A unified framework for structured low-rank matrix learning

Pratik Jawanpuria jawanpur@amazon.com
Amazon.com
Bamdev Mishra bamdevm@amazon.com
Amazon.com

Abstract

We propose a novel optimization framework for learning a low-rank matrix which is also constrained to lie in a linear subspace. Exploiting the duality theory, we present a factorization that decouples the low-rank and structural constraints onto separate factors. The optimization problem is formulated on the Riemannian spectrahedron manifold, where the Riemannian framework allows to develop computationally efficient conjugate gradient and trust-region algorithms. Empirically, our algorithms outperform state-of-the-art on various applications.

1 Introduction

Our focus in this paper is on learning structured low-rank matrices with the formulation

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times T}} \ \frac{1}{2}R(\mathbf{W}) + CL(\mathbf{W},\mathbf{Y}), \quad \text{subject to } \mathbf{W}\in\mathcal{D}, \tag{1}$$

where $\mathbf{Y}\in\mathbb{R}^{d\times T}$ is a given matrix, $L:\mathbb{R}^{d\times T}\times\mathbb{R}^{d\times T}\to\mathbb{R}$ is a loss function, R is a low-rank promoting regularizer and $C>0$ is the cost parameter. $\mathcal{D}$ is the *linear* subspace corresponding to structural constraints, $\mathcal{D}:=\{\mathbf{W}:\mathcal{A}(\mathbf{W})=\mathbf{0}\}$, where $\mathcal{A}:\mathbb{R}^{d\times T}\to\mathbb{R}^n$ is a linear map.

Low-rank matrices are commonly learned for matrix completion [1], multivariate regression [2], etc. In addition to the low-rank constraint, other structural constraints may exist, e.g., entry-wise non-negative/bounded constraints [3]. Several linear dynamical models require learning a low-rank *Hankel* matrix [4, 5]. A Hankel matrix has all its anti-diagonal entries to be the same. In robust matrix completion [6], the sparse structure is modeled effectively by the ℓ_1-loss function [7, 8].

We propose a unified optimization framework for (1), which is suitable for a variety of loss functions L, structural constraints $\mathcal{D}$, and is scalable. Using the duality theory, we present a novel modeling of structured low-rank matrix $\mathbf{W}$ as $\mathbf{W}=\mathbf{U}\mathbf{U}^\top(\mathbf{Z}+\mathbf{A})$, where $\mathbf{U}\in\mathbb{R}^{d\times r}$ and $\mathbf{Z},\mathbf{A}\in\mathbb{R}^{d\times T}$. Our factorization naturally decouples the low-rank and structural constraints on $\mathbf{W}$: the low-rank is enforced with $\mathbf{U}$, the structural constraint is modeled by $\mathbf{A}$, and the loss specific structure is modeled by $\mathbf{Z}$. The separation of constraints onto separate factors makes the optimization conceptually simpler.

Our approach leads to an optimization problem on the *Riemannian spectrahedron* manifold. We exploit the Riemannian framework to develop computationally efficient conjugate gradient and trust-region algorithms. The proposed algorithms outperform state-of-the-art in standard, robust and non-negative matrix completion problems as well as low-rank Hankel matrix learning applications.

1.1 Related work

Matrix completion: [9, 10] proposes singular value thresholding and active learning algorithms for (1) with $R(\mathbf{W})=\|\mathbf{W}\|_*$ (trace-norm regularizer) and without the constraint $\mathbf{W}\in\mathcal{D}$. Fixed-rank approaches [11, 12, 13, 14, 15] learn a low-rank matrix by fixing the rank explicitly.

Robust matrix completion: [7] proposes to use ℓ_1-loss as the loss function along with the low-rank constraint on $\mathbf{W}$ for the robust matrix completion problem. [8] employs the pseudo-Huber loss as a proxy for the non-smooth ℓ_1-loss and develop a large-scale Riemannian conjugate gradient algorithm.

OPTML 2017: 10th NIPS Workshop on *Optimization for Machine Learning* (NIPS 2017).

Non-negative matrix completion: Certain recommender system and image completion based applications desire matrix completion with non-negative entries [3]. Recently, [3] proposes a large-scale alternating direction method of multipliers (ADMM) algorithm.

Hankel matrix learning: [4] uses $R(\mathbf{W}) = \|\mathbf{W}\|_*$ in (1) to learn a low-rank Hankel matrix (enforced by $\mathcal{D}$) with the ADMM approaches. [16] also employs the $\|\mathbf{W}\|_*$ regularization, but relax the Hankel constraints with a corresponding penalty term in the objective function. [5] learns a Hankel matrix by fixing the rank *a priori* and strictly enforcing the structural constraints.

2 Novel formulation for structured low-rank matrix learning

A few notations first. The set of $d \times d$ positive semi-definite matrices with unit trace is denoted by $\mathcal{P}^d$. The pseudoinverse of a matrix Θ is represented as $\Theta^\dagger$ and $\text{range}(\Theta) = \{\Theta z : z \in \mathbb{R}^d\}$.

We study a variant of problem (1) in which we employ a variational characterization of $R(\mathbf{W}) = \|\mathbf{W}\|_*^2$ [17, Theorem 4.1]: $\|\mathbf{W}\|_*^2 = \min_{\Theta \in \mathcal{P}^d} \left\langle \Theta^\dagger \mathbf{W}, \mathbf{W} \right\rangle$, subject to $\text{range}(\mathbf{W}) \subseteq \text{range}(\Theta)$. Hence, our the propose primal formulation for structured low-rank matrix learning is as follows:

$$\min_{\Theta \in \mathcal{P}^d,\, \text{range}(\mathbf{W}) \subseteq \text{range}(\Theta)} \; \min_{\mathbf{W} \in \mathbb{R}^{d \times T}} \; \frac{1}{2} \left\langle \Theta^\dagger \mathbf{W}, \mathbf{W} \right\rangle + CL(\mathbf{Y}, \mathbf{W}), \quad \text{subject to } \mathcal{A}(\mathbf{W}) = \mathbf{0}. \tag{2}$$

The matrix Θ learned in (2) is low-rank because $\Theta \in \mathcal{P}^d$. In addition, it can be proved that the rank of Θ and $\mathbf{W}$ are equal at optimality. Hence, the low-rank constraint on $\mathbf{W}$ is *transferred* to Θ in (2). In the following, we propose a dual formulation of (2) that provides further insights into the optimal solution of (2) and is suitable in large-scale structured matrix learning.

Theorem 1 *Let L^* be the Fenchel conjugate function of the loss: $L : \mathbb{R}^{d \times T} \to \mathbb{R}$, $v \mapsto L(\mathbf{Y}, v)$ and let $\mathcal{A}^* : \mathbb{R}^n \to \mathbb{R}^{d \times T}$ be the adjoint of $\mathcal{A}$. The dual problem of (2) with respect to $\mathbf{W}$ is*

$$\min_{\Theta \in \mathcal{P}^d} \; \max_{\mathbf{Z} \in \mathbb{R}^{d \times T},\, s \in \mathbb{R}^n} \; -CL^*(-\mathbf{Z}/C) - \langle \Theta(\mathbf{Z} + \mathcal{A}^*(s)), \mathbf{Z} + \mathcal{A}^*(s) \rangle / 2. \tag{3}$$

If $\{\bar{\Theta}, \bar{\mathbf{Z}}, \bar{s}, \}$ is an optimal solution of (3), the optimal solution $\bar{\mathbf{W}}$ of (2) is $\bar{\mathbf{W}} = \bar{\Theta}(\bar{\mathbf{Z}} + \mathcal{A}^*(\bar{s}))$.

From the above theorem, we can observe that in an optimal $\bar{\mathbf{W}}$, the low-rank constraint is enforced through $\bar{\Theta}$, the loss-specific structure (encoded in L^*) is enforced through $\bar{\mathbf{Z}}$, and the structural constraint is enforced through $\mathcal{A}^*(\bar{s})$. Overall, such a decoupling of constraints onto separate variables facilitates the use of simpler optimization techniques.

An algorithm for (3) need not produce intermediate iterates with low-rank Θ matrix. For large-scale optimization, this observation motivates a fixed-rank parameterization of Θ as discussed below.

We model $\Theta \in \mathcal{P}^d$ as a rank r matrix as follows: $\Theta = \mathbf{U}\mathbf{U}^\top$, where $\mathbf{U} \in \mathbb{R}^{d \times r}$ and $\|\mathbf{U}\|_F = 1$. The proposed modeling has two-fold benefits in large-scale problems, where $r \ll \min\{d, T\}$ is a common setting. First, the $\Theta \in \mathcal{P}^d$ constraint is always satisfied, thereby saving the costly projection operations to ensure $\Theta \in \mathcal{P}^d$. Second, the dimension of the search space of (3) with $\Theta = \mathbf{U}\mathbf{U}^\top$ is $rd - 1 - r(r-1)/2$, which is much lower than the dimension $(d(d+1)/2 - 1)$ of $\Theta \in \mathcal{P}^d$.

Instead of solving a minimax objective directly, as in (3), we solve a minimization problem after incorporating the $\Theta = \mathbf{U}\mathbf{U}^\top$ parameterization as follows:

$$\min_{\mathbf{U} \in \mathbb{R}^{d \times r}, \|\mathbf{U}\|_F = 1} \; g(\mathbf{U}), \quad \text{where} \tag{4}$$

$$g(\mathbf{U}) \coloneqq \max_{\mathbf{Z} \in \mathbb{R}^{d \times T}, s \in \mathbb{R}^n} \; -CL^*(-\mathbf{Z}/C) - \left\|\mathbf{U}^\top(\mathbf{Z} + \mathcal{A}^*(s))\right\|_F^2 / 2. \tag{5}$$

An important outcome of the above modeling is that the *expression* for the gradient of $g(\mathbf{U})$ in (4) is *independent* of the application at hand (refer Lemma 1). The application specific information in (5) is encoded only through the *values* of variables $\mathbf{Z}$ and s, which are used to compute the gradient of $g(\mathbf{U})$. This allows the development of a *unified* optimization framework for various low-rank matrix learning problems. Finally, the matrix $\mathbf{W}$ is learned as $\mathbf{U}\mathbf{U}^\top(\mathbf{Z} + \mathcal{A}^*(s))$. The following theorem provides the duality gap optimality criterion of any *feasible* solution $\hat{\mathbf{U}}$ of (4).

Algorithm 1 Proposed first- and second-order algorithms for (4)

Input: $\mathbf{Y}$, rank r, regularization parameter C.
Initialize $\mathbf{U} \in \mathcal{S}_r^d$.
repeat
1: Solve for $\{\mathbf{Z}, s\}$ by computing $g(\mathbf{U})$ in (5).
2: Compute $\nabla_{\mathbf{U}} g(\mathbf{U})$ as given in Lemma 1.
3: Riemannian CG step: compute a conjugate direction $\mathbf{V}$ and step size α, using of $\nabla_{\mathbf{U}} g(\mathbf{U})$.
3: Riemannian TR step: compute a search direction $\mathbf{V}$ that minimizes the trust region sub-problem, using $\nabla_{\mathbf{U}} g(\mathbf{U})$ and its directional derivative. Step size $\alpha = 1$.
4: Update: $\mathbf{U} = (\mathbf{U} + \alpha\mathbf{V}) / \|\mathbf{U} + \alpha\mathbf{V}\|_F$ (retraction step)
until convergence
Output: $\{\mathbf{U}, \mathbf{Z}, s\}$ and $\mathbf{W} = \mathbf{U}\mathbf{U}^\top(\mathbf{Z} + \mathcal{A}^*(s))$.

Theorem 2 *Let $\hat{\mathbf{U}}$ be a feasible solution of (4), $\{\hat{\mathbf{Z}}, \hat{s}\}$ be an optimal solution of the convex problem (5) at $\hat{\mathbf{U}}$, and σ_1 be the maximum singular of $\hat{\mathbf{Z}} + \mathcal{A}^*(\hat{s})$. A candidate solution for (3) is $\{\hat{\Theta}, \hat{\mathbf{Z}}, \hat{s}\}$, where $\hat{\Theta} = \hat{\mathbf{U}}\hat{\mathbf{U}}^\top$. The associated duality gap (Δ) is given by $\Delta = (\sigma_1^2 - \left\|\hat{\mathbf{U}}^\top(\hat{\mathbf{Z}} + \mathcal{A}^*(\hat{s}))\right\|^2)/2$.*

The cost of computing σ_1 is computationally cheap as it requires only a few *power iteration* updates.

3 Optimization on spectrahedron manifold

The matrix $\mathbf{U}$ lies in, what is popularly known as, the *spectrahedron* manifold $\mathcal{S}_r^d := \{\mathbf{U} \in \mathbb{R}^{d\times r} : \|\mathbf{U}\|_F = 1\}$, which has the structure of a compact Riemannian quotient manifold [2]. The quotient structure takes the rotational invariance of the constraint $\|\mathbf{U}\|_F = 1$ into account. The Riemannian optimization framework generalizes various classical first- and second-order Euclidean algorithms to manifolds and provide concrete convergence guarantees [18, 2, 19].

We implement the Riemannian conjugate gradient (CG) and trust-region (TR) algorithms for (4). These require the notions of the *Riemannian gradient* (first-order derivative of $g(\mathbf{U})$ on the manifold), *Riemannian Hessian* along a search direction (the *covariant* derivative of the Riemannian gradient along a tangential direction on the manifold), and the *retraction* operator (that ensures that we always stay on the manifold). The above require computations of the Euclidean gradient ($\nabla_{\mathbf{U}} g(\mathbf{U})$) and the directional derivative of this gradient along a given direction, expressed in the following lemma.

Lemma 1 *Let $\{\hat{\mathbf{Z}}, \hat{s}\}$ be an optimal solution of the convex problem (5) at $\mathbf{U}$. Then,*

$$\nabla_{\mathbf{U}} g(\mathbf{U}) = -(\hat{\mathbf{Z}} + \mathcal{A}^*(\hat{s}))(\hat{\mathbf{Z}} + \mathcal{A}^*(\hat{s}))^\top \mathbf{U}.$$

Let $\mathrm{D}\nabla_{\mathbf{U}} g(\mathbf{U})[\mathbf{V}]$ denote the directional derivative of the gradient $\nabla_{\mathbf{U}} g(\mathbf{U})$ along $\mathbf{V} \in \mathbb{R}^{d\times r}$. Let $\{\dot{\mathbf{Z}}, \dot{s}\}$ denote the directional derivative of $\{\mathbf{Z}, s\}$ along $\mathbf{V}$ at $\{\hat{\mathbf{Z}}, \hat{s}\}$. Then,

$$\mathrm{D}\nabla_{\mathbf{U}} g(\mathbf{U})[\mathbf{V}] = (\dot{\mathbf{Z}} + \mathcal{A}^*(\dot{s}))(\hat{\mathbf{Z}} + \mathcal{A}^*(\hat{s}))^\top \mathbf{U} + (\hat{\mathbf{Z}} + \mathcal{A}^*(\hat{s}))\Big((\dot{\mathbf{Z}} + \mathcal{A}^*(\dot{s}))^\top \mathbf{U} - (\hat{\mathbf{Z}} + \mathcal{A}^*(\hat{s}))^\top \mathbf{V}\Big).$$

Riemannian CG algorithm: It computes the Riemannian *conjugate* gradient direction by employing $\nabla_{\mathbf{U}} g(\mathbf{U})$ (Lemma 1). *Armijo* line search is employed to compute the step-size.

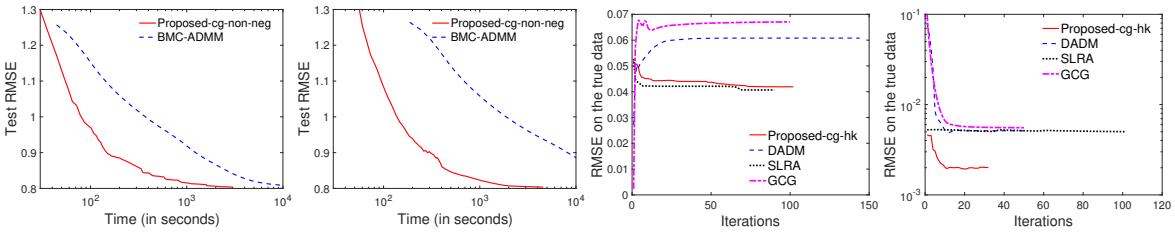
Riemannian TR algorithm: It solves a Riemannian trust-region *sub-problem* (in a neighborhood) at every iteration. It makes use of $\nabla_{\mathbf{U}} g(\mathbf{U})$ and its directional derivative $\mathrm{D}\nabla_{\mathbf{U}} g(\mathbf{U})[\mathbf{V}]$ (Lemma 1).

Overall algorithm: Algorithm 1 summarizes the proposed first- and second-order algorithms for solving (4). Our approach can be readily extended to a stochastic setting, e.g., when $\mathcal{A}(\mathbf{W}) = \mathbf{0}$ imposes column-wise constraints and the columns are streamed one by one.

Computational complexity: The spectrahedron manifold operations cost $O(dr + r^3)$. The additional computational cost of solving (5) for specific problems is as follows: (a) Matrix completion: (5) is a least-squares problem which can be solved in closed form in $O(|\Omega r^2|)$; (b) Robust matrix completion: (5) can be solved using dual co-ordinate descent algorithm [20] in $O(|\Omega|r^2)$; (c) Non-negative matrix completion: (5) can be solved using the non-negative least-squares algorithm [21] in $O(dTr + |\Omega|r^2)$; and (d) Hankel matrix learning: (5) can be solved using preconditioned gradient descent algorithm in $O(dTr)$.

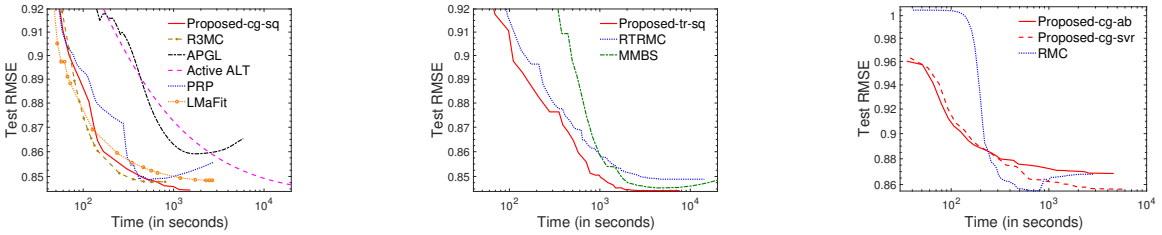
4 Experiments

We perform experiments on four different applications.



(a) MovieLens10m (b) MovieLens20m (c) Small data set (d) Large data set

Figure 1: (a)&(b) Evolution of test RMSE on non-negative matrix completion problems. (c)&(d) Performance on different stochastic system realization problems — learning low-rank Hankel matrices.



(a) First-order algorithms (b) Second-order algorithms (c) First-order algorithms

Figure 2: Evolution of test RMSE on the Netflix data set. (a)&(b) Comparison of matrix completion algorithms; (c) Comparison of robust matrix completion algorithms.

Non-negative matrix completion: We compare our first-order algorithm (Proposed-cg-non-neg) against BMC-ADMM [3] on MovieLens10m (ML10m) and MovieLens20m (ML20m) datasets. The rank for both is set as 10. Figures 1(a)&(b) plot the evolution of the test RMSE with training time. Proposed-cg-non-neg outperform BMC-ADMM and converge to the best test RMSE.

Hankel matrix learning: We compare our first-order algorithm (Proposed-cg-hk) with GCG [16], SLRA [22, 23], and DADM [4]. In small-scale experiment, we follow [4, 16] to generate the data, with $d = 21$, $T = 100$, and $r = 10$. In large-scale experiment, we follow [22, 23] to generate the data, with $d = 1000$, $T = 10000$, and $r = 5$. Figure 1(c)&(d) shows the variation of RMSE with respect to true data across iterations for the two experiments. We observe that our algorithm outperform GCG and DADM, and obtain significantly better true RMSE than SLRA in the large-scale experiment.

Matrix completion: We compare our first- and second-order algorithms (Proposed-cg-sq and Proposed-tr-sq, respectively) against several solvers: APGL [9], Active ALT [10], R3MC [14], LMaFit [12], MMBS [24], RTRMC [11, 25], and PRP [15]. The rank (or maximum rank) for all is set to 10. Figures 2(a)&(b) display the evolution of test RMSE on the Netflix data set for first- and second-order algorithms, respectively. Proposed-cg-sq is among the most efficient first-order method and Proposed-tr-sq is the best second-order method.

Robust matrix completion: We develop two first-order robust algorithms: Proposed-cg-ab (with ℓ_1-loss), and Proposed-cg-svr (with ϵ-SVR loss), and compare against RMC [8]. Figure 2(c) displays the results on the Netflix data set. We observe that both our algorithms scale effortlessly even with non-smooth loss functions, with Proposed-cg-svr obtaining the lowest test RMSE.

5 Conclusion

We have proposed a novel factorization for structured low-rank matrix learning problems, which stems from the application of duality theory and rank-constrained parameterization of positive semi-definite matrices. This allows to develop a conceptually simpler and unified optimization framework for various applications. State-of-the-art performance of our algorithms on several applications demonstrate the efficacy of our approach.

Acknowledgement

We thank Léopold Cambier, Ivan Markovsky, and Konstantin Usevich for useful discussions on the work.

References

[1] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[2] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

[3] H. Fang, Z. Zhen, Y. Shao, and C.-J. Hsieh. Improved bounded matrix completion for large-scale recommender systems. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1654–1660, 2017.

[4] M. Fazel, P. T. Kei, D. Sun, and P. Tseng. Hankel matrix rank minimization with applications to system identification and realization. *SIAM Journal on Matrix Analysis and Applications*, 34(3):946–977, 2013.

[5] I. Markovsky and K. Usevich. Structured low-rank approximation with missing data. *SIAM Journal on Matrix Analysis and Applications*, 34(2):814–830, 2013.

[6] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Neural Information Processing Systems conference (NIPS)*, 2009.

[7] J. He, L. Balzano, and A. Szlam. Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1568–1575, 2012.

[8] L. Cambier and P. A. Absil. Robust low-rank matrix completion by Riemannian optimization. *SIAM J. Sci. Comput.*, 38(5):S440—S460, 2016.

[9] K. C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific Journal of Optimization*, 6(3):615–640, 2010.

[10] C.-J. Hsieh and P. A. Olsen. Nuclear norm minimization via active subspace selection. In *International Conference on Machine learning (ICML)*, 2014.

[11] N. Boumal and P.-A. Absil. RTRMC: A Riemannian trust-region method for low-rank matrix completion. In *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 406–414, 2011.

[12] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, 2012.

[13] B. Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

[14] B. Mishra and R. Sepulchre. R3MC: A Riemannian three-factor algorithm for low-rank matrix completion. In *Proceedings of the 53rd IEEE Conference on Decision and Control (CDC)*, pages 1137–1142, 2014.

[15] M. Tan, S. Xiao, J. Gao, D. Xu, A. van den Hengel, and Q. Shi. Proximal riemannian pursuit for large-scale trace-norm minimization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] A. W. Yu, Ma W., Yu Y., Carbonell J. G., and Sra S. Efficient structured matrix rank minimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[17] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Neural Information Processing Systems conference (NIPS)*, 2006.

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

[19] H. Sato and T. Iwai. A new, globally convergent Riemannian conjugate gradient method. *Optimization: A Journal of Mathematical Programming and Operations Research*, 64(4):1011–1031, 2013.

[20] C. H. Ho and C. J. Lin. Large-scale linear support vector regression. *Journal of Machine Learning Research*, 13(1):3323–3348, 2012.

[21] D. Kim, S. Sra, and I. S. Dhillon. A non-monotonic method for large-scale non-negative least squares. *Optimization Methods and Software*, 28(5):1012–1039, 2013.

[22] I. Markovsky. Recent progress on variable projection methods for structured low-rank approximation. *Signal Processing*, 96(Part B):406–419, 2014.

[23] I. Markovsky and K. Usevich. Software for weighted structured low-rank approximation. *J. Comput. Appl. Math.*, 256:278–292, 2014.

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

[25] N. Boumal and P.-A. Absil. Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. *Linear Algebra and its Applications*, 475:200–239, 2015.