Trace Norm Regularization: Reformulations, Algorithms, and Multi-task Learning

Ting Kei Pong[*] Paul Tseng[†] Shuiwang Ji[‡] Jieping Ye[§]

March 21, 2010

Abstract

We consider a recently proposed optimization formulation of multi-task learning based on trace norm regularized least squares. While this problem may be formulated as a semidefinite program (SDP), its size is beyond general SDP solvers. Previous solution approaches apply proximal gradient methods to solve the primal problem. We derive new primal and dual reformulations of this problem, including a reduced dual formulation that involves minimizing a convex quadratic function over an operator-norm ball in matrix space. This reduced dual problem may be solved by gradient-projection methods, with each projection involving a singular value decomposition. The dual approach is compared with existing approaches and its practical effectiveness is illustrated on simulations and an application to gene expression pattern analysis.

Key words. Multi-task learning, gene expression pattern analysis, trace norm regularization, convex optimization, duality, semidefinite programming, proximal gradient method.

1 Introduction

In various applications we seek a low rank $W \in \mathbb{R}^{n\times m}$ to minimize the discrepancy between AW and B, where $A \in \mathbb{R}^{p\times n}$ and $B \in \mathbb{R}^{p\times m}$. This problem is NP-hard in general, and various approaches have been proposed to solve this problem approximately. One promising heuristic entails regularizing the objective by the trace norm (also called nuclear norm) $\|W\|_\star$, defined as the sum of the singular values of W. This results in a convex optimization problem that can be reformulated as a semidefinite program (SDP) and be solved by various methods. This approach has been applied to minimum-order system realization [19], low-rank matrix completion [14, 15], low-dimensional Euclidean embedding [20], dimension reduction in multivariate linear regression [33, 54], multi-class classification [2], and multi-task learning [1, 5, 41]. We consider the regularized least squares formulation

$$\upsilon \ := \ \min_W \frac{1}{2}\|AW - B\|_F^2 + \mu\|W\|_*, \tag{1}$$

where $\mu > 0$ and $\|\cdot\|_F$ denotes the Frobenius-norm; see [1, 5, 33, 41, 47]. By dividing A and B with $\sqrt{\mu}$, we can without loss of generality assume that $\mu = 1$. However, this is computationally inefficient

[*] Department of Mathematics, University of Washington, Seattle, WA 98195, USA (tkpong@math.washington.edu)

[†] Department of Mathematics, University of Washington, Seattle, WA 98195, USA (tseng@math.washington.edu)

[‡] Department of Computer Science and Engineering, Center for Evolutionary Functional Genomics, The Biodesign Institute, Arizona State University, Tempe, AZ 85287, USA (shuiwang.ji@asu.edu)

[§] Department of Computer Science and Engineering, Center for Evolutionary Functional Genomics, The Biodesign Institute, Arizona State University, Tempe, AZ 85287, USA (jieping.ye@asu.edu)

when solving (1) for multiple values of μ. We may view μ as the Lagrange multiplier associated with the constrained formulation

$$\min_W \frac{1}{2}\|AW - B\|_F^2 \quad \text{subject to} \quad \|W\|_* \le \omega, \tag{2}$$

for some $\omega \ge 0$ [54], or view $1/\mu$ as the Lagrange multiplier associated with the alternative constrained formulation

$$\min_W \|W\|_* \quad \text{subject to} \quad \frac{1}{2}\|AW - B\|_F^2 \le \rho, \tag{3}$$

for some $\rho \ge 0$.

Our motivation stems from multi-task learning [16], which has recently received attention in broad areas such as machine learning, data mining, computer vision, and bioinformatics [3–5, 22, 53]. Multi-task learning aims to improve the generalization performance of classifiers by learning from multiple related tasks. This can be achieved by learning the tasks simultaneously, and meanwhile exploiting their intrinsic relatedness, based on which the informative domain knowledge is allowed to be shared across the tasks, thus facilitating "cooperative" task learning. This approach is particularly effective when only limited training data for each task is available.

Multi-task learning has been investigated by many researchers using different approaches such as sharing hidden units of neural networks among similar tasks [9, 16], modeling task relatedness using a common prior distribution in hierarchical Bayesian models [8], extending kernel methods and regularization networks to multi-task learning [18], and multi-task learning with clustered tasks [6, 24]. Recently, there is growing interest in studying multi-task learning in the context of feature learning and selection [2, 4, 5, 40]. Specifically, [4] proposes the alternating structure optimization (ASO) algorithm to learn the predictive structure from multiple tasks. In ASO, a separate linear classifier is trained for each of the tasks and dimension reduction is applied on the predictor (classifier) space, finding low-dimensional structures with the highest predictive power. This approach has been applied successfully in several applications [3, 42]. However, the optimization formulation is non-convex and the alternating optimization procedure used is not guaranteed to find a global optimum [4]. Recently, trace norm regularization has been proposed for multi-task learning [1, 5, 41], resulting in the convex optimization problem (1). We explain its motivation in more detail below.

Assume that we are given m supervised (binary-class) learning tasks. Each of the learning tasks is associated with a predictor f_ℓ and training data $\{(x_1, y_1^\ell), \cdots, (x_p, y_p^\ell)\} \subset \mathbb{R}^n \times \{-1, 1\}$ $(\ell = 1, \ldots, m)$. We focus on linear predictors $f_\ell(x) = w_\ell^\top x$, where w_ℓ is the weight vector for the ℓth task. The convex multi-task learning formulation based on the trace norm regularization can be formulated as the following optimization problem:

$$\min_{\{w_\ell\}} \sum_{\ell=1}^{m} \left(\sum_{i=1}^{p} L(w_\ell^\top x_i, y_i^\ell) \right) + \mu \|W\|_\star, \tag{4}$$

where L is the loss function, and $\|W\|_\star$ is the trace norm of the matrix W, given by the summation of the singular values of W. For the least squares loss $L(s, t) = \frac{1}{2}(s - t)^2$, (4) reduces to (1) with $A = [x_1, \cdots, x_p]^\top \in \mathbb{R}^{p \times n}$, and the (i, j)-th entry of $B \in \mathbb{R}^{p \times m}$ is y_i^j. Trace norm regularization has the effect of inducing W to have low rank.

A practical challenge in using trace norm regularization is to develop efficient methods to solve the convex, but non-smooth, optimization problem (1). It is well known that a trace norm minimization problem can be formulated as an SDP [19, 46]. However, such formulation is computationally expensive for existing SDP solvers. To overcome the nonsmoothness, nonconvex smooth reformulations of (1) have been proposed that are solved by conjugate gradient or alternating minimization method [43, 51, 52]. However, the nonconvex reformulation introduces spurious stationary points and convergence to a global minimum is not guaranteed. Recently, proximal gradient methods have been applied to solve (1). These methods have good theoretical convergence properties and their practical performances seem promising. In [33], A is assumed to have full column rank and a variant of Nesterov's smooth method [50, Algorithm 3] is applied to solve a certain saddle point reformulation of (1). Numerical results on random instances of A, B, with m up to 60 and $n = 2m$, $p = 10m$, show the method solves (1) much faster than the general SDP solver SDPT3. In [34], a proximal gradient method is used to solve (1) (see (26)), with μ gradually decreased to accelerate convergence. Each iteration of this method involves a singular value decomposition (SVD) of an $n \times m$ matrix. It solved randomly generated matrix completion problems of size up to 1000×1000. In [27, 47], an accelerated proximal gradient method is used. In [47], additional techniques to accelerate convergence and reduce SVD computation are developed, and matrix completion problems of size up to $10^5 \times 10^5$ are solved. For the constrained version of the problem, corresponding to (3) with $\rho = 0$, primal-dual interior-point method [31] and smoothed dual gradient method [14] have been proposed. The successive regularization approach in [34] entails solving a sequence of problems of the form (1).

In this paper, we derive alternative primal and dual reformulations of (1) with varying advantages for numerical computation. In particular, we show that (1) is reducible to the case where A has full column rank, i.e., $r = n$, where

$$r := \text{rank}(A);$$

see Section 2. We then show that this reduced problem has a dual that involves minimizing a quadratic function over the unit-operator-norm ball in $\mathbb{R}^{r\times m}$; see Section 3. This reduced dual problem may be solved by a conditional gradient method and (accelerated) gradient-projection methods, with each projection involving an SVD of an $r \times m$ matrix. The primal and dual gradient methods complement each other in the sense that the iteration complexity of the former is proportional to the largest eigenvalue of $A^\mathsf{T}A$ while that of the latter is inversely proportional to the smallest nonzero eigenvalue of $A^\mathsf{T}A$; see Section 4. For multi-task learning, m is small relative to r, so that computing an SVD of an $r \times m$ matrix is relatively inexpensive. Alternative primal formulations that may be advantageous when $r < m$ are also derived; see Section 5. Numerical tests on randomly generated data suggest that the dual problem is often easier to solve than the primal problem, particularly by gradient-projection methods; see Section 6. In Section 7 we report the results of experiments conducted on the automatic gene expression pattern image annotation task [26, 48]. The results demonstrate the efficiency and effectiveness of the proposed multi-task learning formulation for m up to 60 and $n = 3000$ and p up to 3000.

In our notation, for any $W, Z \in \mathbb{R}^{n\times m}$, $\langle W, Z\rangle = \text{tr}(W^\mathsf{T}Z)$, so that $\|Z\|_F = \sqrt{\langle Z, Z\rangle}$. For any $W \in \mathbb{R}^{n\times m}$, $\sigma_{\max}(W)$ denotes its largest singular value. For any symmetric $C, D \in \mathbb{R}^{n\times n}$, $\lambda_{\min}(C)$ and $\lambda_{\max}(C)$ denote, respectively, the smallest and the largest eigenvalues of C, and $C \succeq D$ (respectively, $C \succ D$) means $C - D$ is positive semidefinite (respectively, positive definite) [23, Section 7.7].

2 Problem reduction when A lacks full column rank

Suppose A lacks full column rank, i.e., $r < n$. (Recall that $r = \text{rank}(A)$.) We show below that (1) is reducible to one for which $r = n$.

We decompose

$$A = R \begin{bmatrix} \widetilde{A} & 0 \end{bmatrix} S^\mathsf{T},$$

where $R \in \mathbb{R}^{p\times p}$ and $S \in \mathbb{R}^{n\times n}$ are orthogonal, and $\widetilde{A} \in \mathbb{R}^{p\times r}$. In a QR decomposition of A, $\widetilde{A}$ is lower triangular with positive diagonals and R is a permutation matrix. In a singular value decomposition

(SVD) of A, $\widetilde{A}$ is diagonal with positive diagonals. In general, QR decomposition is much faster to compute than SVD. The following result shows that A and B in (1) are reducible to $\widetilde{A}$ and $R^{\mathsf{T}}B$.

Proposition 1.

$$v \;=\; \min_{\widetilde{W}} \frac{1}{2}\|\widetilde{A}\widetilde{W} - R^{\mathsf{T}}B\|_F^2 + \mu\|\widetilde{W}\|_\star,$$

where $\widetilde{W} \in \mathbb{R}^{r\times m}$*.*

Proof. For any $W \in \mathbb{R}^{n\times m}$, let

$$\begin{bmatrix}\widetilde{W}\\ \hat{W}\end{bmatrix} = S^{\mathsf{T}}W,$$

where $\widetilde{W} \in \mathbb{R}^{r\times m}$ and $\hat{W} \in \mathbb{R}^{(n-r)\times m}$. Then

$$\|W\|_\star = \|S^{\mathsf{T}}W\|_\star = \left\|\begin{bmatrix}\widetilde{W}\\ \hat{W}\end{bmatrix}\right\|_\star \geq \|\widetilde{W}\|_\star,$$

where the inequality uses $\begin{bmatrix}\widetilde{W}^{\mathsf{T}} & \hat{W}^{\mathsf{T}}\end{bmatrix}\begin{bmatrix}\widetilde{W}\\ \hat{W}\end{bmatrix} = \widetilde{W}^{\mathsf{T}}\widetilde{W} + \hat{W}^{\mathsf{T}}\hat{W} \succeq \widetilde{W}^{\mathsf{T}}\widetilde{W}$ and the order-preserving property of the eigenvalue map [23, Corollary 7.7.4(c)]. Hence

$$\begin{aligned} p(W) \;&:=\; \frac{1}{2}\|AW - B\|_F^2 + \mu\|W\|_\star \qquad (5)\\ &=\; \frac{1}{2}\left\|R\begin{bmatrix}\widetilde{A} & 0\end{bmatrix}S^{\mathsf{T}}W - B\right\|_F^2 + \mu\|W\|_\star\\ &=\; \frac{1}{2}\left\|\begin{bmatrix}\widetilde{A} & 0\end{bmatrix}\begin{bmatrix}\widetilde{W}\\ \hat{W}\end{bmatrix} - R^{\mathsf{T}}B\right\|_F^2 + \mu\|S^{\mathsf{T}}W\|_\star\\ &\geq\; \frac{1}{2}\left\|\widetilde{A}\widetilde{W} - R^{\mathsf{T}}B\right\|_F^2 + \mu\|\widetilde{W}\|_\star\\ &=:\; \tilde{p}(\widetilde{W}). \end{aligned}$$

It follows that $v = \min_W p(W) \geq \min_{\widetilde{W}} \tilde{p}(\widetilde{W})$. On the other hand, for each $\widetilde{W} \in \mathbb{R}^{r\times m}$, letting

$$W = S\begin{bmatrix}\widetilde{W}\\ 0\end{bmatrix}$$

yields $W^{\mathsf{T}}W = \widetilde{W}^{\mathsf{T}}\widetilde{W}$, so that $\|W\|_\star = \|\widetilde{W}\|_\star$. Hence

$$\begin{aligned} \tilde{p}(\widetilde{W}) &= \frac{1}{2}\|\widetilde{A}\widetilde{W} - R^{\mathsf{T}}B\|_F^2 + \mu\|\widetilde{W}\|_\star\\ &= \frac{1}{2}\left\|R\begin{bmatrix}\widetilde{A} & 0\end{bmatrix}\begin{bmatrix}\widetilde{W}\\ 0\end{bmatrix} - B\right\|_F^2 + \mu\|W\|_\star\\ &= \frac{1}{2}\left\|R\begin{bmatrix}\widetilde{A} & 0\end{bmatrix}S^{\mathsf{T}}S\begin{bmatrix}\widetilde{W}\\ 0\end{bmatrix} - B\right\|_F^2 + \mu\|W\|_\star\\ &= \frac{1}{2}\|AW - B\|_F^2 + \mu\|W\|_\star\\ &= p(W). \end{aligned}$$

Hence $\min_{\widetilde{W}} p(\widetilde{W}) \geq \min_W p(W) = v$ and the proposition is proved. □

3 A reduced dual problem

By Proposition 1, (1) is reducible to the case of $r = n$, which we assume throughout this section. Using this, we will derive a dual of (1), defined on the same space of $\mathbb{R}^{r\times m}$, that has additional desirable properties and in some sense complements the primal problem. In our numerical experience, this dual problem is often easier to solve by gradient methods; see Sections 6 and 7.

First, note that the trace norm is the dual norm of the operator norm, namely

$$\|W\|_\star = \max_{\sigma_{\max}(\Lambda)\le 1} -\langle \Lambda, W\rangle = \max_{\Lambda^\mathsf{T}\Lambda\preceq I} -\langle \Lambda, W\rangle,$$

where $\Lambda \in \mathbb{R}^{r\times m}$. Thus (1) can be rewritten as the minimax problem:

$$\upsilon \;=\; \min_{W} \max_{\Lambda^\mathsf{T}\Lambda\preceq I} \frac{1}{2}\|AW - B\|_F^2 - \mu\langle \Lambda, W\rangle.$$

Switching "min" and "max" yields its dual:

$$\max_{\Lambda^\mathsf{T}\Lambda\preceq I} \min_{W} \frac{1}{2}\|AW - B\|_F^2 - \mu\langle \Lambda, W\rangle. \tag{6}$$

Since the dual feasible set $\{\Lambda \mid \Lambda^\mathsf{T}\Lambda \preceq I\}$ is convex and compact, there is no duality gap, i.e., the optimal value of (6) equals υ; see [45, Corollary 37.3.2]. The minimization in W is attained at $A^\mathsf{T}(AW-B)-\mu\Lambda = 0$. Solving for W yields $W = \mu C\Lambda + E$, where we let

$$M := A^\mathsf{T}A, \qquad C := M^{-1}, \qquad E := CA^\mathsf{T}B. \tag{7}$$

Plugging this into (6) yields

$$\upsilon \quad = \quad \max_{\Lambda^\mathsf{T}\Lambda\preceq I} -\frac{\mu^2}{2}\langle \Lambda, C\Lambda\rangle - \mu\langle E, \Lambda\rangle - \frac{1}{2}\langle E, A^\mathsf{T}B\rangle + \frac{1}{2}\|B\|_F^2.$$

Negating the objective and scaling Λ by μ, this dual problem reduces to

$$\min_{\Lambda^\mathsf{T}\Lambda\preceq \mu^2 I} d(\Lambda) := \frac{1}{2}\langle \Lambda, C\Lambda\rangle + \langle E, \Lambda\rangle + \frac{1}{2}\langle E, A^\mathsf{T}B\rangle - \frac{1}{2}\|B\|_F^2. \tag{8}$$

(We scale Λ to make the dual objective independent of μ.) This is a convex quadratic minimization over the μ-operator-norm ball (since $\Lambda^\mathsf{T}\Lambda \preceq \mu^2 I$ if and only if $\sigma_{\max}(\Lambda) \le \mu$). Moreover, given any dual solution Λ^*, a primal solution W^* can be recovered by

$$W^* = C\Lambda^* + E. \tag{9}$$

On the other hand, given any primal solution W^*, a dual solution Λ^* can be recovered by

$$\Lambda^* = M(W^* - E). \tag{10}$$

As we shall see in Section 4, this provides a practical termination criterion for methods solving either the primal problem (1) or the dual problem (8).

The following result shows that, when $r \ge m$, minimizing a linear function over the constraint set of (8) and projecting onto it requires only an SVD of an $r \times m$ matrix. This will be key to the efficient implementation of solution methods for (8).

Proposition 2. *For any $\Phi \in \mathbb{R}^{r\times m}$ with $r \geq m$, let $\Phi = R\begin{bmatrix} D \\ 0 \end{bmatrix} S^\mathsf{T}$ be its SVD, i.e., $R \in \mathbb{R}^{r\times r}$, $S \in \mathbb{R}^{m\times m}$ are orthogonal and $D \in \mathbb{R}^{m\times m}$ is diagonal with $D_{ii} \geq 0$ for all i. Then a minimizer of*

$$\min_{\Lambda^\mathsf{T}\Lambda \preceq \mu^2 I} \langle \Phi, \Lambda \rangle. \tag{11}$$

is $R\begin{bmatrix} -\mu I \\ 0 \end{bmatrix} S^\mathsf{T}$, and the unique minimizer of

$$\min_{\Lambda^\mathsf{T}\Lambda \preceq \mu^2 I} \|\Phi - \Lambda\|_F^2. \tag{12}$$

is $R\begin{bmatrix} \min\{D, \mu I\} \\ 0 \end{bmatrix} S^\mathsf{T}$, where the minimum is taken entrywise.

Proof. Letting $\begin{bmatrix} F \\ G \end{bmatrix} = R^\mathsf{T}\Lambda S$ with $F \in \mathbb{R}^{m\times m}$ and $G \in \mathbb{R}^{(r-m)\times m}$, we can rewrite (11) as

$$\min_{F,G} \left\langle \begin{bmatrix} D \\ 0 \end{bmatrix}, \begin{bmatrix} F \\ G \end{bmatrix} \right\rangle = \sum_{i=1}^m D_{ii}F_{ii} \quad \text{subject to} \quad F^\mathsf{T}F + G^\mathsf{T}G \preceq \mu^2 I. \tag{13}$$

Notice that the constraint implies $F^\mathsf{T}F \preceq \mu^2 I$, which in turn implies $\sum_{j=1}^m F_{ij}^2 \leq \mu^2$ for all i, which in turn implies $F_{ii}^2 \leq \mu^2$ for all i. Thus

$$\min_{F,G} \sum_{i=1}^m D_{ii}F_{ii} \quad \text{subject to} \quad F_{ii}^2 \leq \mu^2,\ i = 1, \ldots, m, \tag{14}$$

is a relaxation of (13). Since $D_{ii} \geq 0$, it is readily seen that a minimizer of (14) is $F_{ii} = -\mu$ for all i, $F_{ij} = 0$ for all $i \neq j$, and $G = 0$. Since this solution is feasible for (13), it is a minimizer of (13) also.

Similarly, we can rewrite (12) as

$$\min_{F,G} \left\| \begin{bmatrix} D \\ 0 \end{bmatrix} - \begin{bmatrix} F \\ G \end{bmatrix} \right\|_F^2 = \|D - F\|_F^2 + \|G\|_F^2 \quad \text{subject to} \quad F^\mathsf{T}F + G^\mathsf{T}G \preceq \mu^2 I. \tag{15}$$

Notice that the constraint implies $F^\mathsf{T}F \preceq \mu^2 I$, which in turn implies $\sum_{j=1}^m F_{ij}^2 \leq \mu^2$ for all i, which in turn implies $F_{ii}^2 \leq \mu^2$ for all i. Thus

$$\min_{F,G} \|D - F\|_F^2 + \|G\|_F^2 \quad \text{subject to} \quad F_{ii}^2 \leq \mu^2,\ i = 1, \ldots, m, \tag{16}$$

is a relaxation of (15). Since $\|D - F\|_F^2 = \sum_i |D_{ii} - F_{ii}|^2 + \sum_{i\neq j} |F_{ij}|^2$, the relaxed problem (16) decomposes entrywise. Since $D_{ii} \geq 0$, it is readily seen that its unique minimizer is $F_{ii} = \min\{D_{ii}, \mu\}$ for all i, $F_{ij} = 0$ for all $i \neq j$, and $G = 0$. Since this solution is feasible for (15), it is a minimizer of (15) also. □

It can be seen that Proposition 2 readily extends to the case of $r < m$ by replacing $\begin{bmatrix} D \\ 0 \end{bmatrix}$, $\begin{bmatrix} -\mu I \\ 0 \end{bmatrix}$, $\begin{bmatrix} \min\{D, \mu I\} \\ 0 \end{bmatrix}$ with, respectively, $\begin{bmatrix} D & 0 \end{bmatrix}$, $\begin{bmatrix} -\mu I & 0 \end{bmatrix}$, $\begin{bmatrix} \min\{D, \mu I\} & 0 \end{bmatrix}$, where $D \in \mathbb{R}^{r\times r}$ is diagonal with $D_{ii} \geq 0$ for all i. Specifically, in its proof we replace $\begin{bmatrix} F \\ G \end{bmatrix}$ with $\begin{bmatrix} F & G \end{bmatrix}$, so that (13) becomes

$$\min_{F,G} \sum_{i=1}^m D_{ii}F_{ii} \quad \text{subject to} \quad \begin{bmatrix} F^\mathsf{T}F & F^\mathsf{T}G \\ G^\mathsf{T}F & G^\mathsf{T}G \end{bmatrix} \preceq \mu^2 I$$

and (15) becomes

$$\min_{F,G} \; \|D - F\|_F^2 + \|G\|_F^2 \quad \text{subject to} \quad \begin{bmatrix} F^\mathsf{T}F & F^\mathsf{T}G \\ G^\mathsf{T}F & G^\mathsf{T}G \end{bmatrix} \preceq \mu^2 I.$$

The remainder of the proof proceeds as before.

How does the dual problem (8) compare with the primal problems (1) and (2)? Unlike (1), the dual problem has a compact feasible set, which allows a duality gap bound (23) to be derived. It can also be solved by more algorithms; see Section 4.1. While (2) also has a compact feasible set, the set has a more complicated structure; see [33]. Specifically, projection onto this set has no closed form solution. Notice that the feasible sets of (2) and (8) scale with ω and μ, respectively. However, when μ is varied, warm starting by projecting the current feasible solution on to the feasible set of (8) appears to work better than scaling; see Section 6. Finally,

$$\|\nabla d(\Lambda) - \nabla d(\Lambda')\|_F = \|C(\Lambda - \Lambda')\|_F \le \lambda_{\max}(C)\|\Lambda - \Lambda'\|_F \quad \forall \Lambda, \Lambda' \in \mathbb{R}^{r\times m}, \tag{17}$$

with Lipschitz constant

$$L_{\mathrm{D}} := \lambda_{\max}(C) = \lambda_{\min}(M)^{-1} \tag{18}$$

(since $C = M^{-1}$). Thus the gradient of the objective function of (8) is Lipschitz continuous with Lipschitz constant $L_{\mathrm{D}} = \lambda_{\min}(M)^{-1}$. In contrast, the gradient of the quadratic function in (1) and (2) is Lipschitz continuous with Lipschitz constant $L_{\mathrm{P}} = \lambda_{\max}(M)$; see (25). As we shall see, the Lipschitz constant affects the iteration complexity of gradient methods for solving (1) and (8). Thus the primal and dual problems complement each other in the sense that L_{P} is small when $\lambda_{\max}(M)$ is small and L_{D} is small when $\lambda_{\min}(M)$ is large. The 'hard case' is when $\lambda_{\max}(M)$ is large and $\lambda_{\min}(M)$ is small.

4 Solution approaches

In this section we consider practical methods for solving the primal problem (1) and the dual problem (8). Due to the large problem size in multi-task learning, we consider first-order methods, specifically, gradient methods. (We also considered a primal-dual interior-point method using Nesterov-Todd direction, which is a second-order method, but the computational cost per iteration appears prohibitive for our applications). In view of Proposition 1, we assume without loss of generality that $r = n$. Then it can be seen that (1) has a unique solution W_μ which approaches the least squares solution E as $\mu \to 0$. Moreover,

$$W_\mu = 0 \quad \Longleftrightarrow \quad \mu \;\ge\; \mu_0 := \sigma_{\max}(A^\mathsf{T}B), \tag{19}$$

which follows from the optimality condition $0 \in -A^\mathsf{T}B + \mu\partial\|0\|_\star$ and $\partial\|0\|_\star$ being the unit-operator-norm ball.

4.1 Dual gradient methods

In this subsection, we consider methods for solving the reduced dual problem (8). Since its feasible set is compact convex and, by Proposition 2, minimizing a linear function over or projecting onto this feasible set requires only an SVD of an $r \times m$ matrix, we can use either a conditional gradient method or gradient-projection method. In what follows, we assume for simplicity that $r \ge m$, which holds for multi-task learning. Extension to the $r < m$ case is straighforward.

In the conditional gradient method, proposed originally by Frank and Wolfe for quadratic programming, given a feasible point Λ, we replace the objective function by its linear approximation at Λ (using $\nabla d(\Lambda) = C\Lambda + E$):

$$\min_{\Gamma^{\mathsf{T}}\Gamma \preceq \mu^2 I} \langle C\Lambda + E, \Gamma \rangle \tag{20}$$

and then do a line search on the line segment joining Λ and a solution $\hat{\Lambda}$ of (20); see [12, Section 2.2] and references therein. Specifically, letting $\Delta = \hat{\Lambda} - \Lambda$, it can be seen that

$$d(\Lambda + \alpha\Delta) = d(\Lambda) + \alpha\langle C\Lambda + E, \Delta\rangle + \frac{\alpha^2}{2}\langle \Delta, C\Delta\rangle.$$

Minimizing this over $\alpha \in [0, 1]$ yields

$$\alpha = \min\left\{1, -\frac{\langle C\Lambda + E, \Delta\rangle}{\langle \Delta, C\Delta\rangle}\right\}. \tag{21}$$

By Proposition 2, the linearized problem (20) can be solved from the SVD of $C\Lambda + E$. We thus have the following method for solving (8):

Dual conditional gradient (DCG) method:

0. Choose any Λ satisfying $\Lambda^{\mathsf{T}}\Lambda \preceq \mu^2 I$. Go to Step 1.

1. Compute the SVD:

$$C\Lambda + E = R \begin{bmatrix} D \\ 0 \end{bmatrix} S^{\mathsf{T}},$$

where $R \in \mathbb{R}^{r\times r}$, $S \in \mathbb{R}^{m\times m}$ are orthogonal, and $D \in \mathbb{R}^{m\times m}$ is diagonal with $D_{ii} \geq 0$ for all i. Set

$$\hat{\Lambda} = R \begin{bmatrix} -\mu I \\ 0 \end{bmatrix} S^{\mathsf{T}}, \qquad \Delta = \hat{\Lambda} - \Lambda.$$

Compute α by (21) and update $\Lambda^{\text{new}} = \Lambda + \alpha\Delta$. If a termination criterion is not met, go to Step 1.

It is known that every cluster point of the Λ-sequence generated by the DCG method is a solution of (8). A common choice for termination criterion is $\|\Lambda - \Lambda^{\text{new}}\|_F \leq tol$, where $tol > 0$. Another is $d(\Lambda) - d(\Lambda^{\text{new}}) \leq tol$. However, these criteria are algorithm dependent and offer no guarantee on the solution accuracy. In fact, for a fixed tol, the solution accuracy obtained can vary wildly across problem instances, which is undesirable. We will use a termination criterion, based on duality gap, with guaranteed solution accuracy. In particular, (9) shows that, as Λ approaches a solution of the dual problem (8), $W = C\Lambda + E$ approaches a solution of the primal problem (1). This suggests terminating the method when the relative duality gap is small, namely,

$$\frac{|p(W) + d(\Lambda)|}{|d(\Lambda)| + 1} < tol, \tag{22}$$

where $p(W)$ denotes the objective function of (1); see (5). To save computation, we can check this criterion every, say, 10 or r iterations.

Notice that we only need the first m columns of R in the SVD, which can be found in $O(m^2r + m^3)$ floating point operations (flops) [21, page 254]. It takes $O(m^2r)$ flops to compute $\hat{\Lambda}$, and $O(mr^2)$ flops to compute $C\Lambda + E$, as well as the numerator and denominator in (21). The rest takes $O(mr)$ flops. Thus the work per iteration is $O(mr^2)$ (since $r \geq m$). If A is sparse, then instead of storing the $r \times r$ matrix $C = (A^{\mathsf{T}}A)^{-1}$ which is typically dense, it may be more efficient to store C in a factorized form (e.g., using a combination of Cholesky factorization and rank-1 update to handle dense columns).

The above DCG method can be slow. Gradient-projection methods, proposed originally by Goldstein and Levitin and Polyak, are often faster; see [12, Section 2.3] and references therein. Gradient projection involves moving along the negative gradient direction and projecting onto the feasible set of (8) which, by Proposition 2, can be done efficiently through an SVD. Since the objective function of (8) is quadratic, exact line search can be used. The following describes a basic implementation.

Dual gradient-projection (DGP) method:

0. Choose any Λ satisfying $\Lambda^{\mathsf{T}}\Lambda \preceq \mu^2 I$ and any $L > 0$. Go to Step 1.

1. Compute the SVD:

$$\Lambda - \frac{1}{L}(C\Lambda + E) = R \begin{bmatrix} D \\ 0 \end{bmatrix} S^{\mathsf{T}},$$

where $R \in \mathbb{R}^{r\times r}$, $S \in \mathbb{R}^{m\times m}$ are orthogonal, and $D \in \mathbb{R}^{m\times m}$ is diagonal with $D_{ii} \geq 0$ for all i. Set

$$\hat{\Lambda} = R \begin{bmatrix} \min\{D, \mu I\} \\ 0 \end{bmatrix} S^{\mathsf{T}}, \qquad \Delta = \hat{\Lambda} - \Lambda.$$

Compute α by (21), and update $\Lambda^{\text{new}} = \Lambda + \alpha\Delta$. If a termination criterion is not met, go to Step 1.

Since the feasible set of (8) is compact, it can be shown that $0 \leq \upsilon + d(\Lambda) \leq \epsilon$ after $O(\frac{L_{\mathrm{D}}}{\epsilon})$ iterations; see [55, Theorem 5.1]. Like the DCG method, the work per iteration for the above DGP method is $O(mr^2)$ flops. Although global convergence of this method is guaranteed for any $L > 0$, for faster convergence, the constant L should be smaller than $\lambda_{\max}(C)$, the Lipschitz constant for the dual function d (see (17)), but not too small. In our implementation, we use $L = \lambda_{\max}(C)/8$. For the termination criterion, we use (22) with $W = C\Lambda + E$. A variant of the DGP method uses $L > L_{\mathrm{D}}/2$ and constant stepsize $\alpha = 1$, which has the advantage of avoiding computing (21).

Gradient-projection method can be accelerated using Nesterov's extrapolation technique [35–38]. Here we use Algorithm 2 described in [50], which is the simplest; also see [10] for a similar method. It is an extension of the method in [35] for unconstrained problems. (Other accelerated methods can also be used; see [36, 38, 50].) This method applies gradient projection from a Ψ that is extrapolated from Λ of the last two iterations. This method also requires a Lipschitz constant for ∇d. By (17), L_{D} is such a constant. We describe this method below.

Dual accelerated gradient-projection (DAGP) method:

0. Choose any Λ satisfying $\Lambda^{\mathsf{T}}\Lambda \preceq \mu^2 I$. Initialize $\Lambda_- = \Lambda$ and $\theta_- = \theta = 1$. Set $L = L_{\mathrm{D}}$. Go to Step 1.

1. Set

$$\Psi = \Lambda + \left(\frac{\theta}{\theta_-} - \theta\right)(\Lambda - \Lambda_-).$$

Compute the SVD:

$$\Psi - \frac{1}{L}(C\Psi + E) = R \begin{bmatrix} D \\ 0 \end{bmatrix} S^{\mathsf{T}},$$

where $R \in \mathbb{R}^{r\times r}$, $S \in \mathbb{R}^{m\times m}$ are orthogonal, and $D \in \mathbb{R}^{m\times m}$ is diagonal with $D_{ii} \geq 0$ for all i. Update

$$\Lambda^{\text{new}} = R \begin{bmatrix} \min\{D, \mu I\} \\ 0 \end{bmatrix} S^{\mathsf{T}}, \qquad \Lambda_-^{\text{new}} = \Lambda,$$

$$\theta^{\text{new}} = \frac{\sqrt{\theta^4 + 4\theta^2} - \theta^2}{2}, \qquad \theta_-^{\text{new}} = \theta.$$

If a termination criterion is not met, go to Step 1.

It can be shown that $\theta \leq \frac{2}{k+2}$ after k iterations. Moreover, $0 \leq \upsilon + d(\Lambda) \leq \epsilon$ after $O\big(\sqrt{\frac{L_{\text{D}}}{\epsilon}}\big)$ iterations; see [10,35] and [50, Corollary 2 and the proof of Corollary 1]. Like the previous two methods, the work per iteration for the above DAGP method is $O(mr^2)$ flops. Unlike the gradient-projection method, L cannot be chosen arbitrarily here (and line search does not seem to help). We also tested two other variants of the DAGP method [50, Algorithms 1 and 3], but neither performed better. We use the termination criterion (22) either with $W = C\Psi + E$ or with W initialized to the zero matrix and updated in Step 1 by

$$W^{\text{new}} = (1-\theta)W + \theta(C\Psi + E).$$

For the latter choice of W, it can be shown, by using $d(\Lambda) = \max_W \langle \Lambda, W\rangle - \frac{1}{2}\|AW - B\|_F^2$ (see (6)) and the proof of [50, Corollary 2], that

$$0 \leq p(W^{\text{new}}) + d(\Lambda^{\text{new}}) \leq \theta^2 L_{\text{D}} \Delta, \tag{23}$$

where $\Delta := \max_{\Gamma^{\mathsf{T}}\Gamma \preceq \mu^2 I} \frac{1}{2}\|\Gamma - \Lambda^{\text{init}}\|_F^2$. Since $\theta \leq \frac{2}{k+2}$ after k iterations, (22) is satisfied after at most $O\big(\sqrt{\frac{L_{\text{D}}}{tol}}\big)$ iterations. Such a duality gap bound, shown originally by Nesterov for related methods [37,38], is a special property of these accelerated methods. In our implementation, we check (22) for both choices of W.

4.2 Primal gradient method

The objective function of the primal problem (1) is the sum of a convex quadratic function

$$f_{\text{P}}(W) := \frac{1}{2}\|AW - B\|_F^2,$$

and a "simple" nonsmooth convex function $\|W\|_\star$. Moreover, we have from $\nabla f_{\text{P}}(W) = MW - A^{\mathsf{T}}B$ that

$$\|\nabla f_{\text{P}}(W) - \nabla f_{\text{P}}(W')\|_F = \|M(W - W')\|_F \leq L_{\text{P}}\|W - W'\|_F \quad \forall W, W' \in \mathbb{R}^{n\times m}, \tag{24}$$

with Lipschitz constant

$$L_{\text{P}} := \lambda_{\max}(M). \tag{25}$$

We can apply accelerated proximal gradient methods [10,39,50] to solve (1). Here we consider Algorithm 2 in [50], which is the simplest; also see FISTA in [10] for a similar method. (Other accelerated methods can also be used; see [39,50].) The same type of method was used by Toh and Yun for matrix completion [47]. It extrapolates from W of the last two iterations to obtain an $Y \in \mathbb{R}^{n\times m}$, and then solves

$$\min_W \langle \nabla f_{\text{P}}(Y), W\rangle + \frac{L}{2}\|W - Y\|_F^2 + \mu\|W\|_\star \tag{26}$$

to obtain the new W, where L is set to $L_{\rm P}$, the Lipschitz constant for $\nabla f_{\rm P}$. Like the dual gradient methods of Section 4.1, each iteration requires only one SVD of the $n \times m$ matrix $Y - \frac{1}{L}\nabla f_{\rm P}(Y)$; see [14,47]. Unlike the dual problem, we need not reduce A to have full column rank before applying this method. We describe the basic method below.

Primal accelerated proximal gradient (PAPG) method:

0. Choose any W. Initialize $W_- = W$, $\theta_- = \theta = 1$. Set $L = L_{\rm P}$. Go to Step 1.

1. Set

$$Y = W + \left(\frac{\theta}{\theta_-} - \theta\right)(W - W_-).$$

Compute the SVD:

$$Y - \frac{1}{L}(MY - A^{\mathsf T}B) = R\begin{bmatrix} D \\ 0 \end{bmatrix} S^{\mathsf T},$$

where $R \in \mathbb{R}^{n\times n}$, $S \in \mathbb{R}^{m\times m}$ are orthogonal, and $D \in \mathbb{R}^{m\times m}$ is diagonal with $D_{ii} \geq 0$ for all i. Update

$$W^{\rm new} = R\begin{bmatrix} \max\{D - \frac{\mu}{L}I, 0\} \\ 0 \end{bmatrix} S^{\mathsf T}, \qquad W_-^{\rm new} = W,$$

$$\theta^{\rm new} = \frac{\sqrt{\theta^4 + 4\theta^2} - \theta^2}{2}, \qquad \theta_-^{\rm new} = \theta.$$

If a termination criterion is not met, go to Step 1.

It can be shown that $0 \leq p(W) - \upsilon \leq \epsilon$ after $O\big(\sqrt{\frac{L_{\rm P}}{\epsilon}}\big)$ iterations; see [10], [50, Corollary 2 and the proof of Corollary 1]. In our implementation, we use the termination criterion (22) with

$$\Lambda = \mathrm{Proj}_{\mathcal{D}}(M(W - E)), \tag{27}$$

where $\mathrm{Proj}_{\mathcal{D}}(\cdot)$ denotes the nearest-point projection onto the feasible set $\mathcal{D}$ of the dual problem (8). By (10), Λ approaches a solution of the reduced dual problem (8) as W approaches a solution of (1). To save computation, we can check this criterion every, say, 10 or r iterations.

A variant of PAPG, based on a method of Nesterov for smooth constrained convex optimization [38], replaces the proximity term $\|W - Y\|_F^2$ in (26) by $\|W\|_F^2$ and replaces $\nabla f_{\rm P}(Y)$ by a sum of $\nabla f_{\rm P}(Y)/\theta$ over all past iterations; see [50, Algorithm 3]. Thus the method uses a weighted average of all past gradients instead of the most recent gradient. It also computes Y and updates W differently, and uses a specific initialization of $W = 0$. In our tests with fixed μ, this variant performed more robustly than PAPG; see Section 6. We describe this method below.

Primal accelerated proximal gradient-average (PAPG$_{\rm avg}$) method:

0. Initialize $Z = W = 0$, $\theta = 1$, $\vartheta = 0$, and $G = 0$. Set $L = L_{\rm P}$. Go to Step 1.

1. Set $Y = (1 - \theta)W + \theta Z$, and update

$$G^{\rm new} = G + \frac{1}{\theta}(MY - A^{\mathsf T}B), \qquad \vartheta^{\rm new} = \vartheta + \frac{1}{\theta}.$$

Compute the SVD:

$$-\frac{1}{L}G^{\rm new} = R\begin{bmatrix} D \\ 0 \end{bmatrix} S^{\mathsf T},$$

where $R \in \mathbb{R}^{n\times n}$, $S \in \mathbb{R}^{m\times m}$ are orthogonal, and $D \in \mathbb{R}^{m\times m}$ is diagonal with $D_{ii} \geq 0$ for all i. Update

$$Z^{\text{new}} = R \begin{bmatrix} \max\{D - \frac{\mu\vartheta^{\text{new}}}{L} I, 0\} \\ 0 \end{bmatrix} S^{\mathsf{T}}, \qquad W^{\text{new}} = (1-\theta)W + \theta Z^{\text{new}},$$

$$\theta^{\text{new}} = \frac{\sqrt{\theta^4 + 4\theta^2} - \theta^2}{2}.$$

If a termination criterion is not met, go to Step 1.

How do the primal and dual gradient methods compare? Since $C = M^{-1}$, we have that $L_{\text{D}} = \frac{1}{\lambda_{\min}(M)}$. The iteration complexity results suggest that when $L_{\text{P}} = \lambda_{\max}(M)$ is not too large, a primal gradient method (e.g., PAPG, PAPG$_{\mathbf{avg}}$) should be applied; and when $\lambda_{\min}(M)$ is not too small so that L_{D} is not too large, a dual gradient method (e.g., DCG, DGP, DAGP) should be applied. The 'tricky' case is when $\lambda_{\max}(M)$ is large and $\lambda_{\min}(M)$ is small.

5 Alternative primal formulations

In this section we derive alternative formulations of (1) that may be easier to solve than (1) and (8) when $r \leq m$. Although this case does not arise in the multi-task learning applications we consider later, it can arise when there are relatively few data points or few features.

An equivalent reformulation of the trace norm is given by (see [19, Lemma 1])

$$\|W\|_\star = \frac{1}{2} \inf_{Q\succ 0} \langle W, Q^{-1}W\rangle + \langle I, Q\rangle,$$

where $Q \in \mathbb{R}^{n\times n}$ is symmetric and positive definite. Then the primal problem (1) can be written in augmented form as

$$\upsilon \;=\; \inf_{Q\succ 0, W} \frac{1}{2}\, f(Q,W), \tag{28}$$

where

$$f(Q,W) := \|AW - B\|_F^2 + \mu\langle W, Q^{-1}W\rangle + \mu\langle I, Q\rangle. \tag{29}$$

By (5), $p(W) = \inf_{Q\succ 0} \frac{1}{2} f(Q,W)$.

Any global minimizer (Q,W) of (28) is a stationary point of f, i.e., $\nabla f(Q,W) = 0$. On the other hand, by [11, Proposition 8.5.15(xiv)], $(Q,W) \mapsto W^T Q^{-1} W$ is an operator-convex mapping. Hence f is convex, and any stationary point of f is a global minimizer. However, a stationary point may not exist due to Q becoming singular along a minimizing sequence (e.g., $B = 0$). In fact, we shall see that A having full column rank (i.e., $r = n$) is a necessary condition for a stationary point of f to exist.

5.1 Existence/uniqueness of stationary point for augmented problem

We derive below a simple necessary and sufficient condition for a stationary point of f to exist. Moreover, the stationary point, if it exists, is unique, and is obtainable from taking one matrix square root and two matrix inversions of $r \times r$ symmetric positive definite matrices.

Proposition 3. **(a)** *If a stationary point of f exists, then it is unique and is given by*

$$Q = (EE^{\mathsf{T}})^{\frac{1}{2}} - \mu C, \tag{30}$$

$$W = (\mu Q^{-1} + M)^{-1} A^{\mathsf{T}} B, \tag{31}$$

with $M \succ 0$, where M, C and E are given by (7).

(b) *A stationary point of f exists if and only if $M \succ 0$ and*

$$(EE^{\mathsf{T}})^{\frac{1}{2}} \succ \mu C. \tag{32}$$

Proof. (a) It is straightforward to verify that

$$\nabla f(Q, W) = \left(-\mu Q^{-1}WW^{\mathsf{T}}Q^{-1} + \mu I, 2\mu Q^{-1}W + 2A^{\mathsf{T}}(AW - B)\right).$$

Suppose (Q, W) is a stationary point of f, so that $Q \succ 0$ and $\nabla f(Q, W) = 0$. Then

$$\begin{aligned} (W^{\mathsf{T}}Q^{-1})^{\mathsf{T}}W^{\mathsf{T}}Q^{-1} &= I, & (33)\\ \mu Q^{-1}W + MW &= A^{\mathsf{T}}B. & (34) \end{aligned}$$

By (33), the columns of $W^{\mathsf{T}}Q^{-1}$ are orthogonal and hence $\operatorname{rank}(W) = n$. Since $M \succeq 0$ so that $\mu Q^{-1}+M \succ 0$, this implies that the left-hand side of (34) has rank n. Thus $\operatorname{rank}(A^{\mathsf{T}}B) = n$, implying $r = \operatorname{rank}(A) = n$ and hence $M = A^{\mathsf{T}}A \succ 0$.

Upon multiplying (34) on both sides by their respective transposes and using (33), we have

$$(\mu I + MQ)(\mu I + QM) = A^{\mathsf{T}}BB^{\mathsf{T}}A.$$

Since M is invertible, multiplying both sides by M^{-1} yields

$$(\mu M^{-1} + Q)^2 = M^{-1}A^{\mathsf{T}}BB^{\mathsf{T}}AM^{-1}.$$

Since $\mu M^{-1} + Q \succ 0$, this in turn yields

$$\mu M^{-1} + Q = (M^{-1}A^{\mathsf{T}}BB^{\mathsf{T}}AM^{-1})^{\frac{1}{2}}.$$

Solving for Q and using (7) yields (30). The formula (31) for W follows from (34). This shows that the stationary point of f, if it exists, is unique and given by (30) and (31).

(b) If a stationary point (Q, W) of f exists, then $Q \succ 0$ and, by (a), $M \succ 0$ and Q is given by (30). This proves (32). Conversely, assume $M \succ 0$ and (32). Let Q be give by (30). Then $Q \succ 0$. Hence the right-hand side of (31) is well defined. Let W be given by (31). Then (Q, W) satisfies (34). Let $U = W^{\mathsf{T}}Q^{-1}$. Then

$$\begin{aligned} U^{\mathsf{T}}U &= Q^{-1}WW^{\mathsf{T}}Q^{-1}\\ &= Q^{-1}(\mu Q^{-1} + M)^{-1}A^{\mathsf{T}}BB^{\mathsf{T}}A(\mu Q^{-1} + M)^{-1}Q^{-1}\\ &= (\mu I + MQ)^{-1}A^{\mathsf{T}}BB^{\mathsf{T}}A(\mu I + QM)^{-1}. \end{aligned} \tag{35}$$

We also have from (7) and (30) that

$$\begin{aligned} & & Q &= (M^{-1}A^{\mathsf{T}}BB^{\mathsf{T}}AM^{-1})^{\frac{1}{2}} - \mu M^{-1}\\ \iff & & (\mu M^{-1} + Q)^2 &= M^{-1}A^{\mathsf{T}}BB^{\mathsf{T}}AM^{-1}\\ \iff & & (\mu I + MQ)(\mu I + QM) &= A^{\mathsf{T}}BB^{\mathsf{T}}A, \end{aligned}$$

which together with (35) yields $U^{\mathsf{T}}U = I$. It follows that (Q, W) satisfies (33) and (34) and hence is a stationary point of f. □

Proposition 3 shows that $r = n$ is necessary for (28) to have a stationary point. By Proposition 1, we can without loss of generality assume that $r = n$, so that $M \succ 0$. Then it remains to check the condition (32), which is relatively cheap to do. Since E is $r \times m$, a necessary condition for (32) to hold is $r \leq m$. For Gaussian or uniformly distributed A and B, (32) appears to hold with probability near 1 whenever $r \leq m$. Thus when $r \leq m$, it is worth checking (32) first. A sufficient condition for (32) to hold is $A^{\mathsf{T}}BB^{\mathsf{T}}A \succ \mu^2 I$ (since this implies $M^{-1}A^{\mathsf{T}}BB^{\mathsf{T}}AM^{-1} \succ \mu^2 M^{-2}$ and matrix square root is operator-monotone), which is even easier to check than (32).

5.2 A reduced primal formulation

By Proposition 1, we can without loss of generality assume that $r = n$, so that $M \succ 0$. Although the minimum in (28) still may not be attained in Q, we show below that in this case (28) is reducible to a 'nice' convex optimization problem in Q for which the minimum is always attained. Since Q is $r \times r$ and symmetric, this reduced problem has lower dimension than (1) and (8) when $r \leq m$.

Consider the reduced primal function

$$h(Q) := \inf_{W} f(Q, W) \qquad \forall Q \succ 0.$$

Since $f(Q, \cdot)$ is convex quadratic, its infimum is attained at W given by (31). Plugging this into (29) and using $M = A^{\mathsf{T}} A$ and $(\mu Q^{-1} + M)^{-1} = C - \mu C(\mu C + Q)^{-1} C$ yields

$$h(Q) = \mu \langle I, Q \rangle + \mu \langle E E^{\mathsf{T}}, (\mu C + Q)^{-1} \rangle - \langle E, A^{\mathsf{T}} B \rangle + \|B\|_F^2. \tag{36}$$

Since f is convex, this and the continuity of $h(Q)$ on $Q \succeq 0$ imply $h(Q)$ is convex on $Q \succeq 0$; see for example [13, Section 3.2.5] and [45, Theorem 5.7]. Moreover, we have

$$v = \min_{Q \succeq 0} \; h(Q). \tag{37}$$

The following result shows that (37) always has a minimizer and, from any minimizer of (37), we can construct an ϵ-minimizer of (28) for any $\epsilon > 0$.

Lemma 1. *The problem* (37) *has a minimizer. Let* $Q^* \succeq 0$ *be any of its minimizers. For any* $\epsilon > 0$, *define*

$$Q_\epsilon := Q^* + \epsilon I, \qquad W_\epsilon := (\mu Q_\epsilon^{-1} + M)^{-1} A^{\mathsf{T}} B. \tag{38}$$

Then $\lim_{\epsilon \downarrow 0} f(Q_\epsilon, W_\epsilon) = \inf_{Q \succ 0, W} f(Q, W)$.

Proof. We first show that a minimizer of (37) exists. By (29), we have $f(Q, W) \geq \mu \langle I, Q \rangle$ for all $Q \succ 0$ and W. Hence $h(Q) \geq \mu \langle I, Q \rangle$. Since $h(Q)$ is continuous on $Q \succeq 0$, this holds for all $Q \succeq 0$. Then, for any $\alpha \in \mathbb{R}$, the set $\{Q \succeq 0 \mid h(Q) \leq \alpha\}$ is contained in $\{Q \succeq 0 \mid \mu \langle I, Q \rangle \leq \alpha\}$, which is bounded (since $\mu > 0$). Hence a minimizer of (37) exists.

Next, for any $\epsilon > 0$, define Q_ϵ and W_ϵ as in (38). It follows from (31) and the definition of W_ϵ that $f(Q_\epsilon, W_\epsilon) = h(Q_\epsilon)$. Then the continuity of h and the definition of Q^* yield that

$$\lim_{\epsilon \downarrow 0} f(Q_\epsilon, W_\epsilon) = \lim_{\epsilon \downarrow 0} h(Q_\epsilon) = h(Q^*) = \min_{Q \succeq 0} h(Q).$$

Since $\inf_{Q \succ 0, W} f(Q, W) = \min_{Q \succeq 0} h(Q)$, the conclusion follows. □

In view of Lemma 1, it suffices to solve (37), which then yields the optimal value and a nearly optima solution of (28). Direct calculation using (36) yields that

$$\nabla h(Q) = \mu I - \mu (\mu C + Q)^{-1} E E^{\mathsf{T}} (\mu C + Q)^{-1}. \tag{39}$$

It can be shown using $R^{-1} - S^{-1} = R^{-1}(S - R)S^{-1}$ for any nonsingular $R, S \in \mathbb{R}^{r \times r}$ that ∇h is Lipschitz continuous on the feasible set of (37) with Lipschitz constant

$$L_h = \frac{2}{\mu^2} \lambda_{\max}(E E^{\mathsf{T}}) \cdot \lambda_{\max}(M)^3.$$

Thus gradient-projection methods can be applied to solve (37). Computing $\nabla h(Q)$ by (39) requires $O(r^3)$ flops, as does projection onto the feasible set of (37), requiring an eigen-factorization. The feasible set of (37) is unbounded, so we cannot use conditional gradient method nor obtain a duality gap bound analogous to (23). For the multi-task learning applications we consider later, r is much larger than m, so (37) has a much higher dimension than (1) and (8). However, when $r \le m$ or L_h is smaller than L_{P} and L_{D}, (37) may be easier to solve than (1) and (8). Unlike L_{P} and L_{D}, L_h depends on B also (through E) and is small when $\|B\|_F$ is small. However, L_h grows cubically with $\lambda_{\max}(M)$.

5.3 An alternative SDP reformulation

By using (36), we can rewrite (37) as

$$\begin{array}{ll} \min\limits_{Q,U} & \mu\langle I,Q\rangle + \mu\langle I,U\rangle - \langle E, A^{\mathsf{T}}B\rangle + \|B\|_F^2 \\ \text{subject to} & Q \succeq 0, \qquad U \succeq E^{\mathsf{T}}(Q+\mu C)^{-1}E. \end{array}$$

Since $Q + \mu C$ is invertible for any $Q \succeq 0$, it follows from a basic property of Schur's complement that this problem is equivalent to

$$\begin{array}{ll} \min\limits_{Q,U} & \mu\langle I,Q\rangle + \mu\langle I,U\rangle - \langle E, A^{\mathsf{T}}B\rangle + \|B\|_F^2 \\ \text{subject to} & Q \succeq 0, \qquad \begin{bmatrix} Q+\mu C & E \\ E^{\mathsf{T}} & U \end{bmatrix} \succeq 0, \end{array} \tag{40}$$

where Q and U are symmetric $r \times r$ and $m \times m$ matrices respectively. It can be shown that the Lagrangian dual of (40) is reducible to the dual problem (8). For $n \ge 1000$, the size of the SDP (40) is beyond existing SDP solvers such as Sedumi, CSDP, SDPT3. Whether interior-point methods can be implemented to efficiently solve (40) requires further study.

Remark 1. *In [5, 7], a framework of regularization functions is considered. For a given spectral matrix function F, they define for any $Q \succ 0$ and any W the regularization function $(Q, W) \mapsto \langle W, F(Q)W\rangle$ and consider the following regularization problem*

$$\inf\left\{\|AW - B\|_F^2 + \mu\langle W, F(Q)W\rangle : \ Q \succ 0, \mathrm{tr}(Q) \le 1\right\}. \tag{41}$$

Observe that the regularization function is homogeneous with respect to W. They then consider, for a fixed $\epsilon > 0$, an approximate problem

$$\inf\left\{\|AW - B\|_F^2 + \mu \mathrm{tr}(F(Q)(WW^T + \epsilon I)) : \ Q \succ 0, \mathrm{tr}(Q) \le 1\right\}. \tag{42}$$

The problem (42) is then solved by alternately minimizing with respect to Q and W. The introduction of the ϵI term guarantees that, when minimizing with respect to Q for a fixed W, there is always a positive definite Q as solution, and hence alternating minimization is applicable. There is a similarity between our approaches: with $F(Q) = Q^{-1}$ in (42), the minimizer W of the objective function for fixed Q, if exists, takes the same formula as (31).

However, our formulation (29) does not fall into the framework in [5, 7] since we do not have the constraint $\mathrm{tr}(Q) \le 1$ and we have an additional $\langle I, Q\rangle$ term in the regularization function. With this extra $\langle I, Q\rangle$ term, the regularization function is no longer homogeneous with respect to W. Hence, it is not clear how to adapt the method in [7] to solve (28). In particular, we do not see an analogue of (42) for our formulation.

6 Comparing solution methods

In this section we compare the solution methods described in Section 4 on simulated and real data, with μ fixed or varying. We consider different initialization strategies, as well as warm start when μ is varying. All methods are coded in Matlab, using Matlab's rank, QR decomposition, and SVD routines. Reported times are obtained using Matlab's tic/toc timer on an HP DL360 workstation, running Red Hat Linux 3.5 and Matlab Version 7.2.

We use eight data sets in our tests. The first six are simulated data, with the entries of A generated independently and uniformly in $[0, 1]$ and the entries of B generated independently in $\{-1, 1\}$ with equal probabilities. These simulate the data in our application of gene expression pattern analysis, which we discuss in more detail in Section 7. The last two are real data from this application, with A having entries in $[0, 1]$ and B having entries in $\{-1, 1\}$ (10–30% entries are 1). These tests are used to select those methods best suited for our application. (We also ran tests with random Gaussian A and B, and the relative performances of the methods seem largely unchanged.) When $r < n$, we reduce the columns of A using its QR decomposition as described in Section 2. Table 1 shows the size of the data sets, as well as $\lambda_{\max}(M)$ and $\lambda_{\max}(C)$ and μ_0 for the reduced problem, with M and C given by (7) and μ_0 given by (19). We see that $\lambda_{\max}(M)$ is generally large, while $\lambda_{\max}(C)$ is generally small except when n is near p. As we shall see, this has a significant effect on the performance of the solution methods. Table 1 also reports time$_{\text{red}}$, which is the time (in seconds) to compute $r = \text{rank}(A)$ and, when $r < n$, to compute the reduced data matrices; see Proposition 1.

	m	n	p	$\lambda_{\max}(M)$	$\lambda_{\max}(C)$	μ_0	time$_{\text{red}}$
1	50	500	500	6e4	3e4	2e3	1e0
2	50	500	1500	2e5	4e-2	3e3	2e0
3	50	2000	1500	8e5	3e-1	6e3	9e1
4	50	2000	3500	2e6	6e-2	8e3	1e2
5	50	3000	1500	1e6	5e-2	8e3	1e2
6	50	3000	3500	3e6	6e-1	1e4	3e2
7	10	3000	2228	2e3	4e2	3e3	3e2
8	60	3000	2754	2e3	4e3	2e4	4e2

Table 1: Statistics for the eight data sets used in our tests. Floating point numbers are shown their first digits only.

We first consider solving (1) with fixed $\mu < \mu_0$. Based on the values of μ_0 in Table 1, we choose $\mu = 100, 10, 1$ in our tests. We implemented nine methods: DCG, DGP, DGP variant with $L = L_{\text{D}}/1.95$ and $\alpha = 1$, DAGP and its two variants, PAPG, PAPG$_{\text{avg}}$ and their other variant. All methods are terminated based on the relative duality gap criterion described in Section 4, with $tol = 10^{-3}$; see (22). This criterion is checked every 500 iterations. For DCG and DGP, we also terminate when the line search stepsize (21) is below 10^{-8}. For the other methods, we also terminate when the iterate changes by less than 10^{-8}, i.e., $\|\Lambda^{\text{new}} - \Lambda\|_F < 10^{-8}$ for DAGP and its variants, and $\|W^{\text{new}} - W\|_F < 10^{-8}$ for PAPG and its variants. These additional termination criteria are checked at each iteration. We cap the number of iterations at 5000. All methods except PAPG$_{\text{avg}}$ require an initial iterate. For μ large, the solution of (1) is near 0, suggesting the "zero-W (ZW) initialization" $W = 0$ for primal methods and, correspondingly, $\Lambda = \text{Proj}_{\mathcal{D}}(-ME)$ for dual methods; see (27). We also tried scaling $-ME$ instead of projecting, but it resulted in slower convergence. For μ small, the solution of (1) is near the least-squares solution E, suggesting the "least-squares (LS) initialization" $W = E$ for primal methods and, correspondingly, $\Lambda = 0$ for dual methods.

In our tests, DGP consistently outperforms DCG, while DGP performs comparably to its $\alpha = 1$ variant; DAGP mostly outperforms its two variants; and PAPG$_{\text{avg}}$ is more robust than PAPG but PAPG is useful for warm start. Also, DGP using the ZW initialization performs comparably to using the LS initialization, and DAGP using the LS initialization is never 10% slower than using the ZW initialization and is often faster. Thus we report the results for DGP (ZW initialization), DAGP (LS initialization), PAPG (ZW and LS initializations), and PAPG$_{\text{avg}}$ only. Specifically, we report in Table 2 the run time (in seconds and including time$_{\text{red}}$), number of iterations (iter), and the relative duality gap (gap), given by (22), on termination.

		PAPG (ZW init)	**PAPG** (LS init)	**PAPG$_{\text{avg}}$**	**DGP** (ZW init)	**DAGP** (LS init)
	μ	(iter/time/gap)	(iter/time/gap)	(iter/time/gap)	(iter/time/gap)	(iter/time/gap)
1	100	**500/3e1/3e-4**	max/3e2/3e-1	**500/3e1/6e-4**	max/4e2/5e-1	max/3e2/3e-2
	10	4000/2e2/6e-4	max/3e2/2	2000/1e2/7e-4	max/4e2/3e-1	**2000/1e2/6e-4**
	1	max/3e2/2e-2	max/3e2/2	max/3e2/1e-2	3500/3e2/5e-4	**500/3e1/1e-4**
2	100	1000/6e1/3e-4	1000/6e1/7e-4	1000/6e1/5e-4	**38/7/2e-16**	176/1e1/0
	10	1500/8e1/7e-4	500/3e1/5e-4	1000/6e1/8e-4	**10/4/1e-16**	**17/4/1e-16**
	1	2000/1e2/7e-4	500/3e1/4e-5	2000/1e2/5e-4	**4/4/0**	**7/4/2e-16**
3	100	2000/8e2/6e-4	max/2e3/2e-3	2500/9e2/6e-4	**64/1e2/6e-15**	459/3e2/5e-15
	10	max/2e3/1e-2	max/2e3/2e-3	max/2e3/2e-3	**22/1e2/2e-14**	**38/1e2/3e-14**
	1	max/2e3/4e-1	max/2e3/2e-3	max/2e3/3e-1	**5/1e2/4e-13**	**12/1e2/2e-13**
4	100	2500/2e3/3e-4	2000/1e3/1e-4	2000/1e3/8e-4	**22/2e2/2e-15**	**85/2e2/2e-15**
	10	max/3e3/1e-3	1000/7e2/9e-5	4000/2e3/6e-4	**9/2e2/1e-15**	**16/2e2/2e-15**
	1	max/3e3/3e-3	1000/7e2/1e-5	max/3e3/2e-3	**7/2e2/3e-15**	**7/2e2/3e-15**
5	100	3500/1e3/8e-4	1500/6e2/3e-4	3000/1e3/8e-4	**21/2e2/6e-15**	**81/2e2/7e-15**
	10	max/2e3/1e-2	3500/1e3/7e-4	max/2e3/2e-3	**7/2e2/7e-14**	**15/2e2/6e-14**
	1	max/2e3/4e-1	3500/1e3/7e-4	max/2e3/3e-1	**5/2e2/7e-13**	**7/2e2/5e-13**
6	100	2500/3e3/1e-3	max/6e3/7e-3	2500/3e3/1e-3	**84/9e2/2e-15**	500/1e3/6e-16
	10	max/6e3/1e-2	3500/5e3/7e-4	max/6e3/8e-3	**17/5e2/3e-15**	**41/5e2/2e-15**
	1	max/6e3/3e-2	3500/5e3/4e-4	max/6e3/3e-2	**7/5e2/7e-15**	**10/5e2/2e-15**
7	100	**500/4e2/2e-8**	1000/5e2/2e-4	**500/4e2/8e-7**	max/2e3/9e-3	2000/7e2/7e-4
	10	**500/4e2/9e-4**	max/1e3/2e-3	1000/5e2/9e-5	500/5e2/2e-4	**500/4e2/4e-6**
	1	5000/1e3/9e-4	max/1e3/5e-3	3500/9e2/7e-4	**74/4e2/3e-14**	**275/4e2/1e-15**
8	100	**500/1e3/1e-5**	1500/2e3/2e-4	**500/1e3/2e-6**	max/1e4/9e-2	max/7e3/3e-3
	10	**1000/2e3/9e-4**	max/7e3/2e-3	**1000/2e3/2e-4**	max/1e4/2e-3	**1000/2e3/5e-4**
	1	max/7e3/2e-3	max/7e3/2e-2	4000/6e3/7e-4	500/2e3/1e-5	**500/1e3/2e-7**

Table 2: Comparing the performances of PAPG, PAPG$_{\text{avg}}$, DGP and DAGP on data sets from Table 1, with fixed μ and using one of two initialization strategies. Floating point numbers are shown their first digits only. ("Max" indicates the number of iterations reaches the maximum limit of 5000. The lowest time is highlighted in each row.)

In general, the dual methods perform better (i.e., lower time, iter, and gap) when μ is small, while the primal method performs better when μ is large. As might be expected, when μ is large, the ZW initialization works better and, when μ is small, the LS initialization works better. Also, the primal method performs worse when $\lambda_{\max}(M)$ is large, while the dual methods perform worse when $\lambda_{\max}(C)$ is large, which corroborates the analysis at the end of Section 4.2. The dual methods DGP and DAGP have comparable performance when $\lambda_{\max}(C)$ and μ are not large, while DAGP significantly outperforms DGP when $\lambda_{\max}(C)$ is large. DAGP is also the most robust in the sense that it solves all but two instances under 5000 iterations. DGP is the next most robust, followed by PAPG$_{\text{avg}}$ and then PAPG, whose performance is very sensitive to the initialization. However, PAPG is useful for warm start, in contrast to PAPG$_{\text{avg}}$. When $p < n$ are both large, column reduction takes a significant amount of time. Thus the methods can all benefit from more efficient rank computation and QR decomposition.

Since QR decomposition is computationally expensive when $p < n$ are both large, we also applied PAPG to the primal problem (1) without the column reduction, which we call PAPG_orig. However, without column reduction, M may be singular in which case the dual function d given by (8) is undefined and we cannot use duality gap to check for termination. Instead, we use the same termination criterion as in [47], namely,

$$\frac{\|W^{\text{new}} - W\|_F}{\|W\|_F + 1} < 10^{-4}.$$

For comparison, the same termination criterion is used for PAPG. The initialization $W = 0$ is used for both. Perhaps not surprisingly, PAPG and PAPG_orig terminate in the same number of iterations with the same objective value. Their run times are comparable (within 20% of each other) on data sets 1, 2, 4, 6. On 3 and 5, PAPG_orig is about 1.5–3 times slower than PAPG for all three values of μ. On 7 and 8, PAPG is about 3 times slower than PAPG_orig for $\mu = 100$ and comparable for other values of μ. This seems to be due to a combination of (i) very high cost of column reduction in PAPG (see Table 1) and (ii) relatively few iterations in PAPG_orig (due to 0 being near the solution), so the latter's higher computational cost per iteration is not so significant. Thus, it appears that PAPG_orig has an advantage over PAPG mainly when $p < n$ are both large (so column reduction is expensive) and a good initial W is known, such as when μ is large ($W = 0$) or when μ is small (W being the least squares solution).

We next consider solving (1) with varying $\mu < \mu_0$, for which warm start can be used to resolve the problem as μ varies. In our tests, we use $\mu = 100, 50, 10, 5, 1, 0.1$ in descending order, which roughly covers the range of μ values used in our application in Section 7. For simplicity, we use only the last four data sets from Table 1. As for the solution method, the results in Tables 1 and 2 suggest using PAPG when μ and $\lambda_{\max}(C)$ are large, and otherwise use DAGP. After some experimentation, we settled on the following switching rule: use PAPG when

$$\mu \geq \frac{\lambda_{\max}(M)}{\frac{r}{2}\lambda_{\max}(C) + \lambda_{\max}(M)}\,\mu_0, \tag{43}$$

and otherwise use DAGP, where μ_0 is given by (19) and $r = \text{rank}(A)$. We initialize PAPG with $W = 0$ and DAGP with $\Lambda = 0$, as is consistent with Table 2. The column reduction described in Section 2 is done once at the beginning. The reduced data matrices are then used for all values of μ. The termination criteria for each μ value are the same as in our tests with fixed μ. Each time when μ is decreased, we use the solutions W and Λ found for the old μ value to initialize either PAPG or DAGP for the new μ value. For DAGP, we further project Λ onto the feasible set of (8) corresponding to the new μ value (since the new feasible set is smaller than before). We also tried scaling Λ instead of projecting, but the resulting method performed worse.

In Table 3, we report the performance of this primal-dual method. It shows, for each μ value, the method used (p/d), the (cumulative) run time (in seconds), the (cumulative) number of iterations (iter), and the relative duality gap (gap), computed as in (22), on termination. Compared with Table 2, the switching rule (43) seems effective in selecting the "right" method for each μ.

7 Experiment

The *Drosophila* gene expression pattern images document the spatial and temporal dynamics of gene expression and provide valuable resources for explicating the gene functions, interactions, and networks during *Drosophila* embryogenesis [29]. To provide text-based pattern searching, the images in the Berkeley *Drosophila* Genome Project (BDGP) are annotated with Controlled Vocabulary (CV) terms manually

	$\mu = 100$	$\mu = 50$	$\mu = 10$	$\mu = 5$	$\mu = 1$	$\mu = 0.1$
	(alg/iter/time/gap)	(alg/iter/time/gap)	(alg/iter/time/gap)	(alg/iter/time/gap)	(alg/iter/time/gap)	(alg/iter/time/gap)
5	d/81/2e2/7e-15	d/121/2e2/1e-14	d/136/2e2/4e-14	d/146/2e2/1e-13	d/151/2e2/5e-13	d/153/2e2/3e-12
6	d/500/1e3/2e-15	d/708/1e3/7e-16	d/750/1e3/6e-15	d/773/1e3/4e-15	d/783/1e3/6e-15	d/787/1e3/6e-15
7	p/500/4e2/2e-8	p/1000/5e2/4e-5	d/1500/6e2/1e-6	d/2000/7e2/1e-8	d/2317/7e2/6e-14	d/2340/7e2/2e-13
8	p/500/1e3/1e-5	p/1000/2e3/8e-5	p/2000/3e3/9e-4	d/2500/4e3/5e-5	d/3000/4e3/1e-7	d/3452/5e3/4e-12

Table 3: The performances of the primal-dual method on the last four data sets from Table 1, with varying μ and using warm start. Floating point numbers are shown their first digits only. ("p" stands for PAPG and "d" stands for DAGP.)

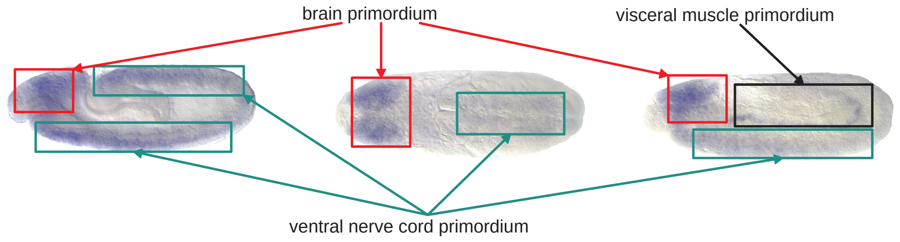

Figure 1: Three CV terms assigned to a group of three images from the gene *Actn* in the stage range 11-12 and their corresponding local expression patterns. The left and right images are taken from the lateral view, and the middle image is taken from the dorsal view.

by human curators [48]. In particular, images in BDGP are annotated collectively in small groups based on the genes and the developmental stages[1] (Figure 1). It has been shown that different CV terms are correlated [28, 49] due to the co-expression of a large number of genes. It is thus desirable to capture such correlation information in the development of automated CV term annotation systems. In this experimental study, we demonstrate the effectiveness of the proposed methods for CV term annotation.

We first represent each image group as a feature vector based on the bag-of-words and the sparse coding [25, 28]. Since each CV term is often associated with only a part of an image (see Figure 1), we extract *local visual features*, which are invariant to certain geometric transformations (e.g., scaling and translation) [32]. We then apply the k-means clustering algorithm on a subset of the local features and the cluster centers are used as the "visual words" in the codebook. Each image will then be represented as a bag of visual words, and the resulting representation is often referred to as a "bag-of-words" representation. Specifically, let $\Omega = [a_1, \cdots, a_d] \in \mathbb{R}^{r \times d}$ denote the codebook matrix in which $\{a_i\}_{i=1}^d$ are the codebook words obtained from the clustering algorithm. Given a local feature vector u, the traditional bag-of-words approach assigns u to the closest visual words and represents u as a d-dimensional vector x in which the entry corresponding to the closest visual word is one, and all other entries are zero. This reduces to computing x by solving the following optimization problem:

$$\min_x \frac{1}{2} ||\Omega x - u||^2 \tag{44}$$

subject to the constraint that only one entry in x is one, and all other entries are zero. Hence, an entire image group is represented as a global histogram counting the number of occurrences of each word in the codebook. We call this approach the "hard assignment" approach as one feature is only assigned to one codebook word.

[1] The images in BDGP are grouped into six stage ranges: 1-3, 4-6, 7-8, 9-10, 11-12 and 13-16.

In addition, we consider the "soft assignment" approach based on the sparse coding, in which each feature is assigned to multiple codebook words simultaneously. Specifically, we include an additional 1-norm term to the objective in (44), resulting in the following sparse coding formulation [28]:

$$\begin{array}{ll} \min\limits_{x} & \frac{1}{2}||\Omega x - u||^2 + \lambda ||x||_1 \\ \text{subject to} & x_i \geq 0, \text{ for } i = 1, \cdots, d, \end{array} \tag{45}$$

where $\lambda > 0$ is a tunable parameter and $||x||_1 = \sum_{i=1}^{d} |x_i|$ denotes the vector 1-norm.

In our implementation the local features used are the SIFT features [32] and the number of visual words is set to 3000. The entire data set is partitioned into 6 subsets according to developmental stage ranges, as most of the terms are stage range specific. Note that each of the 6 subsets are used independently. For each subset, we extract a number of data sets based on the number of terms, resulting in multiple data sets for each stage range. The regularized least squares formulation (4) of this multi-task learning problem is solved using the primal-dual method of Section 6 (**RLSPD**). For our image annotation task, x_i corresponds to the feature vector[2] of the i-th image group and y_i^ℓ equals to 1 if the i-th image group is associated with the ℓ-th CV term, and 0 otherwise. We employ the linear predictor $f_\ell(x) = w_\ell^T x$ for the prediction of the ℓ-th CV term, where w_ℓ is the i-th column of W. The parameter μ is tuned via 5-fold cross-validation from the range $\{0.02, 0.1, 1, 2, 3, 5, 6, 8, 10, 12, 15, 20, 50, 70, 100\}$. Other μ values below 0.02 or above 100 were also tested initially, but they were never selected.

We compare the performance of different approaches in terms of AUC, macro F1, and micro F1 [17,30]. The AUC (the area under the receiver operating characteristic curve) can be interpreted as the probability that, when we randomly pick one positive and one negative example, the classifier will assign a higher score to the positive example than to the negative example. The F1 score is the harmonic mean of precision and recall [30, Section 5.3]. To measure the performance across multiple terms, we use both the macro F1 (average of F1 across all terms) and the micro F1 (F1 computed from the sum of per-term contingency tables) scores, which are commonly used in text and image applications [17].

The results on 15 data sets in stage ranges 7-8, 9-10, 11-12 and 13-16 are summarized in Tables 4–7. We report the performance of the support vector machines (SVM) in which each term is classified in the one-against-rest manner [44]. For **RLSPD** and SVM, we report the performance based on both the hard assignment and the soft assignment representations. We also report the performance of the existing methods based on the pyramid match kernel [26], denoted as PMK_{star}, $\text{PMK}_{\text{clique}}$, and PMK_{cca}. We can observe from the tables that **RLSPD** outperforms SVM and the existing methods significantly. In addition, the soft assignment representation outperforms the hard assignment representation consistently.

8 Conclusions

In this paper we derived new primal and dual reformulations of the trace norm regularized least squares problem that, depending on the problem dimensions and the regularization parameter μ, are more easily solved by first-order gradient methods. Based on this, a hybrid primal-dual method for solving the problem with varying μ is developed and is shown to be effective for multi-task learning, both in simulations and in an application to gene expression pattern analysis.

Acknowledgement. We thank Defeng Sun for suggestions on Matlab coding that significantly improved efficiency. We would also like to thank the anonymous referees for their suggestions and for pointing to us references [6, 7].

[2] We consider both the hard assignment representation based on the bags-of-words approach and the soft assignment representation based on the sparse coding.

Table 4: Summary of performance in terms of AUC (top section), macro F1 (middle section), and micro F1 (bottom section) for stage range 7-8. $\mathbf{RLSPD}_\mathrm{h}$ and $\mathbf{RLSPD}_\mathrm{s}$ refer to the proposed **RLSPD** method applied to the hard and soft assignment representations, respectively. Similarly, SVM_h and SVM_s refer to SVM applied to the hard and soft assignment representations, respectively. $\mathrm{PMK}_\mathrm{star}$, $\mathrm{PMK}_\mathrm{clique}$, and $\mathrm{PMK}_\mathrm{cca}$ are three existing methods based on the pyramid match kernel [26].

m	$\mathbf{RLSPD}_\mathrm{h}$	$\mathbf{RLSPD}_\mathrm{s}$	SVM_h	SVM_s	$\mathrm{PMK}_\mathrm{star}$	$\mathrm{PMK}_\mathrm{clique}$	$\mathrm{PMK}_\mathrm{cca}$
10	76.44±0.84	78.03±0.84	73.65±0.91	77.02±0.91	71.92±0.97	72.35±0.90	72.27±0.78
20	76.52±1.00	78.98±1.03	72.05±1.30	74.62±1.40	69.47±1.17	69.25±1.22	69.79±1.18
10	48.43±1.33	51.84±1.15	48.03±1.07	52.22±1.27	44.75±1.31	46.35±1.11	43.32±1.18
20	29.98±1.32	34.03±1.65	32.37±1.04	34.63±1.40	26.63±1.02	26.92±1.03	23.43±0.97
10	56.48±1.34	58.48±1.16	53.36±1.24	55.37±1.34	52.35±1.56	51.94±1.42	52.65±1.39
20	52.87±1.24	55.82±1.21	50.25±1.32	53.29±1.52	49.57±1.34	48.07±1.39	49.47±1.25

Table 5: Summary of performance in terms of AUC (top section), macro F1 (middle section), and micro F1 (bottom section) for stage range 9-10. See the caption of Table 4 for details.

m	$\mathbf{RLSPD}_\mathrm{h}$	$\mathbf{RLSPD}_\mathrm{s}$	SVM_h	SVM_s	$\mathrm{PMK}_\mathrm{star}$	$\mathrm{PMK}_\mathrm{clique}$	$\mathrm{PMK}_\mathrm{cca}$
10	77.22±0.63	78.86±0.58	74.89±0.68	78.51±0.60	71.80±0.81	71.98±0.87	72.28±0.72
20	78.95±0.82	80.90±1.02	76.38±0.84	78.94±0.86	72.01±1.01	71.70±1.20	72.75±0.99
10	52.57±1.19	54.89±1.24	52.25±0.98	55.64±0.69	46.20±1.18	47.06±1.16	45.17±1.06
20	33.15±1.37	37.25±1.25	35.62±0.99	39.18±1.18	28.21±1.00	28.11±1.09	25.45±0.83
10	59.92±1.04	60.84±0.99	55.74±1.02	59.27±0.80	53.25±1.15	53.36±1.20	54.11±0.95
20	55.33±0.88	56.79±0.72	51.70±1.17	54.25±0.93	49.59±1.24	48.14±1.34	49.80±1.09

References

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

[2] Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering shared structures in multiclass classification. In *ICML*, pages 17–24, 2007.

[3] R. K. Ando. BioCreative II gene mention tagging system at IBM Watson. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, 2007.

[4] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

[5] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[6] A. Argyriou, A. Maurer, and M. Pontil. An algorithm for transfer learning in a heterogeneous environment. In *ECML/PKDD*, page 85, 2008.

[7] A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying. A spectral regularization framework for multi-task structure learning. In *NIPS*, 2007.

Table 6: Summary of performance in terms of AUC (top section), macro F1 (middle section), and micro F1 (bottom section) for stage range 11-12. See the caption of Table 4 for details.

m	$\mathbf{RLSPD}_\mathrm{h}$	$\mathbf{RLSPD}_\mathrm{s}$	SVM_h	SVM_s	$\mathrm{PMK}_\mathrm{star}$	$\mathrm{PMK}_\mathrm{clique}$	$\mathrm{PMK}_\mathrm{cca}$
10	84.06±0.53	86.18±0.45	83.05±0.54	84.84±0.57	78.68±0.58	78.52±0.55	78.64±0.57
20	84.37±0.36	86.59±0.28	82.73±0.38	84.43±0.27	76.44±0.68	76.01±0.64	76.97±0.67
30	81.83±0.46	83.85±0.36	79.18±0.51	81.31±0.48	71.85±0.61	71.13±0.53	72.90±0.63
40	81.04±0.57	82.92±0.57	77.52±0.63	80.29±0.54	71.12±0.59	70.28±0.66	72.12±0.64
50	80.56±0.53	82.87±0.53	76.19±0.72	78.75±0.68	69.66±0.81	68.80±0.68	70.73±0.83
10	60.30±0.92	64.00±0.85	60.37±0.88	62.61±0.82	54.61±0.68	55.19±0.62	53.25±0.87
20	48.23±0.94	51.81±0.93	46.03±0.83	49.87±0.59	33.61±0.83	36.11±0.88	31.07±0.74
30	35.20±0.85	39.15±0.83	35.32±0.75	37.38±0.95	22.30±0.70	24.85±0.63	20.44±0.38
40	28.25±0.95	31.85±1.09	27.85±0.79	30.83±0.85	17.14±0.53	19.01±0.63	15.36±0.42
50	23.07±0.86	26.67±1.05	23.46±0.60	26.26±0.65	14.07±0.48	15.04±0.46	12.40±0.39
10	66.89±0.79	68.92±0.68	65.67±0.60	66.73±0.74	62.06±0.54	61.84±0.51	62.55±0.68
20	59.91±0.61	62.37±0.69	54.59±0.87	57.33±0.74	51.77±0.56	50.51±0.54	53.61±0.59
30	55.66±0.64	56.70±0.68	48.87±0.85	51.52±0.96	47.08±0.81	44.81±0.66	49.61±0.64
40	53.76±0.69	54.47±0.83	46.40±0.91	50.36±1.00	45.41±0.58	43.33±0.70	48.10±0.75
50	52.92±0.78	54.54±0.70	47.18±0.84	47.97±0.90	44.25±0.65	42.49±0.70	47.41±0.61

[8] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99, 2003.

[9] J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

[10] A. Beck and M. Teboulle. Fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2008. In press.

[11] D. S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear System Theory.* Princeton University Press, 2005.

[12] D. P. Bertsekas. *Nonlinear Programming, 2nd edition.* Athena Scientific, Belmont, 1999.

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

[14] J.-F. Cai, E. J. Candés, and Z. Shen. A singular value thresholding algorithm for matrix completion. Technical Report 08-77, UCLA Computational and Applied Math., 2008.

[15] E. J. Candés and B. Recht. Exact matrix completion via convex optimization. Technical Report 08-76, UCLA Computational and Applied Math., 2008.

[16] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[17] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.

[18] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

Table 7: Summary of performance in terms of AUC (top section), macro F1 (middle section), and micro F1 (bottom section) for stage range 13-16. See the caption of Table 4 for details.

m	**RLSPD**$_{\mathrm{h}}$	**RLSPD**$_{\mathrm{s}}$	SVM$_{\mathrm{h}}$	SVM$_{\mathrm{s}}$	PMK$_{\mathrm{star}}$	PMK$_{\mathrm{clique}}$	PMK$_{\mathrm{cca}}$
10	87.38±0.36	89.43±0.31	86.66±0.35	88.42±0.35	82.07±0.41	82.53±0.62	81.87±0.76
20	84.57±0.34	87.25±0.34	83.25±0.35	85.36±0.36	76.46±0.36	77.09±0.55	77.02±0.52
30	82.76±0.36	85.86±0.34	81.13±0.46	83.45±0.38	73.34±0.46	73.73±0.52	74.05±0.68
40	81.61±0.31	84.68±0.33	79.56±0.32	82.04±0.30	70.61±0.57	70.84±0.51	71.48±0.45
50	80.52±0.34	83.43±0.32	78.17±0.38	80.61±0.36	68.38±0.51	68.67±0.59	69.15±0.73
60	80.17±0.40	83.32±0.45	77.18±0.46	79.75±0.47	67.15±0.57	67.11±0.64	68.24±0.48
10	64.43±0.77	67.42±0.78	62.97±0.68	66.38±0.71	57.37±0.91	58.42±0.94	57.54±0.96
20	49.98±0.81	54.13±0.86	50.45±0.62	52.75±0.79	40.02±0.64	41.02±0.72	37.86±0.81
30	42.48±0.87	47.39±0.91	41.92±0.76	45.07±0.68	29.62±0.67	31.04±0.82	27.40±0.71
40	34.72±0.63	40.66±0.71	35.26±0.63	38.67±0.54	22.40±0.58	23.78±0.48	20.55±0.53
50	28.70±0.91	34.12±0.93	29.74±0.45	33.22±0.57	18.44±0.47	19.22±0.53	16.65±0.57
60	24.78±0.67	29.84±0.62	25.49±0.55	28.72±0.57	15.65±0.46	16.13±0.48	14.01±0.46
10	67.85±0.60	70.50±0.58	66.67±0.45	68.79±0.60	60.98±0.74	61.87±0.77	62.07±0.90
20	58.25±0.69	61.21±0.68	55.66±0.65	57.67±0.89	48.74±0.51	49.88±0.70	50.56±0.59
30	53.74±0.45	57.04±0.69	48.11±0.90	51.19±0.83	43.50±0.70	44.14±0.78	45.48±0.82
40	50.68±0.47	53.93±0.49	44.26±0.92	47.96±0.80	40.28±0.75	41.02±0.65	42.55±0.67
50	49.45±0.60	51.57±0.57	43.53±0.77	45.18±0.76	38.63±0.51	39.90±0.52	41.07±0.89
60	48.79±0.60	51.35±0.58	42.84±0.76	44.48±0.84	37.28±0.81	38.29±0.78	40.65±0.49

[19] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, pages 4734–4739, 2001.

[20] M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *Proceedings of the American Control Conference*, pages 2156–2162, 2003.

[21] G. H. Golub and C. F. Van Loan. *Matrix Computation, 3rd edition.* Johns Hopkins University Press, Baltimore, 1996.

[22] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. In *NIPS*, 2001.

[23] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, 2005.

[24] L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: A convex formulation. In *NIPS*, 2008.

[25] S. Ji, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye. A bag-of-words approach for drosophila gene expression pattern annotation. *BMC Bioinformatics*, 10(1):119, 2009.

[26] S. Ji, L. Sun, R. Jin, S. Kumar, and J. Ye. Automated annotation of *Drosophila* gene expression patterns using a controlled vocabulary. *Bioinformatics*, 24(17):1881–1888, 2008.

[27] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *ICML*, 2009.

[28] S. Ji, L. Yuan, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye. *Drosophila* gene expression pattern annotation using sparse features and term-term interactions. In *KDD*, 2009.

[29] E. Lécuyer and P. Tomancak. Mapping the gene expression universe. *Current Opinion in Genetics & Development*, 2008.

[30] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[31] Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. Technical report, UCLA Electrical Engineering Department, 2008.

[32] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[33] Z. Lu, R. D. C. Monteiro, and M. Yuan. Convex optimization methods for dimension reduction and coefficient estimation in multivariate linear regression. *Submitted to Mathematical Programming*, 2008 (revised March 2009).

[34] S. Ma, D. Goldfarb, and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. Technical Report 08-78, UCLA Computational and Applied Math., 2008.

[35] Y. Nesterov. A method for solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Dokl.*, 27(2):372–376, 1983.

[36] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Kluwer Academic Publishers, 2003.

[37] Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization*, 16(1):235–249, 2005.

[38] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[39] Y. Nesterov. Gradient methods for minimizing composite objective function. Technical Report 2007/76, CORE, Université catholique de Louvain, 2007.

[40] G. Obozinski, B. Taskar, and M. I. Jordan. Multi-task feature selection. In *Technical report, Dept. of Statistics, UC Berkeley*, 2006.

[41] G. Obozinski, B. Taskar, and M. I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 2009. In press.

[42] A. Quattoni, M. Collins, and T. Darrell. Learning visual representations using images with captions. In *CVPR*, 2007.

[43] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, pages 713–719, 2005.

[44] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *J. Mach. Learn. Res.*, 5:101–141, 2004.

[45] R. T. Rockafellar. *Convex Analysis.* Princeton University Press, 1970.

[46] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *NIPS*, pages 1329–1336. 2005.

[47] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. Technical report, Department of Mathematics, National University of Singapore, Singapore, 2009.

[48] P. Tomancak, A. Beaton, R. Weiszmann, E. Kwan, S. Q. Shu, S. E. Lewis, S. Richards, M. Ashburner, V. Hartenstein, S. E. Celniker, and G. Rubin. Systematic determination of patterns of gene expression during *Drosophila* embryogenesis. *Genome Biology*, 3(12), 2002.

[49] P. Tomancak, B. Berman, A. Beaton, R. Weiszmann, E. Kwan, V. Hartenstein, S. Celniker, and G. Rubin. Global analysis of patterns of gene expression during *Drosophila* embryogenesis. *Genome Biology*, 8(7):R145, 2007.

[50] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2008.

[51] M. Weimer, A. Karatzoglou, Q. Le, and A. Smola. $\text{COFI}^{\text{rank}}$ - maximum margin matrix factorization for collaborative ranking. In *NIPS*, pages 1593–1600. 2008.

[52] M. Weimer, A. Karatzoglou, and A. Smola. Improving maximum margin matrix factorization. *Machine Learning*, 72(3):263–276, 2008.

[53] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.

[54] M. Yuan, A. Ekici, Z. Lu, and R. Monteiro. Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B*, 69(3):329–346, 2007.

[55] S. Yun and P. Tseng. A block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. *Journal of Optimization Theory and Applications*, 140, 2009.