} > 0$. $\beta_{\max} = 0$ only at optimality. This proves the proposition.

A.4. Proof of Proposition (2.3). Without loss of generality we introduce a dummy variable $\mathbf{Z} \in \mathbb{R}^{n\times m}$ to rephrase the optimization problem (1.1) as

$$\begin{aligned} \min_{\mathbf{X},\mathbf{Z}} \quad & f(\mathbf{X}) + \lambda\|\mathbf{Z}\|_* \\ \text{subject to} \quad & \mathbf{Z} = \mathbf{X}. \end{aligned}$$

The Lagrangian of the problem with dual variable $\mathbf{M} \in \mathbb{R}^{n\times m}$ is $\mathcal{L}(\mathbf{X},\mathbf{Z},\mathbf{M}) = f(\mathbf{X}) + \lambda\|\mathbf{Z}\|_* + \mathrm{Trace}(\mathbf{M}^T(\mathbf{Z}-\mathbf{X}))$. The *Lagrangian dual* function g of the Lagrangian $\mathcal{L}$ is, then, computed as [10, 4]

$$\begin{aligned} & g(\mathbf{M}) = \min_{\mathbf{X},\mathbf{Z}} \quad f(\mathbf{X}) - \mathrm{Trace}(\mathbf{M}^T\mathbf{X}) + \mathrm{Trace}(\mathbf{M}^T\mathbf{Z}) + \lambda\|\mathbf{Z}\|_* \\ \Rightarrow \quad & g(\mathbf{M}) = \min_{\mathbf{X}} \quad \{f(\mathbf{X}) - \mathrm{Trace}(\mathbf{M}^T\mathbf{X})\} + \min_{\mathbf{Z}} \quad \{\mathrm{Trace}(\mathbf{M}^T\mathbf{Z}) + \lambda\|\mathbf{Z}\|_*\}. \end{aligned}$$

Using the duality trace norm, i.e., operator norm we have

$$\min_{\mathbf{Z}} \quad \mathrm{Trace}(\mathbf{M}^T\mathbf{Z}) + \lambda\|\mathbf{Z}\|_* = 0 \quad \text{if} \quad \|\mathbf{M}\|_{op} \leq \lambda.$$

Similarly, using the concept of the Fenchel conjugate of a function we have

$$\min_{\mathbf{X}} \quad f(\mathbf{X}) - \mathrm{Trace}(\mathbf{M}^T\mathbf{X}) = -f^*(\mathbf{M}),$$

where f^* is the Fenchel conjugate [4, 10] of f, defined as $f^*(\mathbf{M}) = \sup_{\mathbf{X}\in\mathbb{R}^{n\times m}}[\mathrm{Trace}(\mathbf{M}^T\mathbf{X}) - f(\mathbf{X})]$. Therefore when $\|\mathbf{M}\|_{op} \leq \lambda$, the final expression for the dual function is $g(\mathbf{M}) = -f^*(\mathbf{M})$[4] and the Lagrangian dual formulation is

$$\max_{\mathbf{M}} -f^*(\mathbf{M}) \quad \text{such that} \quad \|\mathbf{M}\|_{op} \leq \lambda.$$

This proves the proposition.

Acknowledgments. We thank the editor and two anonymous reviewers for carefully checking the paper and providing a number of helpful remarks.

REFERENCES

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

[2] Y. Amit, M. Fink, N. Srebro, and S. Ullman, *Uncovering shared structures in multiclass classification*, in ICML, Zoubin Ghahramani, ed., ACM Internat. Conf. Proc. Ser. 227, ACM, New York, 2007, pp. 17–24.

[3] F. Bach, *Consistency of trace norm minimization*, J. Mach. Learn. Res., 9 (2008), pp. 1019–1048.

[4] F. Bach, R. Jenatton, J. Mairal, and G. Obozinsky, *Convex optimization with sparsity-inducing norms*, in Optimization for Machine Learning, S. Sra, S. Nowozin, S. J. Wright., eds., MIT Press, Cambridge, MA, 2011.

[5] C. G. Baker, P.-A. Absil, and K. A. Gallivan, *GenRTR: The Generic Riemannian Trust-Region Package*, http://www.math.fsu.edu/~cbaker/genrtr/ (2007).

[6] R. H. Bartels and G. W. Stewart, *Solution of the matrix equation $ax+xb = c$[F4] (algorithm 432)*, Commun. ACM, 15 (1972), pp. 820–826.

[7] R. Bhatia, *Positive Definite Matrices*, Princeton Ser. Appl. Math., Princeton University, Princeton, 2007.

[8] S. Bonnabel and R. Sepulchre, *Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1055–1070.

[9] N. Boumal and P.-A. Absil, *RTRMC: A Riemannian trust-region method for low-rank matrix completion*, in Proceedings of the Neural Information Processing Systems Conference, NIPS, Granada, Spain, 2011.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, 2004.

[11] M. Brand, *Fast low-rank modifications of the thin singular value decompositon*, Linear Algebra Appl., 415 (2006), pp. 20–30.

[12] S. Burer and R. D. C. Monteiro, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Math. Program., 95 (2003), pp. 329–357.

[13] J.-F. Cai, E. J. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982.

[14] E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.

[15] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.

[16] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. thesis, Stanford University, Palo Alto, CA, 2002.

[17] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Grundlebren Math. Wiss. 305–306, Springer-Verlag, Berlin, 1993.

[18] M. Journée, *Geometric Algorithms for Component Analysis with a View to Gene Expression Data Analysis*, Ph.D. thesis, University of Liège, Liège, Belgium, 2009.

[19] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre, *Low-rank optimization on the cone of positive semidefinite matrices*, SIAM J. Optim., 20 (2010), pp. 2327–2351.

[20] R. H. Keshavan and A. Montanari, *Regularization for matrix completion*, in IEEE International Symposium on Information Theory (ISIT), IEEE, Piscataway, NJ, 2010, pp. 1503–1507.

[21] R. H. Keshavan and S. Oh, *A Gradient Descent Algorithm on the Grassman Manifold for Matrix Completion*, preprint, arXiv:0910.5260 [cs.NA], 2009.

[22] S. G Krantz and H. R. Parks, *The Implicit Function Theorem: History, Theory, and Applications*, Birkhäuser, Boston, 2002.

[23] R. M. Larsen, *PROPACK - software for large and sparse svd calculations*, http://soi.stanford.edu/~rmunk/PROPACK/.

[24] J. M. Lee, *Introduction to Smooth Manifolds*, Grad. Texts in Math. 218, Springer, New York, 2003.

[25] S. Ma, D. Goldfarb, and L. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Math. Program., 128 (2011), pp. 321–353.

[26] R. Mazumder, T. Hastie, and R. Tibshirani, *Spectral regularization algorithms for learning large incomplete matrices*, J. Mach. Learn. Res., 11 (2010), pp. 2287–2322.

[27] R. Meka, P. Jain, and I. S. Dhillon, *Guaranteed Rank Minimization via Singular Value Projection*, preprint, arXiv:0909.5457 [cs.LG], 2009.
[28] G. Meyer, *Geometric Optimization Algorithms for Linear Regression on Fixed-Rank Matrices*, Ph.D. thesis, University of Liège, Liège, Belgium, 2011.
[29] G. Meyer, S. Bonnabel, and R. Sepulchre, *Regression on fixed-rank positive semidefinite matrices: A Riemannian approach*, J. Mach. Learn. Res., 12 (2010), pp. 593–625.
[30] G. Meyer, S. Bonnabel, and R. Sepulchre, *Linear regression under fixed-rank constraints: A Riemannian approach*, in Proceedings of the 28th International Conference on Machine Learning (ICML 2011), Bellevue, WA, 2011.
[31] B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre, *Fixed-Rank Matrix Factorizations and Riemannian Low-Rank Optimization*, preprint, arXiv:1209.0430 [cs.LG], 2012.
[32] B. Mishra, G. Meyer, and R. Sepulchre, *Low-rank optimization for distance matrix completion*, in Proceedings of the 50th IEEE Conference on Decision and Control, IEEE, Piscataway, NJ, 2011, pp. 4455–4460.
[33] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Appl. Optim. 87, Kluwer Academic, Boston, 2003.
[34] M.-Y. Park and T. Hastie, *Regularization Path Algorithms for Detecting Gene Interactions*, Technical report, Department of Statistics, Stanford University, Palo Alto, CA, 2006.
[35] B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 52 (2010), pp. 471–501.
[36] L. Simonsson and L. Eldén, *Grassmann algorithms for low rank approximation of matrices with missing values*, BIT, 50 (2010), pp. 173–191.
[37] S. T. Smith, *Covariance, subspace, and intrinsic Cramér-Rao bounds*, IEEE Trans. Signal Process., 53 (2005), pp. 1610–1630.
[38] N. Srebro and T. Jaakkola, *Weighted low-rank approximations*, in Proceedings of the 20th International Conference on Machine Learning (ICML), AAAI Press, Menlo Park, CA, 2003, pp. 720–727.
[39] K. C. Toh and S. Yun, *An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems*, Pacific J. Optim., 6 (2010), pp. 615–640.
[40] B. Vandereycken, *Low-rank matrix completion by Riemannian optimization*, SIAM J. Optim., 23 (2013), pp. 1214–1236.
[41] M. Vounou, T. E. Nichols, G. Montana, and Alzheimer's Disease Neuroimaging Initiative, *Discovering genetic associations with high-dimensional neuroimaging phenotypes: A sparse reduced-rank regression approach*, Neuroimage, 53 (2010), pp. 1147–1159.
[42] Z. Wen, W. Yin, and Y. Zhang, *Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm*, Math. Program. Comput., 4 (2012), pp. 333–361.
[43] M. Yuan, A. Ekici, Z. Lu, and R. D. C. Monteiro, *Dimension reduction and coefficient estimation in multivariate linear regression*, J. Roy. Stat. Soc. Ser. B Methodol., 69 (2007), pp. 329–346.