A Fast Augmented Lagrangian Algorithm for Learning Low-Rank Matrices

Ryota Tomioka[1] TOMIOKA@MIST.I.U-TOKYO.ACJP
Taiji Suzuki[1] S-TAIJI@STAT.T.U-TOKYO.AC.JP
Masashi Sugiyama[2] SUGI@CS.TITECH.AC.JP
Hisashi Kashima[1] KASHIMA@MIST.I.U-TOKYO.AC.JP

[1]Department of Mathematical Informatics, The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan.

[2]Department of Computer Science, Tokyo Institute of Technology, Meguro-ku, Tokyo 152-8552, Japan.

Abstract

We propose a general and efficient algorithm for learning low-rank matrices. The proposed algorithm converges super-linearly and can keep the matrix to be learned in a compact factorized representation without the need of specifying the rank beforehand. Moreover, we show that the framework can be easily generalized to the problem of learning multiple matrices and general spectral regularization. Empirically we show that we can recover a 10,000×10,000 matrix from 1.2 million observations in about 5 minutes. Furthermore, we show that in a brain-computer interface problem, the proposed method can speed-up the optimization by two orders of magnitude against the conventional projected gradient method and produces more reliable solutions.

1. Introduction

Explanation in addition to good prediction is valuable in many application areas of machine learning.

Feature extraction/selection can be used as a preprocessing step before applying a classification algorithm to obtain interpretable results. Here our aim is to also *learn* feature extractors *jointly* with a classifier as a low-rank matrix. Learning low-rank matrices has been studied in various contexts, namely, matrix completion (Srebro et al., 2005), multi-task learning (Argyriou et al., 2007), multi-class classification (Amit et al., 2007), and classification of matrices (Tomioka & Aihara, 2007). In the matrix completion problem, the low-rank decomposition of the "hidden" matrix corresponds to representation of the row/column objects in a compressed low-dimensional feature space (Abernethy et al., 2009). In the multi-task/multi-class problems the estimated low-rank matrix squeezes the input data through a low-dimensional feature space shared among several tasks. In the matrix classification problem, we learn a (small) set of row/column feature extractors, which for instance correspond to learning spacial/temporal features.

The trace-norm regularization (see Fazel et al. (2001) and all the references mentioned above) is a principled approach to learning low-rank matrices through convex optimization problems. It can be considered as a generalization of ℓ_1-regularization (Tibshirani, 1996; Chen et al., 1998). In fact, it is shown in Candes & Recht (2009) that under some conditions, a low-rank matrix can be perfectly recovered from incomplete observations.

However, matrix-learning problems with spectral regularizations (which include the trace norm as a special case) are significantly harder to optimize than conventional vector-based learning problems, because of the size of the problem and the possible non-differentiability of the regularizers. For example, when there are R tasks of size C, the size of the multi-task learning problem is RC. Often a non-convex approximation that factorizes the matrix to be learned at a pre-specified rank is employed for the trace-norm minimization (Abernethy et al., 2009; Weimer et al., 2008). However, it is not clear how this approach can be extended to general spectral regularizations. Alternatively a Majorization-Minimization (MM) algorithm, which iteratively minimizes a quadratic upper bound

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

of the trace norm, is proposed in Argyriou et al. (2007) (see also Figueiredo et al. (2007) for detailed discussion on MM algorithms). Although the MM approach has the potential to be extended to more general spectral regularizations (see Argyriou et al. (2008)), it requires solving (full-rank) ridge-regularized minimization, which can be demanding from a computational/storage point of view.

Recently researchers started to focus on spectral soft-thresholding-based approaches, which have the advantages of both convexity (no need to fix the rank beforehand) and low-rank preserving property. Cai et al. (2008) proposed the singular-value thresholding (SVT) algorithm for the problem of low-rank matrix completion. Ji & Ye (2009) proposed the accelerated gradient (AG) method for supervised learning with trace norm regularization. These methods update the matrix $\boldsymbol{W}^t \in \mathbb{R}^{R\times C}$ to be learned as follows:

$$\boldsymbol{W}^{t+1} := \mathrm{ST}_{\lambda\eta_t}\left(\boldsymbol{W}^t + \eta_t \boldsymbol{Y}^t\right), \tag{1}$$

where λ is a regularization constant, η_t is a step-size, and $\boldsymbol{Y}^t$ is the direction of descent (which differs from one algorithm to another). In addition, ST_λ denotes the spectral soft-threshold operator (Cai et al., 2008; Ji & Ye, 2009) defined as follows:

$$\mathrm{ST}_\lambda(\boldsymbol{W}) = \boldsymbol{U}\max(\boldsymbol{S} - \lambda \boldsymbol{I}, 0)\boldsymbol{V}^\top, \tag{2}$$

where $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top$ is the singular-value decomposition (SVD) of $\boldsymbol{W}$. Both Cai et al. (2008) and Ji & Ye (2009) use only first-order information for constructing the descent direction $\boldsymbol{Y}^t$. Generally speaking, first-order methods perform many steps that have small costs. In large-scale matrix-learning problems, however, the cost of one soft-thresholding operation (SVD) can be too high to perform many times.

In this paper, we propose a new soft-threshold-based approach that uses an improved descent direction $\boldsymbol{Y}^t$. The descent direction is obtained by solving a smooth minimization problem. The resulting algorithm converges super-linearly requiring a small number of SVDs. Moreover, it can be run preserving only the set of "active" singular-values/vectors. Thus it can be applied to large problems for which the full matrix cannot fit in the RAM. Moreover, we extend the proposed approach to general spectral regularization and learning multiple matrices.

This paper is organized as follows. In Sec. 2, we introduce our algorithm for the trace-norm regularization problem. We show the convergence result and discuss how to perform the computation preserving a factorized representation. In Sec. 3 we show that the proposed algorithm can be generalized to a class of spectral regularizers. In Sec. 4, we demonstrate the efficiency and high precision of the proposed algorithm in a simulated matrix completion problem, and classification of electroencephalography (EEG) time-series in the context of brain-computer interface (BCI). Finally we conclude the paper in Sec. 5.

2. Trace Norm Regularization

2.1. Matrix-Learning Problem

Given a loss function f_ℓ, let us consider the regularized matrix-learning problem of the following form:

$$\underset{\boldsymbol{W}\in\mathbb{R}^{R\times C}, b\in\mathbb{R}}{\text{minimize}} \quad \underbrace{f_\ell(\mathcal{A}(\boldsymbol{W}) + b\mathbf{1}_m) + \lambda\|\boldsymbol{W}\|_*}_{=f(\boldsymbol{W},b)}, \tag{3}$$

where $\boldsymbol{W}$ is an $R \times C$ matrix to be learned, $\mathcal{A} : \mathbb{R}^{R\times C} \to \mathbb{R}^m$ is a linear (observation) operator, $b \in \mathbb{R}$ is an unregularized bias term, and $\mathbf{1}_m$ is an m-dimensional vector with all one; $\lambda \geq 0$ is the regularization constant. Note that the operator $\mathcal{A}$ can be highly structured; for example, in a matrix completion problem, $\mathcal{A}$ picks m elements of $\boldsymbol{W}$ and vectorizes them into an m-dimensional vector. We assume that the loss function f_ℓ is convex and its convex conjugate[1] f_ℓ^* is twice differentiable. Finally the regularization term is the trace norm of $\boldsymbol{W}$, which is defined as follows:

$$\|\boldsymbol{W}\|_* = \sum_{j=1}^{r} \sigma_j(\boldsymbol{W}), \tag{4}$$

where $\sigma_j(\boldsymbol{W})$ is the j-th singular-value of $\boldsymbol{W}$ and r is the rank of $\boldsymbol{W}$.

2.2. Dual Augmented Lagrangian Algorithm

We propose the Matrix-DAL (M-DAL) algorithm, which is a extension of the dual augmented Lagrangian algorithm (Tomioka & Sugiyama, 2009; Tomioka et al., 2009) to the matrix-learning problem (3). The algorithm can be described as follows:

1. Suitably initialize $(\boldsymbol{W}^1, b^1)$ and choose a sequence of step-sizes $\eta_1 < \eta_2 < \cdots$.

2. Repeat until the relative duality gap (RDG) is less than some tolerance ϵ:

$$\boldsymbol{W}^{t+1} = \mathrm{ST}_{\lambda\eta^t}\left(\boldsymbol{W}^t + \eta^t \mathcal{A}^\top(\boldsymbol{\alpha}^t)\right), \tag{5}$$

$$b^{t+1} = b^t + \eta^t {\mathbf{1}_m}^\top \boldsymbol{\alpha}^t, \tag{6}$$

where $\mathcal{A}^\top : \mathbb{R}^m \to \mathbb{R}^{R\times C}$ is the adjoint operator of $\mathcal{A}$ and $\mathrm{ST}_{\lambda\eta_t}$ is the soft-threshold operation in Eq. (2); $\boldsymbol{\alpha}^t$ is the minimizer of the inner-objective function $\varphi_t(\boldsymbol{\alpha})$

[1] Convex conjugate of a function f is defined as $f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x}\in\mathbb{R}^n}\left(\boldsymbol{y}^\top\boldsymbol{x} - f(\boldsymbol{x})\right)$. If f is a closed proper convex function, $f^{**} = f$.

defined as follows:

$$\varphi_t(\boldsymbol{\alpha}) = f_\ell^*(-\boldsymbol{\alpha}) + \frac{1}{2\eta^t}\left(b^t + \eta^t \mathbf{1}_m{}^\top \boldsymbol{\alpha}\right)^2 + \frac{1}{2\eta^t}\left\|\mathrm{ST}_{\lambda\eta^t}(\boldsymbol{W}^t + \eta^t \mathcal{A}^\top(\boldsymbol{\alpha}))\right\|_{\mathrm{fro}}^2, \quad (7)$$

where f_ℓ^* is the convex conjugate of f_ℓ and $\|\cdot\|_{\mathrm{fro}}$ denotes the Frobenius norm.

Note that the proposed update (Eqs. (5) and (6)) takes a generalized form of Eq. (1) because the bias term b can be considered to be thresholded by a regularization constant zero in Eq. (2).

We can show that the inner objective (Eq. (7)) is twice differentiable almost everywhere. In fact, we show in Sec. 2.4 that the gradient of the $\|\mathrm{ST}_\lambda(\cdot)\|_{\mathrm{fro}}^2$ term in Eq. (7) is the soft-threshold operation $\mathrm{ST}_\lambda(\cdot)$, which is continuous. The soft-threshold operator is nondifferentiable when a singular-value of $\boldsymbol{W}$ crosses λ. However, this unlucky situation can be avoided by a small perturbation of λ. Therefore, in general the solution of Eq. (7) lies on a differentiable point. This motivates us to use a Newton-type method for the minimization of Eq. (7). More specifically, we use the L-BFGS quasi-Newton method (Nocedal & Wright, 2006) for large scale problems (see Sec. 4.1) and the Newton method for medium scale problems (see Sec. 4.2 and 4.3).

2.3. Super-Linear Convergence

In this subsection, we show that the proposed M-DAL algorithm converges super-linearly. More specifically, we show that the distance from the solution obtained after t-outer iterations $(\boldsymbol{W}^t, b^t)$ to the true minimizer $(\boldsymbol{W}^*, b^*)$ of Eq. (3) drops faster than exponential as follows:

$$\sqrt{\|\boldsymbol{W}^t - \boldsymbol{W}^*\|_{\mathrm{fro}}^2 + (b^t - b^*)^2} \le \mathrm{O}(\exp(-c^t)),$$

where c^t is a sequence that increases faster than linear.

The M-DAL algorithm (Eqs. (5)-(7)) can be understood in two ways. One way is to think of it as an augmented Lagrangian (AL) method (Rockafellar, 1976; Bertsekas, 1982) (also known as the method of multipliers (Powell, 1969; Hestenes, 1969)) on the dual problem of Eq. (3) (Tomioka & Sugiyama, 2009). Another way is to think of it as a *proximal minimization* (PM) method in the primal (Tomioka et al., 2009), which iteratively solves the following problem:

$$(\boldsymbol{W}^{t+1}, b^{t+1}) = \operatorname*{argmin}_{\substack{\boldsymbol{W}\in\mathbb{R}^{R\times C}\\ b\in\mathbb{R}}} \left(f(\boldsymbol{W}, b) + \frac{1}{2\eta^t}(b - b^t)^2 + \frac{1}{2\eta^t}\|\boldsymbol{W} - \boldsymbol{W}^t\|_{\mathrm{fro}}^2 \right), \quad (8)$$

where $f(\boldsymbol{W}, b)$ is the objective function in Eq. (3). Equations (5)-(7) can be derived from Eq. (8) by defining $\boldsymbol{w} = (\mathrm{vec}(\boldsymbol{W})^\top, b)^\top \in \mathbb{R}^{RC+1}$ ($\mathrm{vec}(\cdot)$ denotes the column-wise concatenation of a matrix), $\phi_\lambda(\boldsymbol{w}) = \lambda\|\boldsymbol{W}\|_*$, $\overline{\mathrm{ST}}_\lambda(\boldsymbol{w}) = (\mathrm{vec}(\mathrm{ST}_\lambda(\boldsymbol{W}))^\top, b)^\top$, and following the derivation in Tomioka et al. (2009). The general connection between AL method and the PM method can be found in Rockafellar (1976).

The PM view in Eq. (8) is useful in theoretically analyzing the M-DAL algorithm. In particular, generalizing the result in Tomioka et al. (2009), we can show that the proposed M-DAL algorithm converges super-linearly, as follows:

Theorem 1. *Let $(\boldsymbol{W}^1, b^1), (\boldsymbol{W}^2, b^1), \ldots$ be the sequence generated by the M-DAL algorithm (Eq. (5)), and let $(\boldsymbol{W}^*, b^*)$ be the unique minimizer of Eq. (3). In addition, let us assume the following:*

1. The gradient of the loss function ∇f_ℓ is Lipshitz continuous with modulus $1/\gamma$, i.e.,

$$\|\nabla f_\ell(\boldsymbol{z}) - \nabla f_\ell(\boldsymbol{z}')\| \le \frac{1}{\gamma}\|\boldsymbol{z} - \boldsymbol{z}'\| \quad (\forall \boldsymbol{z}, \boldsymbol{z}' \in \mathbb{R}^m).$$

2. There is a positive constant σ such that for all $t = 1, 2, \ldots$

$$f(\boldsymbol{W}^{t+1}, b^{t+1}) - f(\boldsymbol{W}^*, b^*) \ge \sigma\|\boldsymbol{W}^{t+1} - \boldsymbol{W}^*\|_{\mathrm{fro}}^2 + \sigma(b^{t+1} - b^*)^2.$$

3. The minimization of φ_t at each step is solved to the following precision:

$$\|\nabla\varphi_t(\boldsymbol{\alpha}^t)\|^2 \le \frac{\gamma}{\eta^t}\left(\|\boldsymbol{W}^{t+1} - \boldsymbol{W}^t\|_{\mathrm{fro}}^2 + (b^{t+1} - b^t)^2\right). \quad (9)$$

Then for every iteration we have the following relationship:

$$D(\boldsymbol{W}^{t+1}, b^{t+1}) \le \frac{1}{\sqrt{1 + 2\sigma\eta^t}} D(\boldsymbol{W}^t, b^t), \quad (10)$$

where $D(\boldsymbol{W}, b) := \sqrt{\|\boldsymbol{W} - \boldsymbol{W}^*\|_{\mathrm{fro}}^2 + (b - b^*)^2}$.

Note that the coefficient in the right-hand side of Eq. (10) is strictly smaller than one and decreases at every iteration because we choose the step-size η^t to be increasing; i.e., we have a super-linear convergence.

2.4. Factorizing $\boldsymbol{W}$ in a Principled Way

In this subsection, we show that we only need to keep a low-rank factorization of $\boldsymbol{W}^t$ to perform M-DAL algorithm with a quasi-Newton method for the inner minimization (7).

Let us start from examining Eq. (5). In order to compute the soft-threshold operation, we need the largest singular values (to the threshold $\lambda\eta^t$) and their corre-

sponding singular-vectors of a matrix $\boldsymbol{W}^t_{\boldsymbol{\alpha}}$ defined as $\boldsymbol{W}^t_{\boldsymbol{\alpha}} := \boldsymbol{W}^t + \eta^t \mathcal{A}^\top(\boldsymbol{\alpha})$. This can be computed efficiently because $\boldsymbol{W}^t$ is low-ranked and $\mathcal{A}^\top(\boldsymbol{\alpha})$ is often structured. For example in the case of low-rank matrix completion, $\mathcal{A}^\top(\boldsymbol{\alpha})$ is a sparse matrix that has entries only where we have observations. Note that instead of forming $\boldsymbol{W}^t_{\boldsymbol{\alpha}}$ explicitly, we can supply function handles for computing left- and right-vector multiplications to $\boldsymbol{W}^t_{\boldsymbol{\alpha}}$ to an SVD solver (e.g., `lansvd` in PROPACK[2]).

Given a set of singular-values of $\boldsymbol{W}^t_{\boldsymbol{\alpha}}$ that are larger than $\lambda\eta^t$, it is straightforward to evaluate Eq. (7), because the squared Frobenius norm of a matrix is the squared sum of its singular-values.

The remaining task is to compute the gradient of Eq. (7) because we use a quasi-Newton method. Let us denote the inner objective function in Eq. (7) by $\varphi_t(\boldsymbol{\alpha})$. The gradient $\nabla_{\boldsymbol{\alpha}}\varphi_t(\boldsymbol{\alpha})$ can be evaluated as follows:

$$\nabla_{\boldsymbol{\alpha}}\varphi_t(\boldsymbol{\alpha}) = -\nabla_{\boldsymbol{\alpha}} f^*_\ell(-\boldsymbol{\alpha}) + \mathbf{1}_m(b^t + \eta^t {\mathbf{1}_m}^\top \boldsymbol{\alpha}) + \mathcal{A}(\mathrm{ST}_{\lambda\eta^t}(\boldsymbol{W}^t + \eta^t \mathcal{A}^\top(\boldsymbol{\alpha}))), \quad (11)$$

which can be obtained by following the line of Tomioka et al. (2009). Clearly we only need the "active" singular-values and singular-vectors of $\boldsymbol{W}^t_{\boldsymbol{\alpha}}$ because of the soft-threshold operation in Eq. (11).

We can also derive the Hessian of Eq. (7) to perform a full Newton method; however this requires computation of both active and inactive singular-values/vectors and is only suitable for medium-scale problems (see Sec. 4.2 and 4.3).

Note that although the above factorization may sound similar to the commonly used non-convex approximation strategy (e.g., Weimer et al. (2008)), the factorization used here is a consequence of the augmented Lagrangian formulation and does not require the number of components to be fixed beforehand.

2.5. Practical Issues

For the SVD of large (but structured) matrices in Sec. 4.1 (see also Sec. 2.4), we use the Lanczos bidiagonalization algorithm with partial reorthogonalization (Simon, 1984) implemented in PROPACK[2]. For the medium-scale problems in Sec. 4.2 and 4.3, we use the MATLAB `svd` routine because the matrix $\boldsymbol{W}^t_{\boldsymbol{\alpha}}$ has no structure. Unfortunately, both SVD solvers need the number k of singular-values that we want to compute to be fixed. Therefore, we repeatedly call the solvers with larger k until we find a singular-value that is smaller than the desired threshold $\lambda\eta^t$ in the soft-threshold operation (see also Cai et al. (2008)).

In order to run the proposed algorithm efficiently, it is important to avoid solving an inner minimization problem that has a larger rank than the final solution. To this end, we use a simple warm-start strategy. More specifically, we start from a sufficiently large regularization constant λ and iteratively solve Eq. (3) with the M-DAL algorithm (Eqs. (5)-(7)) for smaller and smaller λ using the solution obtained from the previous iteration. As a by-product, we obtain the whole regularization path efficiently. A more intelligent continuation (warm-start) strategy is a topic of future study.

2.6. Learning Multiple Matrices

Let us consider a classification/regression problem in which the input is provided as multiple matrices of possibly different sizes. It is natural to consider how one should combine those different sources of information in an intelligent way (see Lanckriet et al. (2004); Micchelli & Pontil (2005); Argyriou et al. (2005)). We would further like to learn low-rank structure inside each group through the trace-norm regularization. The M-DAL algorithm described above can be easily generalized to handle this situation. In particular, we use the regularizer that is defined as sum of the regularizer in Eq. (4) for each matrix to be learned. This is equivalent to learning a larger matrix that is formed by concatenating the matrices to be learned on the diagonal and applying the regularization in Eq. (4) but there is no need to form such a large matrix in our approach.

3. General Spectral Regularization

The ℓ_1-regularization is known to produce overly sparse solution in some cases (Cortes, 2009; Kloft et al., 2010). Therefore it is desirable to have a mechanism to control the level of sparsity in the solution also for matrix-learning problems.

In this section, we consider the following generalized regularization term:

$$\phi_\lambda(\boldsymbol{W}) = \sum_{j=1}^{r} g_\lambda(\sigma_j(\boldsymbol{W})), \quad (12)$$

where $\sigma_j(\boldsymbol{W})$ is the j-th singular-value of $\boldsymbol{W}$ (r is the rank of $\boldsymbol{W}$) and g_λ is a symmetric one-dimensional convex (possibly non-differentiable) function that takes value zero at the origin. λ is the regularization constant and we assume that $\eta g_\lambda = g_{\lambda\eta}$. Note that it is also easy to generalize Eq. (12) to allow for different g_λ for each j.

[2] `http://soi.stanford.edu/~rmunk/PROPACK/`

It can be shown that Eq. (12) is a convex function of $\boldsymbol{W}$. Moreover, we can show that we can naturally generalize the soft-threshold operation in Eq. (2) as follows:

$$\begin{aligned}\mathrm{ST}^g_\lambda(\boldsymbol{W}) &= \operatorname*{argmin}_{\boldsymbol{X}\in\mathbb{R}^{R\times C}} \left(\phi_\lambda(\boldsymbol{X}) + \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{W}\|^2_{\mathrm{fro}}\right) \\ &= \boldsymbol{U}\mathrm{diag}(\mathrm{ST}^g_\lambda(\sigma_1),\ldots,\mathrm{ST}^g_\lambda(\sigma_r))\boldsymbol{V}^\top, \quad (13)\end{aligned}$$

where $\boldsymbol{W} = \boldsymbol{U}\mathrm{diag}(\sigma_1,\ldots,\sigma_r)\boldsymbol{V}^\top$ is the SVD of $\boldsymbol{W}$ and $\mathrm{ST}^g_\lambda(\sigma_j)$ is a one-dimensional "soft-threshold" operation defined as follows:

$$\mathrm{ST}^g_\lambda(\sigma_j) = \operatorname*{argmin}_{x\in\mathbb{R}} \left(g_\lambda(x) + \frac{1}{2}(x-\sigma_j)^2\right). \quad (14)$$

The inner-objective function $\varphi^g_t(\boldsymbol{\alpha})$ and its gradient can be written as follows:

$$\begin{aligned}\varphi^g_t(\boldsymbol{\alpha}) &= f^*_\ell(-\boldsymbol{\alpha}) + \frac{1}{2\eta^t}(b^t + \eta^t \mathbf{1}_m{}^\top\boldsymbol{\alpha})^2 \\ &\quad + \frac{1}{\eta^t}\sum_{j\in\mathcal{J}_+}\left\{g^*_{\lambda^t}\left(\mathrm{ST}^{g^*}_{\lambda^t}(\sigma_j(\boldsymbol{W}^t_{\boldsymbol{\alpha}}))\right)\right. \\ &\quad \left. + \frac{1}{2}\left(\mathrm{ST}^g_{\lambda^t}(\sigma_j(\boldsymbol{W}^t_{\boldsymbol{\alpha}}))\right)^2\right\}, \\ \nabla_{\boldsymbol{\alpha}}\varphi^g_t(\boldsymbol{\alpha}) &= -\nabla_{\boldsymbol{\alpha}} f^*_\ell(-\boldsymbol{\alpha}) + \mathbf{1}_m(b^t + \eta^t\mathbf{1}_m^\top\boldsymbol{\alpha}) \\ &\quad + \mathcal{A}(\mathrm{ST}^g_{\lambda^t}(\boldsymbol{W}^t_{\boldsymbol{\alpha}})),\end{aligned}$$

where $\boldsymbol{W}^t_{\boldsymbol{\alpha}} = \boldsymbol{W}^t + \eta^t\mathcal{A}^\top(\boldsymbol{\alpha})$ and $\lambda^t = \lambda\eta^t$ are defined for brevity; g^*_λ is the convex conjugate of g_λ and $\mathrm{ST}^{g^*}_\lambda$ is the soft-thresholding with respect to g^*_λ defined as in Eq. (14). In addition, we define the set $\mathcal{J}_+$ as the set of indices of the singular-values of the matrix $\boldsymbol{W}^t_{\boldsymbol{\alpha}}$ that are large enough so that $\mathrm{ST}^g_\lambda(\sigma_j(\boldsymbol{W}^t_{\boldsymbol{\alpha}})) > 0$.

Note that similarly to the observation in Sec. 2.4, we only need to compute and store the "active" singular-values (σ_j: $j\in\mathcal{J}_+$) and their singular-vectors if we only use the first-order information of the inner objective function $\varphi^g_t(\boldsymbol{\alpha})$.

4. Experiments

4.1. Low-Rank Matrix Completion

We apply the proposed M-DAL algorithm to the problem of recovering a partially observed matrix under the trace-norm regularization, which was studied in Cai et al. (2008). In this problem, $\boldsymbol{W}\in\mathbb{R}^{R\times C}$ is the matrix to be recovered; no bias term b is used; the observation operator $\mathcal{A}$ picks (a small number of) elements of $\boldsymbol{W}$ and vectorizes them into an m-dimensional vector; the adjoint operator $\mathcal{A}^\top$, on the other hand, maps an m-dimensional vector $\boldsymbol{\alpha}$ into a $R\times C$ matrix with the only the elements corresponding to the observations filled with the elements of $\boldsymbol{\alpha}$. Moreover, we use the quadratic loss function: $f_\ell(\boldsymbol{z}) = \|\boldsymbol{z}-\boldsymbol{y}\|^2/2$, where $\boldsymbol{y}\in\mathbb{R}^m$ is the vector of observations; for the quadratic loss we can easily compute the modulus of Lipshitz continuity $\gamma = 1$; as is shown in Tomioka & Sugiyama (2009), the convex conjugate of f_ℓ is f_ℓ itself (ignoring constants).

We randomly generated a rank 10 matrix[3] of size 10,000×10,000 and randomly sampled its m =1,200,000 elements (1.2%). We ran the proposed M-DAL algorithm with a warm start over the sequence of regularization constants $\lambda = (1000, 700, 500, 300, 200, 150, 100)$, which was empirically found to work the best. We used the initial step-size $\eta^1 = 10$ and increased η^t by the factor 2 at every iteration. The simulation was repeated 10 times on a computer with two dual core 3.3GHz Xeon processors and 8GB of RAM.

The result is summarized in Table 1. Shown in the table are, the CPU time, the number of outer iterations (Eq. (5)), the number of inner (L-BFGS) steps, and the number of SVDs spent at each optimization, the rank and the subspace-root-mean-square error (S-RMSE) obtained after each optimization, which is defined as follows:

$$\begin{aligned}\text{S-RMSE} = \Big(&\frac{1}{rr^*}\textstyle\sum_{i,j}(\boldsymbol{U}^\top\boldsymbol{U}^* - \boldsymbol{I}_{r,r^*})^2_{i,j} \\ &+ \frac{1}{rr^*}\textstyle\sum_{i,j}(\boldsymbol{V}^\top\boldsymbol{V}^* - \boldsymbol{I}_{r,r^*})^2_{i,j}\Big)^{1/2},\end{aligned}$$

where $(\boldsymbol{U},\boldsymbol{V})$ and $(\boldsymbol{U}^*,\boldsymbol{V}^*)$ are the left- and right-singular-vectors of the estimated matrix $\boldsymbol{W}$ (rank r) and the true matrix $\boldsymbol{W}^*$ (rank r^*), respectively; $\boldsymbol{I}_{r,r^*}$ is the $r\times r^*$ matrix that has one for (i,i)-elements ($i = 1,\ldots,\min(r,r^*)$) and zero in all other elements.

Cai et al. (2008) studied a noise-less ($\lambda\to 0$) low-rank matrix completion problem and proposed the singular-value thresholding (SVT) algorithm. Run on an almost identical problem as our setting, SVT spent 123 iterations, which took 281 seconds. We consider our result as comparable to their result because we obtain a regularization path in a slightly longer time with a smaller number of iterations, and we solve the problem (3) directly without introducing a small Frobenius-norm regularization as in Cai et al. (2008).

4.2. Simulated Matrix Classification Problem

We simulate a classification problem over matrices similar to the setting considered in Tomioka & Aihara

[3] This corresponds to the MATLAB command `W=randn(n,k)*diag(k:-1:1)*randn(k,n)` with n =10,000 and k =10 but only computed implicitly.

Table 1. Statistics of M-DAL algorithm applied to a 10,000×10,000 low-rank matrix completion problem. The rank of the true matrix $\boldsymbol{W}^*$ is 10. The cumulative CPU time, the cumulative number of outer iterations, the cumulative number of inner iterations, the cumulative number of SVDs spent to obtain the solution for each regularization constant λ, as well as the rank and S-RMSE of the estimated matrix $\boldsymbol{W}$ are shown. Note that the cumulative CPU time and the cumulative number of iterations are shown, because we use a warm start strategy. The numbers shown are the mean and the standard-deviation (inside the parenthesis) of 10 random runs.

λ	time (s)	#outer	#inner	#SVDs	rank	S-RMSE
1000	33.1 (±2.0)	5 (±0)	8 (±0)	32 (±0)	2.8 (±0.4)	0.0158 (±0.0024)
700	77.1 (±5.6)	11 (±0)	18 (±0)	71 (±0)	5 (±0)	0.0133 (±0.0008)
500	124 (±7.2)	17 (±0)	28 (±0)	110 (±0)	6.4 (±0.5)	0.0113 (±0.0015)
300	174 (±8.0)	23 (±0)	38.4 (±0.84)	150 (±3.0)	8 (±0)	0.00852 (±0.00039)
200	220 (±9.9)	29 (±0)	48.4 (±0.84)	189 (±3.0)	9 (±0)	0.00767 (±0.00031)
150	257 (±9.9)	35 (±0)	58.4 (±0.84)	230 (±3.0)	9 (±0)	0.00498 (±0.00026)
100	319 (±11)	41 (±0)	70 (±0.82)	276 (±2.7)	10 (±0)	0.00743 (±0.00013)

(2007) in order to compare the proposed M-DAL algorithm against three state-of-the-art methods to solve this kind of problem. These methods are the interior-point (IP) method (Tomioka & Aihara, 2007), the projected gradient method (Tomioka & Sugiyama, 2008), and the accelerated gradient (AG) method (Ji & Ye, 2009).

The problem setting is as follows. We define a random low-rank classifier matrix $\boldsymbol{W}^*$ by taking the largest and smallest k-eigenvalues and their corresponding eigenvectors of a randomly drawn $n \times n$ symmetric matrix[4]. The input matrices $\boldsymbol{X}_i \in \mathbb{R}^{n \times n}$ $(i = 1, \ldots, m)$ are sampled independently from the standard Wishart distribution with n degrees of freedom. The output label y_i for the i-th sample is generated by taking the sign of $\langle \boldsymbol{W}^*, \boldsymbol{X}_i \rangle$, where $\boldsymbol{W}^*$ is the true classifier matrix. Since this is a classification problem, we use the logistic loss function, which is defined as follows:

$$f_\ell(\boldsymbol{z}) = \sum_{i=1}^{m} \log(1 + \exp(-y_i z_i)). \tag{15}$$

The convex conjugate of the logistic loss is known to be the following negative entropy function:

$$f_\ell^*(-\boldsymbol{\alpha}) = \sum_{i=1}^{m} (\alpha_i y_i \log(\alpha_i y_i) + (1 - \alpha_i y_i) \log(1 - \alpha_i y_i)),$$

where $0 \leq \alpha_i y_i \leq 1$. By a simple calculation, we have the modulus of Lipshitz continuity $\gamma = 4$.

For this experiment (and in the next experiment) we also use the Hessian of $\varphi_t(\boldsymbol{\alpha})$ because it results in faster convergence and computing the full SVD is cheap for small matrices.

[4]We used the following MATLAB command: `W=randn(n,n); W=(W+W')/2;`.

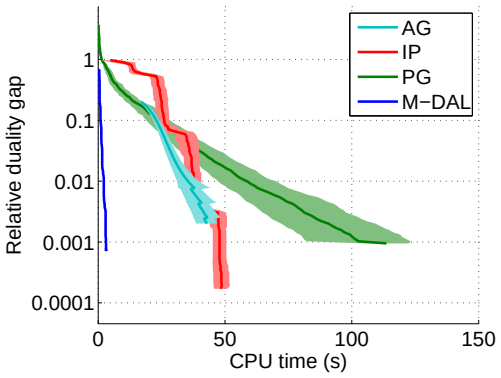

Figure 1. Comparison of the computational efficiency of accelerated gradient method (AG) (Ji & Ye, 2009), interior-point method (IP) (Tomioka & Aihara, 2007), projected gradient method (Tomioka & Sugiyama, 2008), and the proposed M-DAL algorithm. The three algorithms were run on a simulated classification problem over symmetric 64×64 matrices with the trace-norm regularization. Number of samples $m = 1,000$. The regularization constant $\lambda = 800$. The shaded area shows the standard deviation over 10 runs.

Figure 1 shows the decrease in the relative duality gap (RDG, defined as (primal objective − dual objective)/(primal objective); see Tomioka & Sugiyama (2009)) of four algorithms against the CPU time spent for the number of observations $m = 1000$, the size of matrices $n = 64$, the true rank $2k = 16$, and the regularization constant $\lambda = 800$, which was chosen to roughly recover the original rank 16. We used the initial step-size $\eta^1 = 10^{-5}$ and increased η^t by the factor 2 at every iteration. Clearly the proposed M-DAL approach is roughly 10 times faster than the previously proposed methods.

Table 2. Result on the BCI data-set. Three algorithms are compared, namely, the accelerated gradient (AG) method (Ji & Ye, 2009), the projected gradient (PG) method (Tomioka & Sugiyama, 2008), and M-DAL. For each method, shown are the cumulative number of iteration, the cumulative CPU time (in seconds), the relative duality gap (RDG), and the test accuracy (ACC) in %. All three algorithms were applied to the classification problem over multiple matrices (Tomioka & Müller, 2010) with increasingly smaller regularization constants shown at the left-most column of the table.

	AG (Ji & Ye, 2009)				PG (Tomioka & Sugiyama, 2008)				**M-DAL (proposed)**			
λ	#iter.	time	RDG	ACC	#iter.	time	RDG	ACC	#iter.	time	RDG	ACC
6.16	32	1.77	0.0047	76	414	4.59	0.00095	75	5	0.6	0.00029	75
0.886	1263	15.8	0.0229	85	97304	1001	0.00093	84	39	10.2	0.00064	84
0.127	5850	65.4	0.0395	83	458954	4644	0.00355	83	89	43.1	0.00059	83
0.0183	18766	205	0.084	84	858954	8511	0.0108	81	155	118	0.00016	81
0.00264	49164	540	0.201	84	1.3×10^6	12328	0.0243	80	234	217	0.00028	80

4.3. Brain-Computer Interface Data-set

In this subsection, we demonstrate the ability of the proposed M-DAL framework to handle the problem of learning multiple matrices simultaneously. The data-set is taken from a real brain-computer interface (BCI) experiment, where the task is to predict whether the upcoming voluntary finger movement is either right or left hand from the electroencephalography (EEG) measurements. The data-set is made publicly available through the BCI competition 2003 (data-set IV) (Blankertz et al., 2004). More specifically, the data-set consists of short segments of 28 channel multivariate signal of length 50 (500 ms long at 100 Hz sampling). The training set consists of 316 input segments (159 left and 157 right) and we tested the classifier on a separate test-set consisting of 100 test segments.

Following the preprocessing used in Tomioka & Müller (2010), we compute three matrices from each segment. The first matrix is 28×50 and is obtained directly from the original signal by low-path filtering at 20Hz. The second matrix is 28×28 and is derived by computing the covariance between the channels in the frequency band 7-15Hz (known as the α-band). Finally, the third matrix is 28×28 and is computed similarly to the second matrix in the frequency band 15-30Hz (known as the β-band). We chose 20 log-linearly separated values of λ from 10 to 0.001 and used a warm start for all methods. Due to space limitation, only some values of λ are shown in Tab. 2.

The accelerated gradient (AG) method was stopped when the relative change of the function value fell below 10^{-5} (Ji & Ye, 2009). The projected gradient (PG) and the M-DAL algorithms were stopped when the relative duality gap fell below 10^{-3}. Note that PG method did not achieve the desired precision and had to be stopped after 100,000 iterations for $\lambda \leq 0.207$. For the M-DAL algorithm we used the step-sizes $\eta_t = 1, 2, 4, 8, 16, \ldots$ for each λ.

In Tab. 2, the proposed M-DAL algorithm is compared with the AG method (Ji & Ye, 2009) and the PG method (Tomioka & Sugiyama, 2008). The M-DAL algorithm is clearly two orders of magnitude faster than the PG method. Remarkably the speedup obtained by M-DAL against PG method seems to be greater than the simulated problem in the last subsection. This extra speed-up can be explained by the poorly conditioned nature of real EEG signals, which DAL algorithm has shown to tolerate well (Tomioka & Sugiyama, 2009). In addition, M-DAL also shows higher precision than the PG method (smaller relative duality gap). The AG method is slower than M-DAL but the obtained solution is not very precise, which can be seen from the higher RDG. In addition, it does not show clear overfitting for small λ, because the objective is not minimized precisely.

5. Conclusion

In this paper, we have proposed an efficient algorithm M-DAL for learning low-rank matrices with the trace-norm regularization. We have shown that the proposed algorithm converges super-linearly and can be readily extended to general spectral regularization. Experimental results on both simulated and real data-sets have shown the efficiency and high precision of the proposed algorithm in a matrix completion problem and classification problems over matrices. Future work includes better continuation strategy and using the M-DAL algorithm within a more general estimation framework (e.g.,Wipf & Nagarajan (2008)).

Acknowledgments This work was partially supported by MEXT KAKENHI 22700138, 22700289, 80545583, and the FIRST program.

References

Abernethy, J., Bach, F., Evgeniou, T., and Vert, J. P. A new approach to collaborative filtering: Operator estimation with spectral regularization. *The Journal of Machine Learning Research*, 10:803–826, 2009.

Amit, Y., Fink, M., Srebro, N., and Ullman, S. Uncovering Shared Structures in Multiclass Classification. In *Proc. ICML '07*, pp. 17–24, New York, NY, USA, 2007. ACM Press.

Argyriou, A., Micchelli, C.A., and Pontil, M. Learning convex combinations of continuously parameterized basic kernels. In Auer, P. and Meir, R. (eds.), *Proc. COLT2005*. Springer, Berlin, Heidelberg, 2005.

Argyriou, A., Evgeniou, T., and Pontil, M. Multi-task feature learning. In Schölkopf, B., Platt, J., and Hoffman, T. (eds.), *Advances in NIPS 19*, pp. 41–48. MIT Press, Cambridge, MA, 2007.

Argyriou, A., Micchelli, C. A., Pontil, M., and Ying, Y. A spectral regularization framework for multi-task structure learning. In Platt, J.C., Koller, D., Singer, Y., and Roweis, S. (eds.), *Advances in NIPS 20*, pp. 25–32. MIT Press, Cambridge, MA, 2008.

Bertsekas, D. P. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

Blankertz, B., Müller, K.-R., Curio, G., Vaughan, T. M., Schalk, G., Wolpaw, J. R., Schlögl, A., Neuper, C., Pfurtscheller, G., Hinterberger, T., Schröder, M., and Birbaumer, N. The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. *IEEE Trans. Biomed. Eng.*, 51(6):1044–1051, 2004.

Cai, J.-F., Candes, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. arXiv:0810.3286, 2008.

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

Chen, S., Donoho, D., and Saunders, M. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1): 33–61, 1998.

Cortes, C. Can learning kernels help performance? Invited talk at International Conference on Machine Learning (ICML 2009). Montréal, Canada, 2009.

Fazel, M., Hindi, H., and Boyd, S. P. A Rank Minimization Heuristic with Application to Minimum Order System Approximation. In *Proc. of the American Control Conference*, 2001.

Figueiredo, M. A. T., Bioucas-Dias, J. M., and Nowak, R. D. Majorization-Minimization Algorithm for Wavelet-Based Image Restoration. *IEEE Trans. Image Process.*, 16(12), 2007.

Hestenes, M. R. Multiplier and gradient methods. *J. Optim. Theory Appl.*, 4:303–320, 1969.

Ji, S. and Ye, J. An accelerated gradient method for trace norm minimization. In *Proc. ICML '09*, pp. 457–464, New York, NY, 2009. ACM.

Kloft, M., Brefeld, U., Sonnenburg, S., Laskov, P., Müller, K.-R., and Zien, A. Efficient and accurate lp-norm multiple kernel learning. In *Advances in NIPS 22*. 2010.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., El Ghaoui, L., and Jordan, M. I. Learning the Kernel Matrix with Semidefinite Programming. *J. Machine Learning Research*, 5:27–72, 2004.

Micchelli, C.A. and Pontil, M. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.

Nocedal, J. and Wright, S. *Numerical Optimization*. Springer, 2nd edition, 2006.

Powell, M. J. D. A method for nonlinear constraints in minimization problems. In Fletcher, R. (ed.), *Optimization*, pp. 283–298. Academic Press, London, New York, 1969.

Rockafellar, R. T. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. of Oper. Res.*, 1:97–116, 1976.

Simon, H. D. The lanczos algorithm with partial reorthogonalization. *Math. Comput.*, 42(165):115–142, 1984.

Srebro, N., Rennie, J. D. M., and Jaakkola, T. S. Maximum-Margin Matrix Factorization. In *Advances in NIPS17*, pp. 1329–1336. MIT Press, Cambridge, MA, 2005.

Tibshirani, R. Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc. B*, 58(1):267–288, 1996.

Tomioka, R. and Aihara, K. Classifying Matrices with a Spectral Regularization. In *Proc. of the 24th international conference on Machine learning*, pp. 895–902. ACM Press, 2007.

Tomioka, R. and Müller, K.-R. A regularized discriminative framework for EEG analysis with application to brain–computer interface. *Neuroimage*, 49(1):415–432, 2010.

Tomioka, R. and Sugiyama, M. Sparse learning with duality gap guarantee. In *NIPS workshop OPT 2008 Optimization for Machine Learning*, 2008.

Tomioka, R. and Sugiyama, M. Dual Augmented Lagrangian Method for Efficient Sparse Reconstruction. *IEEE Signal Processing Letters*, 16(12):1067–1070, 2009.

Tomioka, R., Suzuki, T., and Sugiyama, M. Super-Linear Convergence of Dual Augmented Lagrangian Algorithm for Sparse Learning, 2009. arXiv:0911.4046 [stat.ML].

Weimer, M., Karatzoglou, A., Le, Q., and Smola, A. Cofi rank - maximum margin matrix factorization for collaborative ranking. In Platt, J.C., Koller, D., Singer, Y., and Roweis, S. (eds.), *Advances in NIPS 20*, pp. 1593–1600. MIT Press, Cambridge, MA, 2008.

Wipf, D. and Nagarajan, S. A new view of automatic relevance determination. In *Advances in NIPS 20*, pp. 1625–1632. 2008.