Efficient Multi-Task Feature Learning with Calibration

Pinghua Gong[1,2], Jiayu Zhou[1,2], Wei Fan[3], Jieping Ye[1,2]
[1]Center for Evolutionary Medicine and Informatics, The Biodesign Institute, ASU, Tempe, AZ 85287
[2]Department of Computer Science and Engineering, ASU, Tempe, AZ 85287
[3]Huawei Noah's Ark Lab, Hong Kong, China
[1,2]{pinghua.gong, jiayu.zhou, jieping.ye}@asu.edu, [3]david.fanwei@huawei.com

ABSTRACT

Multi-task feature learning has been proposed to improve the generalization performance by learning the shared features among multiple related tasks and it has been successfully applied to many real-world problems in machine learning, data mining, computer vision and bioinformatics. Most existing multi-task feature learning models simply assume a common noise level for all tasks, which may not be the case in real applications. Recently, a Calibrated Multivariate Regression (CMR) model has been proposed, which calibrates different tasks with respect to their noise levels and achieves superior prediction performance over the non-calibrated one. A major challenge is how to solve the CMR model efficiently as it is formulated as a composite optimization problem consisting of two non-smooth terms. In this paper, we propose a variant of the calibrated multi-task feature learning formulation by including a squared norm regularizer. We show that the dual problem of the proposed formulation is a smooth optimization problem with a piecewise sphere constraint. The simplicity of the dual problem enables us to develop fast dual optimization algorithms with low per-iteration cost. We also provide a detailed convergence analysis for the proposed dual optimization algorithm. Empirical studies demonstrate that, the dual optimization algorithm quickly converges and it is much more efficient than the primal optimization algorithm. Moreover, the calibrated multi-task feature learning algorithms with and without the squared norm regularizer achieve similar prediction performance and both outperform the non-calibrated ones. Thus, the proposed variant not only enables us to develop fast optimization algorithms, but also keeps the superior prediction performance of the calibrated multi-task feature learning over the non-calibrated one.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

KDD'14, August 24–27, 2014, New York, NY, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623641.

Keywords

Multi-task learning; feature selection; calibration; dual problem; accelerated gradient descent

1. INTRODUCTION

Multi-task feature learning aims to improve the generalization performance by learning multiple related tasks together and exploring the shared features among tasks. It has received a lot of interests and has been successfully applied to a wide range of applications including gene data analysis [18], breast cancer classification [34], neural semantic basis discovery [21] and disease progression prediction [36, 37]. Most existing multi-task feature learning models can be formulated as a regularized optimization problem and they usually focus on how to design a good regularizer to capture the underlying shared features among tasks; examples include group lasso multi-task feature learning [1, 19, 23, 25, 29, 33, 34], tree-guided group lasso multi-task feature learning [18], composite regularized multi-task feature learning [14, 16] and non-convex regularized multi-task feature learning [13, 15].

However, most existing multi-task feature learning models simply assume a common noise level for all tasks, which may not hold in real applications. Moreover, theoretical analysis in Lounici et al. [26] shows that, to achieve the optimal parameter estimation error bounds, the regularized parameter should be chosen in proportion to the maximum standard deviations of the noise for all tasks. In practice, the standard deviations of the noise are unknown or very difficult to estimate, which makes the parameter tuning quite challenging. To this end, Liu et al. [22] propose a Calibrated Multivariate Regression (CMR) model which calibrates each task by employing different noise levels for all tasks and achieve superior prediction performance over the non-calibrated one. Moreover, their theoretical analysis shows that the CMR model can achieve the optimal parameter estimation error bounds by properly tuning the regularized parameter which does not depend on the standard deviations of the noise for all tasks. This makes the parameter tuning of the CMR model much more insensitive to the noise levels. However, one major challenge in the practical use of the CMR model is that it is formulated as a composite optimization problem consisting of two non-smooth terms, which makes the optimization problem challenging to solve.

In this paper, inspired by the great success of the elastic net [38], we propose a variant of the calibrated multi-task feature learning formulation by including a squared norm regularizer. The major contributions of this paper include:

(1) We show that the dual problem of the proposed formulation is a smooth optimization problem with a piecewise sphere constraint. (2) The simplicity of the dual problem enables us to develop fast dual optimization algorithms with low per-iteration cost. (3) We provide a detailed convergence analysis for the proposed algorithm. (4) Experimental results demonstrate that, the dual optimization algorithm quickly converges and it is much more efficient than the primal optimization algorithm. Moreover, the calibrated multi-task feature learning algorithms with and without the squared norm regularizer achieve similar prediction performance and both outperform the non-calibrated ones. These results demonstrate that the proposed variant not only enables us to develop fast optimization algorithms, but also keeps the superior prediction performance of the calibrated multi-task feature learning over the non-calibrated one.

The rest of the paper is organized as follows: We introduce preliminaries in Section 2. We propose a variant of the multi-task feature formulation and develop fast optimization algorithms in Section 3. We report experimental results in Section 4 and we conclude in Section 5.

2. PRELIMINARIES

Assume that we are given m learning tasks associated with the data $\{(X_1, \mathbf{y}_1), \cdots, (X_m, \mathbf{y}_m)\}$ and a linear model:

$$\mathbf{y}_i = X_i \mathbf{w}_i^\star + \boldsymbol{\epsilon}_i,\ i \in \{1, \cdots, m\},$$

where $X_i \in \mathbb{R}^{n_i \times d}$ is the data matrix and $\mathbf{y}_i \in \mathbb{R}^{n_i}$ is the response vector for the i-th task, respectively; $\mathbf{w}_i^\star \in \mathbb{R}^d$ is the underlying true weight and $\boldsymbol{\epsilon}_i \in \mathbb{R}^{n_i}$ is the noise vector for the i-th task, respectively; each entry of $\boldsymbol{\epsilon}_i$ is sampled from the normal distribution with mean zero and standard deviation σ_i. The ordinary (non-calibrated) multi-task feature learning model is formulated as the following problem:

$$\min_{W \in \mathbb{R}^{d \times m}} \left\{ \sum_{i=1}^m \|X_i \mathbf{w}_i - \mathbf{y}_i\|^2 + \lambda r(W) \right\}, \qquad (1)$$

where $\|\cdot\|$ is the Euclidean norm; $W = [\mathbf{w}_1, \cdots, \mathbf{w}_m] \in \mathbb{R}^{d \times m}$ is the weight matrix with the i-th column $\mathbf{w}_i \in \mathbb{R}^d$ being the weight vector for the i-th task; $\lambda > 0$ is a regularized parameter; $r(W)$ is a model-specific regularizer (such as group lasso, tree-guided group lasso, a composite regularizer and a non-convex regularizer). To calibrate each task by considering the different noise levels of all tasks, Liu et al. [22] propose to use the square-root function [6, 11] instead of the least squares function, resulting in the following Calibrated Multivariate Regression (CMR) model:

$$\min_{W \in \mathbb{R}^{d \times m}} \left\{ \sum_{i=1}^m \|X_i \mathbf{w}_i - \mathbf{y}_i\| + \lambda r(W) \right\}. \qquad (2)$$

Liu et al. [22] interpret the CMR model in Eq. (2) as the following weighted regularized least squares problem:

$$\min_{W \in \mathbb{R}^{d \times m}} \left\{ \sum_{i=1}^m \frac{1}{\sigma_i \sqrt{n_i}} \|X_i \mathbf{w}_i - \mathbf{y}_i\|^2 + \lambda r(W) \right\}, \qquad (3)$$

where $\frac{1}{\sigma_i \sqrt{n_i}}$ is the weight for calibrating the i-th task. When we do not have any prior knowledge on σ_i's, σ_i is chosen as

$$\sigma_i = \frac{1}{\sqrt{n_i}} \|X_i \mathbf{w}_i - \mathbf{y}_i\|,\ i \in \{1, \cdots, m\}. \qquad (4)$$

Thus, using the weight defined in Eq. (4), the CMR model in Eq. (2) calibrates different tasks via solving the weighted regularized least squares model in Eq. (3). Although the Robust Feature Selection (RFS) algorithm in [28] also uses a similar square-root loss, it is quite different from the CMR model in Eq. (2) that RFS "calibrates" different samples but not different tasks via the square-root loss. Notice that the regularizer and the loss function in Eq. (2) are usually both non-smooth, which makes the optimization problem challenging to solve.

3. FORMULATION AND EFFICIENT OPTIMIZATION ALGORITHMS

Inspired by the great success of the elastic net [38], we propose a variant of the calibrated multi-task feature learning model by adding a squared norm regularizer as follows:

$$\min_{W \in \mathbb{R}^{d \times m}} \left\{ \sum_{i=1}^m \|X_i \mathbf{w}_i - \mathbf{y}_i\| + \lambda_1 \|W\|_{1,2} + \frac{\lambda_2}{2} \|W\|_F^2 \right\}, \qquad (5)$$

where $\|W\|_{1,2} = \sum_{j=1}^d \|\mathbf{w}^j\|$ with $\mathbf{w}^j$ being the j-th row of W; $\|W\|_F^2 = \sum_{i=1}^m \sum_{j=1}^d w_{ij}^2$ with w_{ij} being the (i, j)-th entry of W; $\lambda_1, \lambda_2 > 0$ are regularized parameters. The major benefit of the proposed formulation is that we can derive a smooth dual optimization problem with a piecewise sphere constraint. The simplicity of the dual problem enables us to develop fast optimization algorithms with low per-iteration cost. Moreover, the proposed variant keeps the superior prediction performance of the calibrated multi-task feature learning over the non-calibrated one.

3.1 The Dual Ascent Optimization Algorithm

We first derive the dual problem of the optimization problem in Eq. (5) and then present key properties of the dual problem. Finally, we propose fast algorithms to solve the dual problem and provide detailed convergence analysis accordingly. Let $\mathbf{z}_i = X_i \mathbf{w}_i - \mathbf{y}_i$. Then Eq. (5) is equivalent to the following constrained optimization problem:

$$\begin{aligned} &\min_{W, \mathbf{z}} \left\{ \sum_{i=1}^m \|\mathbf{z}_i\| + \lambda_1 \|W\|_{1,2} + \frac{\lambda_2}{2} \|W\|_F^2 \right\}, \\ &s.t.\ \ \mathbf{z}_i = X_i \mathbf{w}_i - \mathbf{y}_i,\ i \in \{1, \cdots, m\}, \end{aligned} \qquad (6)$$

where $\mathbf{z} = [\mathbf{z}_1^T, \cdots, \mathbf{z}_m^T]^T \in \mathbb{R}^{\sum_{i=1}^m n_i}$. It is easy to verify that the strong duality [9] holds for the optimization problem in Eq. (6). We can write down the Lagrange function of the above problem as follows:

$$\begin{aligned} \mathcal{L}(W, \mathbf{z}, \boldsymbol{\theta}) = &\sum_{i=1}^m \|\mathbf{z}_i\| + \lambda_1 \|W\|_{1,2} + \frac{\lambda_2}{2} \|W\|_F^2 \\ &+ \sum_{i=1}^m \boldsymbol{\theta}_i^T (X_i \mathbf{w}_i - \mathbf{y}_i - \mathbf{z}_i), \end{aligned} \qquad (7)$$

where $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T, \cdots, \boldsymbol{\theta}_m^T]^T \in \mathbb{R}^{\sum_{i=1}^m n_i}$ with $\boldsymbol{\theta}_i \in \mathbb{R}^{n_i}$ being the Lagrange multiplier corresponding to the constraint $\mathbf{z}_i = X_i \mathbf{w}_i - \mathbf{y}_i$. Minimizing $\mathcal{L}(W, \mathbf{z}, \boldsymbol{\theta})$ with respect to W and $\mathbf{z}$, we obtain the objective function of the dual problem of

Eq. (6) as follows:

$$\tilde{\mathcal{D}}(\boldsymbol{\theta}) = \min_W \left\{ \lambda_1 \|W\|_{1,2} + \frac{\lambda_2}{2}\|W\|_F^2 + \sum_{i=1}^m \boldsymbol{\theta}_i^T (X_i \mathbf{w}_i - \mathbf{y}_i) \right\} + \sum_{i=1}^m \min_{\mathbf{z}_i} \left\{ \|\mathbf{z}_i\| - \boldsymbol{\theta}_i^T \mathbf{z}_i \right\}, \tag{8}$$

where

$$\min_{\mathbf{z}_i} \left\{ \|\mathbf{z}_i\| - \boldsymbol{\theta}_i^T \mathbf{z}_i \right\} = \begin{cases} 0, & \text{if } \|\boldsymbol{\theta}_i\| \le 1, \\ -\infty, & \text{otherwise.} \end{cases} \tag{9}$$

Thus, we obtain the dual problem of Eq. (6) as follows:

$$\max_{\boldsymbol{\theta}} \mathcal{D}(\boldsymbol{\theta}), \quad s.t. \ \ \|\boldsymbol{\theta}_i\| \le 1, \ i \in \{1, \cdots, m\}, \tag{10}$$

where the objective function is

$$\mathcal{D}(\boldsymbol{\theta}) = \min_W \left\{ \lambda_1 \|W\|_{1,2} + \frac{\lambda_2}{2}\|W\|_F^2 + \sum_{i=1}^m \boldsymbol{\theta}_i^T (X_i \mathbf{w}_i - \mathbf{y}_i) \right\}. \tag{11}$$

We show that the optimization problem in Eq. (11) has nice properties which enable us to develop fast algorithms for solving the dual problem in Eq. (10).

THEOREM 1. *The optimization problem in Eq. (11) has a unique solution $W(\boldsymbol{\theta})$. Moreover, $\mathcal{D}(\boldsymbol{\theta})$ is continuously differentiable and L-Lipschitz continuous gradient. Specifically, the gradient of $\mathcal{D}(\boldsymbol{\theta})$ is*

$$\nabla \mathcal{D}(\boldsymbol{\theta}) = [(X_1 \mathbf{w}_1(\boldsymbol{\theta}) - \mathbf{y}_1)^T, \cdots, (X_m \mathbf{w}_m(\boldsymbol{\theta}) - \mathbf{y}_m)^T]^T, \tag{12}$$

where $\mathbf{w}_i(\boldsymbol{\theta})$ is the i-th column of $W(\boldsymbol{\theta})$. The Lipschitz constant of $\nabla \mathcal{D}(\boldsymbol{\theta})$ is

$$L = \frac{\max_{i \in \{1, \cdots, m\}} \sigma_{\max}^2(X_i)}{\lambda_2}, \tag{13}$$

where $\sigma_{\max}(X_i)$ is the largest singular value of X_i.

PROOF. Denote the objective function of the optimization problem in Eq. (11) as

$$f(W, \boldsymbol{\theta}) = \lambda_1 \|W\|_{1,2} + \frac{\lambda_2}{2}\|W\|_F^2 + \sum_{i=1}^m \boldsymbol{\theta}_i^T (X_i \mathbf{w}_i - \mathbf{y}_i). \tag{14}$$

Obviously, $f(W, \boldsymbol{\theta})$ is strongly convex and coercive with respect to W. Hence, the optimization problem in Eq. (11) has a unique solution $W(\boldsymbol{\theta})$.

To prove the remaining part of Theorem 1, we rewrite $f(W, \boldsymbol{\theta})$ in the following compact form:

$$f(W, \boldsymbol{\theta}) = \lambda_1 \|W\|_{1,2} + \frac{\lambda_2}{2}\|W\|_F^2 + \mathrm{Tr}(W^T U(\boldsymbol{\theta})) - \boldsymbol{\theta}^T \mathbf{y},$$

where $\mathrm{Tr}(\cdot)$ denotes the trace norm and

$$\begin{aligned} U(\boldsymbol{\theta}) &= [X_1^T \boldsymbol{\theta}_1, \cdots, X_m^T \boldsymbol{\theta}_m] \in \mathbb{R}^{d \times m}, \\ \boldsymbol{\theta} &= [\boldsymbol{\theta}_1^T, \cdots, \boldsymbol{\theta}_m^T]^T \in \mathbb{R}^{\sum_{i=1}^m n_i}, \\ \mathbf{y} &= [\mathbf{y}_1^T, \cdots, \mathbf{y}_m^T]^T \in \mathbb{R}^{\sum_{i=1}^m n_i}. \end{aligned}$$

Denote

$$g(W) = \lambda_1 \|W\|_{1,2} + \frac{\lambda_2}{2}\|W\|_F^2. \tag{15}$$

Then we have

$$\begin{aligned} \mathcal{D}(\boldsymbol{\theta}) &= \min_W \left\{ f(W, \boldsymbol{\theta}) = g(W) + \mathrm{Tr}(W^T U(\boldsymbol{\theta})) - \boldsymbol{\theta}^T \mathbf{y} \right\} \\ &= -\max_W \left\{ \mathrm{Tr}(-W^T U(\boldsymbol{\theta})) - g(W) \right\} - \boldsymbol{\theta}^T \mathbf{y} \\ &= -g^\star(-U(\boldsymbol{\theta})) - \boldsymbol{\theta}^T \mathbf{y}, \end{aligned} \tag{16}$$

where $g^\star(\cdot)$ is the conjugate function of $g(\cdot)$. Recalling the definition of $g(\cdot)$ in Eq. (15), we know that $g(\cdot)$ is a proper, lower semi-continuous and convex function. Thus, we have $g^{\star\star}(\cdot) = g(\cdot)$ and $g^\star(\cdot)$ is also a proper, lower semi-continuous and convex function. Moreover, $g(\cdot)$ is strongly convex with the parameter λ_2. Thus, by Proposition 12.60 in [31], we obtain that $g^\star(\cdot)$ is continuously differentiable and $\nabla g^\star(\cdot)$ is Lipschitz continuous with constant $1/\lambda_2$. Hence, $\mathcal{D}(\boldsymbol{\theta})$ is continuously differentiable. Noting that $W(\boldsymbol{\theta})$ is the minimizer of the optimization problem in Eq. (11), we have

$$W(\boldsymbol{\theta}) = \arg\max_W \left\{ \mathrm{Tr}(-W^T U(\boldsymbol{\theta})) - g(W) \right\},$$

and hence

$$-U(\boldsymbol{\theta}) \in \partial g(W(\boldsymbol{\theta})).$$

According to Corollary 23.5.1 in [30], we have $-U(\boldsymbol{\theta}) \in \partial g(W(\boldsymbol{\theta}))$ if and only if $W(\boldsymbol{\theta}) = \nabla g^\star(-U(\boldsymbol{\theta}))$, which together with Eq. (16) and the definition of $U(\boldsymbol{\theta})$ implies that

$$\begin{aligned} \nabla \mathcal{D}(\boldsymbol{\theta}) &= X \mathbf{s}(\boldsymbol{\theta}) - \mathbf{y} \\ &= [(X_1 \mathbf{w}_1(\boldsymbol{\theta}))^T, \cdots, (X_m \mathbf{w}_m(\boldsymbol{\theta}))^T]^T - \mathbf{y}, \end{aligned} \tag{17}$$

where

$$X = \begin{bmatrix} X_1 & & \\ & \ddots & \\ & & X_m \end{bmatrix}, \mathbf{s}(\boldsymbol{\theta}) = \begin{bmatrix} \nabla_1 g^\star(-U(\boldsymbol{\theta})) \\ \vdots \\ \nabla_m g^\star(-U(\boldsymbol{\theta})) \end{bmatrix}$$

with $\nabla_i g^\star(-U(\boldsymbol{\theta}))$ being the partial derivative of $g^\star(-U(\boldsymbol{\theta}))$ with respect to the i-th column of $-U(\boldsymbol{\theta})$. Thus, Eq. (12) immediately follows from Eq. (17). Moreover, for any $\boldsymbol{\theta}, \boldsymbol{\gamma} \in \mathbb{R}^{\sum_{i=1}^m n_i}$, we have

$$\|\nabla \mathcal{D}(\boldsymbol{\theta}) - \nabla \mathcal{D}(\boldsymbol{\gamma})\| = \|X(\mathbf{s}(\boldsymbol{\theta}) - \mathbf{s}(\boldsymbol{\gamma}))\|,$$

Recalling that $\nabla g^\star(\cdot)$ is Lipschitz continuous with constant $1/\lambda_2$, we have

$$\begin{aligned} & \|\nabla \mathcal{D}(\boldsymbol{\theta}) - \nabla \mathcal{D}(\boldsymbol{\gamma})\| \le \sigma_{\max}(X) \|\mathbf{s}(\boldsymbol{\theta}) - \mathbf{s}(\boldsymbol{\gamma})\| \\ & \le \frac{\sigma_{\max}(X)}{\lambda_2} \|U(\boldsymbol{\theta}) - U(\boldsymbol{\gamma})\|_F = \frac{\sigma_{\max}(X)}{\lambda_2} \|X^T(\boldsymbol{\theta} - \boldsymbol{\gamma})\| \\ & \le \frac{\sigma_{\max}^2(X)}{\lambda_2} \|\boldsymbol{\theta} - \boldsymbol{\gamma}\|, \end{aligned}$$

where $\sigma_{\max}(X)$ is the largest singular value of X. By noticing that X is a block diagonal matrix, we have

$$\|\nabla \mathcal{D}(\boldsymbol{\theta}) - \nabla \mathcal{D}(\boldsymbol{\gamma})\| \le \frac{\max_{i \in \{1, \cdots, m\}} \sigma_{\max}^2(X_i)}{\lambda_2} \|\boldsymbol{\theta} - \boldsymbol{\gamma}\|.$$

That is, $\nabla \mathcal{D}(\boldsymbol{\theta})$ is Lipschitz continuous with constant L defined in Eq. (13). □

REMARK 1. *The squared norm regularizer $\frac{\lambda_2}{2}\|W\|_F^2$ in Eq. (5) is critical for the establishment of Theorem 1 and hence is crucial to develop fast optimization algorithms on solving the dual problem in Eq. (10).*

Based on Theorem 1, we know that the objective function of the dual problem in Eq. (10) is smooth (Lipschitz continuous gradient) and the constraint is a piecewise sphere. Thus, we can use the framework of FISTA [4, 24, 12, 2] to solve the dual problem in Eq. (10). We present the pseudo codes in Algorithm 1. Notice that the dual problem in Eq. (10) is a *maximization* problem. Therefore, the gradient projection step in Eq. (20) and the line search criterion in Eq. (21) are modified accordingly.

In Algorithm 1, there are two critical steps. In the following, we show that both steps have closed-form solutions with low cost. The first critical step is how to efficiently compute the dual objective function value $\mathcal{D}(\boldsymbol{\theta})$ and the gradient $\nabla\mathcal{D}(\boldsymbol{\theta})$, which depends on how to efficiently obtain the optimal solution $W(\boldsymbol{\theta})$ for the problem in Eq. (11). Next, we show how to efficiently solve the optimization problem in Eq. (11). By exploring the special structure of Eq. (11), we have for all $j \in \{1, \cdots, d\}$:

$$\mathbf{w}^j(\boldsymbol{\theta}) = \arg\min_{\mathbf{w}^j} \left\{ \frac{\lambda_2}{2}\|\mathbf{w}^j\|^2 + \mathbf{u}^j(\boldsymbol{\theta})(\mathbf{w}^j)^T + \lambda_1\|\mathbf{w}^j\| \right\}, \tag{18}$$

where $\mathbf{w}^j(\boldsymbol{\theta}), \mathbf{w}^j$ and $\mathbf{u}^j(\boldsymbol{\theta})$ are the j-th row of $W(\boldsymbol{\theta}), W$ and $U(\boldsymbol{\theta}) = [X_1^T\boldsymbol{\theta}_1, \cdots, X_m^T\boldsymbol{\theta}_m]$, respectively. Eq. (18) has the following closed-form solution [14, 23]:

$$\begin{aligned}\mathbf{w}^j(\boldsymbol{\theta}) &= \arg\min_{\mathbf{w}^j} \left\{ \frac{\lambda_2}{2}\left\|\mathbf{w}^j + \frac{\mathbf{u}^j(\boldsymbol{\theta})}{\lambda_2}\right\|^2 + \lambda_1\|\mathbf{w}^j\| \right\} \\ &= \arg\min_{\mathbf{w}^j} \left\{ \frac{1}{2}\left\|\mathbf{w}^j + \frac{\mathbf{u}^j(\boldsymbol{\theta})}{\lambda_2}\right\|^2 + \frac{\lambda_1}{\lambda_2}\|\mathbf{w}^j\| \right\} \\ &= -\frac{1}{\lambda_2}\max\left(0, 1 - \frac{\lambda_1}{\|\mathbf{u}^j(\boldsymbol{\theta})\|}\right)\mathbf{u}^j(\boldsymbol{\theta}). \end{aligned} \tag{19}$$

The second critical step is how to efficiently solve the gradient projection subproblem. Due to the simplicity of the piecewise sphere constraint, the gradient projection subproblem has a closed-form solution in Eq. (20).

By Theorem 4.4 in [4], we have the following result.

Theorem 2. *(Theorem 4.4 [4]) Let $\{\boldsymbol{\theta}^k\}$ be the sequence generated by Algorithm 1 and $\boldsymbol{\theta}^\star$ be an optimal solution for the problem in Eq. (10). Then for all $k \geq 1$, we have*

$$\mathcal{D}(\boldsymbol{\theta}^\star) - \mathcal{D}(\boldsymbol{\theta}^k) \leq \frac{2\beta L\|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^\star\|^2}{(k+1)^2},$$

where $\boldsymbol{\theta}^0$ is the initial point; $\beta > 1$ is the factor used in the line search (β is defined in Algorithm 1); L is the Lipschitz constant defined in Eq. (13).

Based on Theorem 2, we establish a convergence rate for the sequence $\{W^k\}$ generated by Algorithm 1 in the following theorem:

Theorem 3. *Let $\{W^k\}$ and $\{\boldsymbol{\theta}^k\}$ be the sequences generated by Algorithm 1, and let $W^\star$ and $\boldsymbol{\theta}^\star$ be any optimal solutions for the problems in Eq. (5) and Eq. (10), respectively. Then for all $k \geq 1$, we have*

$$\|W^k - W^\star\|_F \leq 2\sqrt{\frac{\beta L}{\lambda_2}}\frac{\|\boldsymbol{\theta}^0 - \boldsymbol{\theta}^\star\|}{k+1}.$$

Algorithm 1: Accelerated Dual Ascent Algorithm

```
Input : $\boldsymbol{\theta}^0 \in \mathbb{R}^{\sum_{i=1}^m n_i}$, $\eta_0 > 0$, $\beta > 1$
Initialize $\boldsymbol{\theta}^1 \leftarrow \boldsymbol{\theta}^0$; $t_0 \leftarrow 1$; $t_1 \leftarrow 1$;
for $k = 1, 2, \cdots$ do
    $\alpha_k \leftarrow \frac{t_{k-1}-1}{t_k}$;
    $\boldsymbol{\nu}^k \leftarrow \boldsymbol{\theta}^k + \alpha_k(\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k-1})$;
    for $j = 0, 1, 2, \cdots$ do
        $\eta_k \leftarrow \beta^j \eta_{k-1}$;
        Compute $\boldsymbol{\theta}^k$ via gradient projection:
            $\boldsymbol{\theta}^k = \left[\frac{(\tilde{\boldsymbol{\nu}}_1^k)^T}{\max(1, \|\tilde{\boldsymbol{\nu}}_1^k\|)}, \cdots, \frac{(\tilde{\boldsymbol{\nu}}_m^k)^T}{\max(1, \|\tilde{\boldsymbol{\nu}}_m^k\|)}\right]^T$ with
            $\left[(\tilde{\boldsymbol{\nu}}_1^k)^T, \cdots, (\tilde{\boldsymbol{\nu}}_m^k)^T\right]^T = \tilde{\boldsymbol{\nu}}^k = \boldsymbol{\nu}^k + \frac{1}{\eta_k}\nabla\mathcal{D}(\boldsymbol{\nu}^k)$.   (20)
        if the following line search criterion
            $\mathcal{D}(\boldsymbol{\theta}^k) \geq \mathcal{D}(\boldsymbol{\nu}^k) + \nabla\mathcal{D}(\boldsymbol{\nu}^k)^T(\boldsymbol{\theta}^k - \boldsymbol{\nu}^k)$
            $\quad - \frac{\eta_k}{2}\|\boldsymbol{\theta}^k - \boldsymbol{\nu}^k\|^2$   (21)
        is satisfied then
            break;
        end
    end
    Compute $W^k \leftarrow W(\boldsymbol{\theta}^k)$ via Eq. (19);
    if some convergence criterion is satisfied then
        Let $\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^k$ and $W^\star \leftarrow W^k$;
        break;
    end
    $t_{k+1} \leftarrow \frac{1+\sqrt{1+4t_k^2}}{2}$;
end
Output: $\boldsymbol{\theta}^\star$, $W^\star$
```

Proof. We follow the proof of Theorem 4.1 in [5]. Denote

$$h(\mathbf{z}, \boldsymbol{\theta}) = \sum_{i=1}^m \left\{\|\mathbf{z}_i\| - \boldsymbol{\theta}_i^T\mathbf{z}_i\right\}, \tag{22}$$

$$\mathbf{z}^k \in \arg\min_{\mathbf{z}} h(\mathbf{z}, \boldsymbol{\theta}^k), \tag{23}$$

where $\{\boldsymbol{\theta}^k\}$ is the sequence generated by Algorithm 1. Considering Eqs. (7), (14), (22) together, we have

$$\mathcal{L}(W, \mathbf{z}, \boldsymbol{\theta}^k) = f(W, \boldsymbol{\theta}^k) + h(\mathbf{z}, \boldsymbol{\theta}^k). \tag{24}$$

From Algorithm 1, we have

$$W^k = W(\boldsymbol{\theta}^k) = \arg\min_W f(W, \boldsymbol{\theta}^k). \tag{25}$$

By the optimality condition of Eq. (25), we have

$$O \in \partial_{W^k} f(W^k, \boldsymbol{\theta}^k), \tag{26}$$

where O denotes the zero matrix and $\partial_{W^k} f(W^k, \boldsymbol{\theta}^k)$ denotes the sub-differential of $f(W, \boldsymbol{\theta}^k)$ with respect to W at the point $W = W^k$. Recalling that $f(W, \boldsymbol{\theta}^k)$ is strongly convex with respect to W with the parameter λ_2, which together with Eq. (26) implies that for all $W \in \mathbb{R}^{d\times m}$

$$f(W, \boldsymbol{\theta}^k) - f(W^k, \boldsymbol{\theta}^k) \geq \frac{\lambda_2}{2}\|W - W^k\|_F^2. \tag{27}$$

By Eq. (23), we have for all $\mathbf{z} \in \mathbb{R}^{\sum_{i=1}^m n_i}$

$$h(\mathbf{z}, \boldsymbol{\theta}^k) - h(\mathbf{z}^k, \boldsymbol{\theta}^k) \geq 0. \tag{28}$$

Adding Eqs. (27), (28) together and considering Eq. (24), we have for all $W \in \mathbb{R}^{d\times m}$ and $\mathbf{z} \in \mathbb{R}^{\sum_{i=1}^m n_i}$:

$$\mathcal{L}(W, \mathbf{z}, \boldsymbol{\theta}^k) - \mathcal{L}(W^k, \mathbf{z}^k, \boldsymbol{\theta}^k) \geq \frac{\lambda_2}{2}\|W - W^k\|_F^2. \tag{29}$$

From Eqs. (11), (25), we know that

$$\mathcal{D}(\boldsymbol{\theta}^k) = f(W^k, \boldsymbol{\theta}^k). \tag{30}$$

Recalling that the sequence $\{\boldsymbol{\theta}^k\} = \{[(\boldsymbol{\theta}_1^k)^T, \cdots, (\boldsymbol{\theta}_m^k)^T]^T\}$ satisfies the constraint in Eq. (10) and considering Eqs. (9), (23), we have for all $k \geq 1$:

$$h(\mathbf{z}^k, \boldsymbol{\theta}^k) = 0,$$

which together with Eqs. (24), (30) implies that

$$\mathcal{D}(\boldsymbol{\theta}^k) = \mathcal{L}(W^k, \mathbf{z}^k, \boldsymbol{\theta}^k). \tag{31}$$

Denote by $(W^\star, \mathbf{z}^\star)$ as an optimal solution for the optimization problem in Eq. (6). Then by the equality constraint in Eq. (6), we have for all $i \in \{1, \cdots, m\}$

$$\mathbf{z}^\star = X_i \mathbf{w}_i^\star - \mathbf{y}_i,$$

which together with Eq. (7) implies that

$$\mathcal{L}(W^\star, \mathbf{z}^\star, \boldsymbol{\theta}^k) = \sum_{i=1}^m \|\mathbf{z}_i^\star\| + \lambda_1\|W^\star\|_{1,2} + \frac{\lambda_2}{2}\|W^\star\|_F^2. \tag{32}$$

Noticing that Eq. (32) attains the minimum objective function value in Eq. (6) and strong duality holds for the optimization problem in Eq. (6), we have

$$\mathcal{D}(\boldsymbol{\theta}^\star) = \mathcal{L}(W^\star, \mathbf{z}^\star, \boldsymbol{\theta}^k), \tag{33}$$

where $\boldsymbol{\theta}^\star$ is an optimal solution of the dual problem in Eq. (10). Substituting $W = W^\star, \mathbf{z} = \mathbf{z}^\star$ into Eq. (29) and considering Eqs. (31), (33), we obtain

$$\mathcal{D}(\boldsymbol{\theta}^\star) - \mathcal{D}(\boldsymbol{\theta}^k) \geq \frac{\lambda_2}{2}\|W^\star - W^k\|_F^2,$$

which together with Theorem 2 immediately implies that the inequality in Theorem 3 holds. □

Remark 2. *In additional to FISTA [4], we can use many other optimization algorithms to solve the dual problem in Eq. (10). For example, the SpaRSA framework [32] can be adopted to efficiently solve the dual problem. The theoretical convergence rate of SpaRSA is no better than that of FISTA. However, due to the utilization of the Barzilai-Borwein (BB) rule [3, 32], the empirical convergence performance of SpaRSA is much better than that of FISTA for solving the dual problem in Eq. (10) [see Section 4.2].*

3.2 Discussion

Notice that the objective function in Eq. (5) consists of a differentiable term $\frac{\lambda_2}{2}\|W\|_F^2$ and a non-differentiable term $\sum_{i=1}^m \|X_i\mathbf{w}_i - \mathbf{y}_i\| + \lambda_1\|W\|_{1,2}$. Theoretically, we can use the framework of FISTA [4] to directly solve the optimization problem in Eq. (5), achieving a convergence rate of $O(1/k^2)$. If we additionally consider the strong convexity of the objective function in Eq. (5), the proximal gradient method (PGM) can achieve a geometrically linear convergence rate. However, both FISTA and PGM are not practical for directly solving the optimization problem in Eq. (5). Specifically, *at each iteration* of both FISTA and PGM, we need to solve the following proximal operator problem:

$$\min_W \left\{\frac{1}{2}\|W - B\|_F^2 + \rho\left(\sum_{i=1}^m \|X_i\mathbf{w}_i - \mathbf{y}_i\| + \lambda_1\|W\|_{1,2}\right)\right\},$$

where $B \in \mathbb{R}^{d\times m}$ and $\rho > 0$ are both constant with respect to W. Obviously, solving the above proximal operator problem is as difficult as solving the original optimization problem in Eq. (5).

We can also use a similar primal smoothed optimization algorithm (more details are provided in the Appendix A) proposed in [22] to solve the primal problem in Eq. (5) directly. One problem is that the smoothing parameter is quite challenging to tune while this parameter is crucial to the convergence performance of the SPG algorithm.

Another approach which can solve the optimization problem in Eq. (5) is the Alternating Direction Method of Multipliers (ADMM) method [8] (more details are provided in the Appendix B). One problem is that, at each step, we need to solve a Lasso problem which does *not* admit a closed-form solution. Moreover, the penalty parameter is very hard to tune while this parameter is critical for the efficiency of the ADMM method.

A very straightforward approach to solve the optimization problem in Eq. (5) is the sub-gradient method [10]. In the non-smooth case, the objective function value sequence $\{l(W^k)\}$ generated by the sub-gradient method converges to the optimal value $l^\star$ at a rate of $O(1/\sqrt{k})$ under certain conditions [10], where $l(W)$ denotes the objective function in Eq. (5). However, this convergence rate is established without exploiting the strong convexity of the objective function $l(W)$. By considering the strong convexity of $l(W)$, Beck and Teboulle [5] show that the convergence rate of the objective function value has been improved from $O(1/\sqrt{k})$ to $O(\ln(k)/k)$ using the sub-gradient method. However, the sub-gradient method still converges slowly because the sub-gradient method is a very general method for solving non-smooth optimization problems without exploring the special structure of Eq. (5) and the step size at each step is shrinking to zero (or the step size is a very small constant).

The cutting plane method [7] and the level set method [20] are another two approaches which can solve the optimization problem in Eq. (5). They can be viewed as improved methods of the sub-gradient method. Similar to the sub-gradient method, the cutting plane and the level set methods are general methods for solving non-smooth optimization problems and do not exploit the the special structure of Eq. (5).

4. EXPERIMENTS

In this section, we first evaluate the prediction performance of the calibrated multi-task feature algorithms with and without the squared norm regularizer. Then, we present computational efficiency studies for our proposed algorithms. Matlab codes are included at MALSAR package [35].

4.1 Prediction Performance

We include the following algorithms for comparison: (1) **Calibration**: the calibrated multi-task feature learning formulation in Eq. (5) by setting $\lambda_2 = 0$; (2) **Calibration-**

Table 1: The averaged MSE (standard deviation) over 10 random splittings of training and test samples. Training ratio refers to the ratio of training samples to the total samples.

data sets	training ratio	Calibration	Calibration-L2	LeastSquares	LeastSquares-L2
Synthetic	60%	*11.73(0.37)*	11.80(0.42)	12.03(0.29)	12.03(0.29)
	70%	9.19(0.43)	*9.18(0.40)*	10.09(0.49)	10.09(0.49)
	80%	*7.24(0.42)*	7.25(0.42)	8.28(0.65)	8.28(0.65)
ADNI ADAS-Cog	60%	*37.27(2.31)*	37.45(2.24)	38.92(2.75)	39.02(2.81)
	70%	*32.95(1.88)*	33.57(2.21)	34.19(1.99)	34.15(1.98)
	80%	27.11(2.12)	*26.73(1.72)*	27.38(1.82)	27.35(1.75)
ADNI MMSE	60%	21.93(0.87)	*21.80(0.97)*	23.63(0.71)	23.69(0.71)
	70%	18.75(0.24)	*18.52(0.39)*	20.26(0.86)	20.30(0.87)
	80%	15.20(1.00)	*15.08(1.12)*	16.53(0.76)	16.54(0.78)

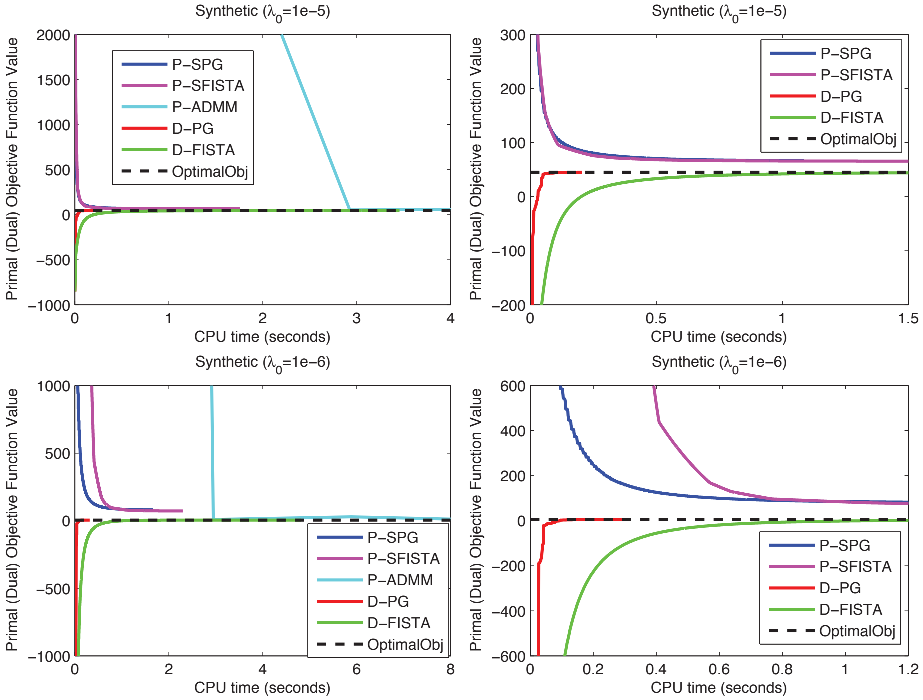

Figure 1: The primal objective function value (P-SPG, P-SFISTA and P-ADMM) and the dual objective function value (D-PG and D-FISTA) vs. CPU time (seconds) plots on the Synthetic data set. The right figures are magnified views of the left figures. The legend "PrimalObj" denotes the optimal primal objective function value in Eq. (5) and regularized parameters are set as $\lambda_1 = \lambda_2 = \lambda_0 \sum_{i=1}^{m} n_i$, where n_i is the number of samples from the i-th task.

L2: the calibrated multi-task feature learning formulation in Eq. (5) by setting $\lambda_2 \neq 0$; (3) **LeastSquares**: the non-calibrated multi-task feature learning formulation in Eq. (1) by setting $r(W) = \|W\|_{1,2}$; (4) **LeastSquares-L2**: the non-calibrated multi-task feature learning formulation in Eq. (1) by setting $r(W) = \|W\|_{1,2}$ and adding a squared norm regularizer $\frac{\lambda_2}{2}\|W\|_F^2$. We conduct experiments on both synthetic and real-world data sets which are described as follows:

Synthetic Data: We adopt the similar procedure in [22] to generate the synthetic data as follows: we set the number of tasks as $m = 10$ and each task has $n_i = 100$ samples which have $d = 200$ features (i.e., the dimensionality is $d = 200$); each row of the data matrix $X_i \in \mathbb{R}^{n_i \times d}$ of the i-th task is independently sampled from the multivariate normal distribution $N(\mathbf{0}, \Sigma)$ with $\sigma_{jj} = 1$ and $\sigma_{jk} = 0.5$ for all $j \neq k$, and it is normalized such that the length of each column of X_i is 1; each entry of the underlying ground truth matrix $W \in \mathbb{R}^{d \times m}$ is independently sampled from the uniform distribution $U(-10, 10)$ and then we randomly set 95% rows of W as zero vectors; each entry of the noise vector $\boldsymbol{\delta}_i \in \mathbb{R}^{n_i}$ is independently sampled from the normal distribution $N(0, \sigma_i^2)$, where $\sigma_i = 2^{-(i-1)/4}$ for $i = 1, \cdots, m$; the response vector $\mathbf{y}_i \in \mathbb{R}^{n_i}$ of the i-th task is computed by $\mathbf{y}_i = X_i \mathbf{w}_i + \sigma_{\max} \boldsymbol{\delta}_i$, where $\sigma_{\max} = 2\sqrt{2}$.

Real-World Data: We conduct experiments on two real-world data sets which are summarized as follows: (I) The Alzheimer's Disease Neuroimaging Initiative (ADNI)[1] is a longitudinal study aiming at identifying important neuroimaging biomarkers that are predictive of the progression of the Alzheimer's disease and building predictive models for prognosis of the disease. In our experiments, we use 310 MRI features from 648 patients to predict the MMSE value, which is a cognitive score that indicates the cognitive functionality of the patients. According to the medical diagnosis, there

[1] www.loni.ucla.edu/ADNI/

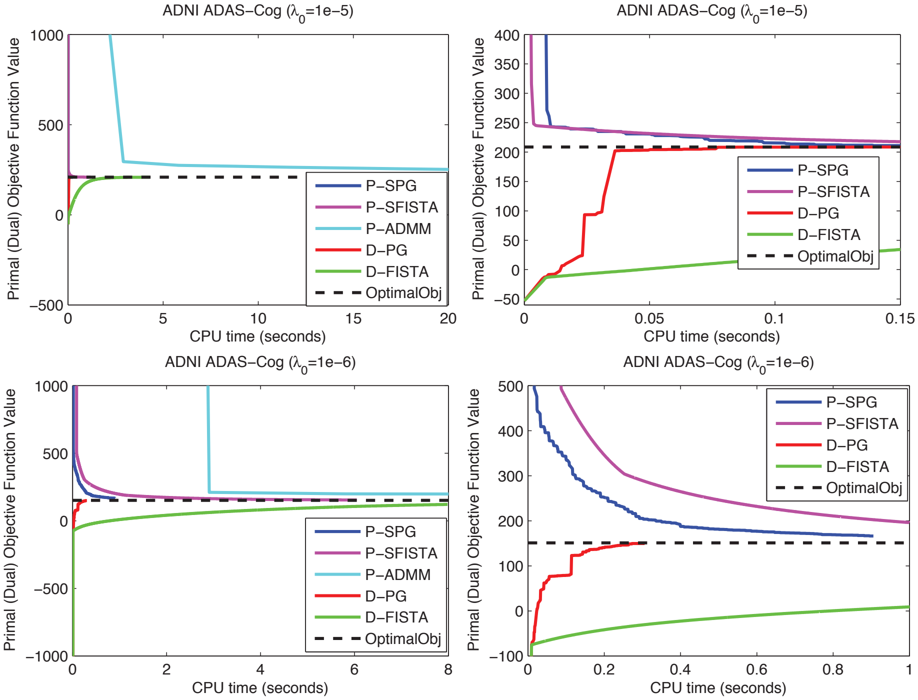

Figure 2: The primal objective function value (P-SPG, P-SFISTA and P-ADMM) and the dual objective function value (D-PG and D-FISTA) vs. CPU time (seconds) plots on the ADNI ADAS-Cog data set. The right figures are magnified views of the left figures. The legend "PrimalObj" denotes the optimal primal objective function value in Eq. (5) and regularized parameters are set as $\lambda_1 = \lambda_2 = \lambda_0 \sum_{i=1}^m n_i$, where n_i is the number of samples from the i-th task.

are 191 normal control (NC), 319 mild cognitive impairment (MCI), and 138 patients with probably Alzheimer's disease (AD) [please refer to [36, 37] for more details about the data]. We thus construct three tasks, and in each medical group we build one regression model to predict the corresponding MMSE scores. (II) We also use the ADNI data set and build multi-task regression models for predicting the ADAS-Cog score, which is another important cognitive score. We construct the task in the same way as in (I).

In the prediction performance experiments, we terminate the comparative algorithms when the relative change of the two consecutive objective function values is less than 10^{-6} or the number of iterations exceeds 10000. We randomly split the samples from each task into training and test samples with different training ratios (60%, 70% and 80%). All parameters of the comparative algorithms are tuned via 5-fold cross validation. For each training ratio, we report the averaged mean squared error (MSE) and the standard deviation over 10 random splittings of training and test samples as shown in Table 1. From these results, we have the following observations: (a) The calibrated multi-task feature learning algorithms (Calibration and Calibration-L2) outperform the non-calibrated ones (LeastSquares and LeastSquares-L2), which shows the superior prediction performance of the calibrated algorithms over the non-calibrated ones. (b) Calibration and Calibration-L2 achieve very similar prediction performance, which demonstrates that the proposed variant keeps the superior prediction performance of the calibrated multi-task feature learning over the non-calibrated one. In the next subsection, we will show that solving the smooth dual problem using a proper optimization algorithm is much more efficient than solving the primal problem directly.

4.2 Computational Efficiency Studies

We study the computational efficiency of solving the optimization problem in Eq. (5) by comparing the following algorithms: (1) **D-PG**: solve the primal optimization problem in Eq. (5) via using SpaRSA [32] to solve the dual optimization problem in Eq. (10); (2) **D-FISTA**: solve the primal optimization problem in Eq. (5) via using FISTA [4] to solve the dual optimization problem in Eq. (10); (3) **P-SPG**: solve the primal optimization problem in Eq. (5) via using SpaRSA [32] to solve the smoothed optimization problem in Eq. (37); (4) **P-SFISTA**: solve the primal optimization problem in Eq. (5) via using FISTA [4] to solve the smoothed optimization problem in Eq. (37); (5) **P-ADMM**: solve the primal optimization problem in Eq. (5) by using ADMM [8]. We conduct the experiments on the same data sets (both synthetic and real-world) described in the last subsection. We initialize D-PG and D-FISTA with $\boldsymbol{\theta}^0$ whose entries are independently sampled from the standard normal distribution, and initialize P-SPG, P-FISTA and P-ADMM with W^0 which is computed by Eq. (19) using $\boldsymbol{\theta}^0$. We terminate the comparative algorithms when the relative change of the two consecutive objective function values[2] is less than 10^{-6} or the number of iterations exceeds 10000. All algorithms are implemented in Matlab and executed on an Intel Core i7-3770 CPU (@3.4GHz) with 32GB memory.

[2] The objective function value indicates the dual objective function value in Eq. (10) and the primal objective function value in Eq. (5) for dual optimization algorithms (D-PG and D-FISTA) and for primal optimization algorithms (P-SPG, P-FISTA and P-ADMM), respectively.

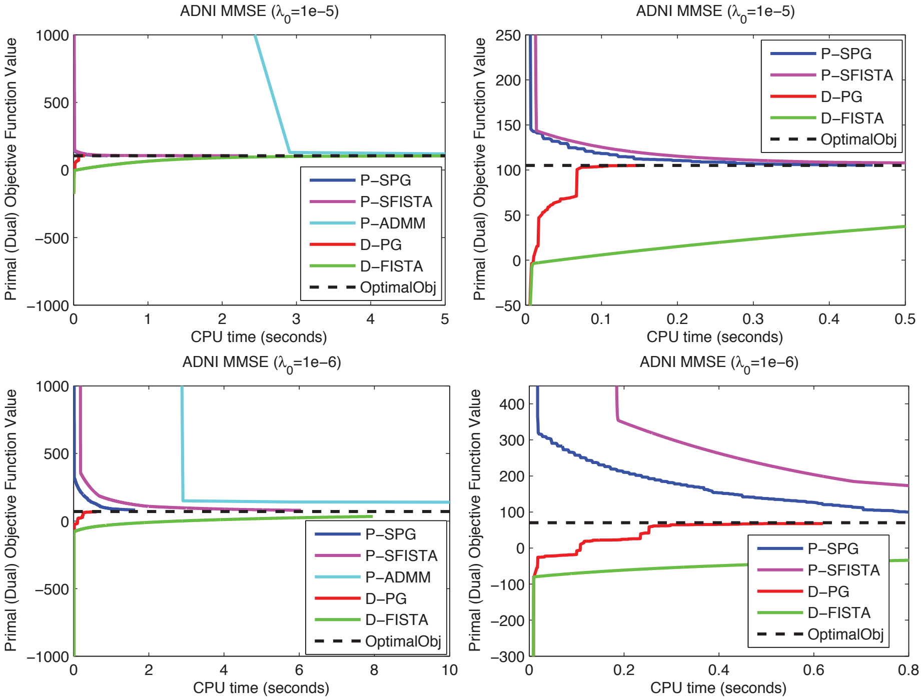

Figure 3: The primal objective function value (P-SPG, P-SFISTA and P-ADMM) and the dual objective function value (D-PG and D-FISTA) vs. CPU time (seconds) plots on the ADNI MMSE data set. The right figures are magnified views of the left figures. The legend "PrimalObj" denotes the optimal primal objective function value in Eq. (5) and regularized parameters are set as $\lambda_1 = \lambda_2 = \lambda_0 \sum_{i=1}^{m} n_i$, where n_i is the number of samples from the i-th task.

To show detailed convergence behaviors of the comparative algorithms and compare their computational efficiency, we report the primal objective function value (for P-SPG, P-FISTA and P-ADMM) and the dual objective function value (for D-PG and D-FISTA) vs. CPU time plots as shown in Figure 1, Figure 2 and Figure 3. To better compare the efficiency of dual and primal optimization algorithms, in each figure, we also draw a horizontal line indicating the optimal primal objective function value in Eq. (5). From these results, we have the following observations: (a) D-PG and D-FISTA increase the dual objective function value which finally converges to the optimal primal objective function value. This validates the fact that the strong duality holds for the optimization problem in Eq. (5). (b) D-PG is the most efficient among all algorithms on all data sets. Specifically, D-PG always quickly increases the dual objective function and rapidly terminates by approaching the optimal primal objective function value. (c) D-PG converges faster than D-FISTA and D-SPG converges faster than P-SFISTA, which demonstrates that SpaRSA indeed has very good empirical convergence performance. Although there is no rigorous proof that D-PG is the most efficient, empirical studies demonstrate that using SpaRSA to solve the dual problem is much more efficient than that by solving the primal problem directly. This may be due to the nice properties of the dual problem and the empirically fast convergence of SpaRSA.

5. CONCLUSIONS

In this paper, we study a variant of the calibrated multi-task feature learning by adding a squared norm regularizer. We derive a smooth dual optimization problem with a piecewise sphere constraint, which enables us to develop fast dual optimization algorithms. We also provide a detailed convergence analysis for the proposed dual optimization algorithm. Empirical studies demonstrate that, the dual optimization algorithm quickly converges and it is much more efficient than the primal optimization algorithm. Moreover, the calibrated multi-task feature learning algorithms with and without the squared norm regularizer achieve similar prediction performance and both outperform the non-calibrated ones. In our future work, we will analyze statistical properties of the proposed formulation and apply it to other applications such as Drosophila image annotation [17].

Acknowledgements

This work is supported in part by China 973 Fundamental R&D Program (No.2014CB340304), NIH (R01 LM010730), and NSF (IIS-0953662, CCF-1025177).

6. REFERENCES

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

[2] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2011.

[3] J. Barzilai and J. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[5] A. Beck and M. Teboulle. A fast dual proximal gradient algorithm for convex minimization and applications. *Operations Research Letters*, 42(1):1–6, 2014.

[6] A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

[7] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[9] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[10] S. Boyd, L. Xiao, and A. Mutapcic. Subgradient methods. *Lecture notes of EE392o, Stanford University*, 2003.

[11] F. Bunea, J. Lederer, and Y. She. The group square-root lasso: Theoretical properties and fast algorithms. *arXiv preprint arXiv:1302.0261*, 2013.

[12] P. Gong, K. Gai, and C. Zhang. Efficient euclidean projections via piecewise root finding and its application in gradient projection. *Neurocomputing*, 74(17):2754–2766, 2011.

[13] P. Gong, J. Ye, and C. Zhang. Multi-stage multi-task feature learning. In *NIPS*, pages 1997–2005, 2012.

[14] P. Gong, J. Ye, and C. Zhang. Robust multi-task feature learning. In *SIGKDD*, pages 895–903, 2012.

[15] P. Gong, J. Ye, and C. Zhang. Multi-stage multi-task feature learning. *Journal of Machine Learning Research*, 14(1):2979–3010, 2013.

[16] A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. A dirty model for multi-task learning. *NIPS*, 2010.

[17] 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.

[18] S. Kim and E. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML*, 2009.

[19] M. Kolar, J. Lafferty, and L. Wasserman. Union support recovery in multi-task learning. *Journal of Machine Learning Research*, 12:2415–2435, 2011.

[20] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69(1):111–147, 1995.

[21] H. Liu, M. Palatucci, and J. Zhang. Blockwise coordinate descent procedures for the multi-task lasso with applications to neural semantic basis discovery. In *ICML*, pages 649–656, 2009.

[22] H. Liu, L. Wang, and T. Zhao. Multivariate regression with calibration. *arXiv preprint arXiv:1305.2238*, 2013.

[23] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *UAI*, 2009.

[24] J. Liu, S. Ji, and J. Ye. Slep: Sparse learning with efficient projections. *Arizona State University*, 2009.

[25] K. Lounici, M. Pontil, A. Tsybakov, and S. Van De Geer. Taking advantage of sparsity in multi-task learning. In *COLT*, 2009.

[26] K. Lounici, M. Pontil, S. Van De Geer, and A. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics*, 39(4):2164–2204, 2011.

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

[28] F. Nie, H. Huang, X. Cai, and C. Ding. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *NIPS*, pages 1813–1821, 2010.

[29] G. Obozinski, B. Taskar, and M. Jordan. Multi-task feature selection. *Statistics Department, UC Berkeley, Tech. Rep*, 2006.

[30] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1997.

[31] R. T. Rockafellar and R. Wets. *Variational Analysis*. Springer, 1998.

[32] S. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

[33] X. Yang, S. Kim, and E. Xing. Heterogeneous multitask learning with joint sparsity constraints. In *NIPS*, 2009.

[34] Y. Zhang, D. Yeung, and Q. Xu. Probabilistic multi-task feature selection. In *NIPS*, 2010.

[35] J. Zhou, J. Chen, and J. Ye. *MALSAR: Multi-tAsk Learning via StructurAl Regularization*. Arizona State University, 2011.

[36] J. Zhou, J. Liu, V. Narayan, and J. Ye. Modeling disease progression via fused sparse group lasso. In *SIGKDD*, pages 1095–1103, 2012.

[37] J. Zhou, L. Yuan, J. Liu, and J. Ye. A multi-task learning formulation for predicting disease progression. In *SIGKDD*, pages 814–822, 2011.

[38] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

APPENDIX

A. Primal Smoothed Optimization Algorithm

We use the smoothing technique [27] to derive a primal smoothed optimization algorithm to solve Eq. (5). Based on the definition of the dual norm, we have

$$\sum_{i=1}^{m} \|X_i\mathbf{w}_i - \mathbf{y}_i\| = \sum_{i=1}^{m} \max_{\|\mathbf{v}_i\|\le 1} \mathbf{v}_i^T(X_i\mathbf{w}_i - \mathbf{y}_i). \tag{34}$$

Define a function as

$$\sum_{i=1}^{m} \|X_i\mathbf{w}_i - \mathbf{y}_i\|_\mu = \sum_{i=1}^{m} \max_{\|\mathbf{v}_i\|\le 1} \left\{ \mathbf{v}_i^T(X_i\mathbf{w}_i - \mathbf{y}_i) - \frac{\mu}{2}\|\mathbf{v}_i\|^2 \right\}, \tag{35}$$

where $\mu > 0$ is a smoothing parameter. We show in the following proposition that $\sum_{i=1}^{m} \|X_i\mathbf{w}_i - \mathbf{y}_i\|_\mu$ is smooth.

Theorem 4. *The optimization problem in Eq. (35) has a unique solution in the following closed-form:*

$$\mathbf{v}_i(\mathbf{w}_i) = \frac{X_i\mathbf{w}_i - \mathbf{y}_i}{\max(\mu, \|X_i\mathbf{w}_i - \mathbf{y}_i\|)}.$$

Moreover, $\sum_{i=1}^{m} \|X_i\mathbf{w}_i - \mathbf{y}_i\|_\mu$ is continuously differentiable and Lipschitz continuous gradient with respect to W. Specifically, the gradient $G^\mu(W)$ of $\sum_{i=1}^{m} \|X_i\mathbf{w}_i - \mathbf{y}_i\|_\mu$ is

$$G^\mu(W) = [X_1^T\mathbf{v}_1(\mathbf{w}_1), \cdots, X_m^T\mathbf{v}_m(\mathbf{w}_m)],$$

and the Lipschitz constant of $G^\mu(W)$ is

$$\tilde{L} = \frac{\max_{i\in\{1,\cdots,m\}} \sigma_{\max}^2(X_i)}{\mu}. \tag{36}$$

Based on Theorem 4 (the proof of Theorem 4 is very similar to that of Theorem 1 and is thus omitted here), we know that $\sum_{i=1}^{m} \|X_i\mathbf{w}_i - \mathbf{y}_i\|_\mu$ is a smoothed surrogate of the nonsmooth term $\sum_{i=1}^{m} \|X_i\mathbf{w}_i - \mathbf{y}_i\|$. Thus, we consider solving the following smoothed version of the optimization problem in Eq. (5):

$$\widehat{W} = \arg\min_{W\in\mathbb{R}^{d\times m}} \left\{ \sum_{i=1}^{m} \|X_i\mathbf{w}_i - \mathbf{y}_i\|_\mu + \frac{\lambda_2}{2}\|W\|_F^2 + \lambda_1\|W\|_{1,2} \right\}. \tag{37}$$

The primal smoothed algorithm in fact solves the optimization problem in Eq. (5) via solving the smoothed optimization problem in Eq. (37). To be more specific, we present a primal smoothed optimization algorithm called accelerated primal smoothed algorithm given in Algorithm 2, in which FISTA [4] is used to solve the smoothed optimization problem in Eq. (37) and some notations are defined as follows:

$$l(W)_\mu = \sum_{i=1}^{m} \|X_i\mathbf{w}_i - \mathbf{y}_i\|_\mu + \frac{\lambda_2}{2}\|W\|_F^2 + \lambda_1\|W\|_{1,2},$$

$$s(W)_\mu = \sum_{i=1}^{m} \|X_i\mathbf{w}_i - \mathbf{y}_i\|_\mu + \frac{\lambda_2}{2}\|W\|_F^2.$$

The convergence analysis of Algorithm 2 is similar to Theorem 2.2 in [22] and is thus omitted here.

Algorithm 2: Accelerated Primal Smoothed Algorithm

```
Input  : $W^0 \in \mathbb{R}^{d\times m}$, $\eta_0 > 0$, $\beta > 1$;
Initialize $W^1 \leftarrow W^0$; $t_0 \leftarrow 1$; $t_1 \leftarrow 1$;
for $k = 1, 2, \cdots$ do
    $\alpha_k \leftarrow \frac{t_{k-1}-1}{t_k}$;
    $V^k = W^k + \alpha_k(W^k - W^{k-1})$;
    for $j = 0, 1, 2, \cdots$ do
        $\eta_k \leftarrow \beta^j\eta_{k-1}$; $\widetilde{V}^k \leftarrow V^k - \frac{1}{\eta_k}\nabla s(V^k)_\mu$;
        Compute $W^k$ via proximal gradient:
            $W^k = \arg\min_W \left\{ \frac{\eta_k}{2}\|W - \widetilde{V}^k\|_F^2 + \lambda_1\|W\|_{1,2} \right\}$ ;
        if the following line search criterion
            $l(W^k)_\mu \le l(W^k)_\mu + tr(\nabla l(V^k)_\mu^T(W^k - V^k)) + \frac{\eta_k}{2}\|W^k - V^k\|_F^2$
        is satisfied then
            break;
        end
    end
    if some convergence criterion is satisfied then
        Let $W^\star \leftarrow W^k$;
        break;
    end
    $t_{k+1} \leftarrow \frac{1+\sqrt{1+4t_k^2}}{2}$;
end
Output: $W^\star$
```

Remark 3. *Similar to the dual optimization algorithm, we can also use SpaRSA [32] to solve the smoothed optimization problem in Eq. (37). Due to the use of the Barzilai-Borwein (BB) rule [3, 32], SpaRSA achieves very good empirical convergence performance.*

B. Alternating Direction Method of Multipliers (ADMM)

We know that the optimization problem in Eq. (5) is equivalent to the constrained optimization problem in Eq. (6) and we can write down the augmented Lagrange function of Eq. (6) as follows:

$$\mathcal{L}_{\mathcal{A}}(W, \mathbf{z}, \boldsymbol{\theta}, \rho) = \sum_{i=1}^{m}\|\mathbf{z}_i\| + \lambda_1\|W\|_{1,2} + \frac{\lambda_2}{2}\|W\|_F^2, \\ + \sum_{i=1}^{m}\left\{\boldsymbol{\theta}_i^T(\mathbf{y}_i - \mathbf{z}_i - X_i\mathbf{w}_i) + \frac{\rho}{2}\|\mathbf{y}_i - \mathbf{z}_i - X_i\mathbf{w}_i\|^2\right\},$$

where $\mathbf{z} = [\mathbf{z}_1^T, \cdots, \mathbf{z}_m^T]^T \in \mathbb{R}^{\sum_{i=1}^m n_i}$; $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T, \cdots, \boldsymbol{\theta}_m^T]^T \in \mathbb{R}^{\sum_{i=1}^m n_i}$ with $\boldsymbol{\theta}_i \in \mathbb{R}^{n_i}$ being the Lagrange multiplier corresponding to the constraint $\mathbf{z}_i = \mathbf{y}_i - X_i\mathbf{w}_i$; $\rho > 0$ is a penalty parameter.

The ADMM solves the optimization problem in Eq. (5) by alternatively updating the variables as follows:
(1) Update W:

$$\begin{aligned} W^k = \arg\min_W &\left\{\lambda_1\|W\|_{1,2} + \frac{\lambda_2}{2}\|W\|_F^2 \right. \\ &+ \sum_{i=1}^{m}\Big[(\boldsymbol{\theta}_i^{k-1})^T(\mathbf{y}_i - \mathbf{z}_i^{k-1} - X_i\mathbf{w}_i) \\ &\left. + \frac{\rho}{2}\|\mathbf{y}_i - \mathbf{z}_i^{k-1} - X_i\mathbf{w}_i\|^2\Big]\right\} \\ = \arg\min_W &\left\{\lambda_1\|W\|_{1,2} + \frac{\lambda_2}{2}\|W\|_F^2 \right. \\ &\left. + \frac{\rho}{2}\sum_{i=1}^{m}\left\|\frac{\boldsymbol{\theta}_{\boldsymbol{i}}^{k-1}}{\rho} - \mathbf{z}_i^{k-1} + \mathbf{y}_i - X_i\mathbf{w}_i\right\|^2\right\} \end{aligned}$$

We can use FISTA [4] or SpaSRA [32] to efficiently solve the above problem, where the gradient of the smooth part of the objective function is given by:

$$\nabla_W f_s(W) = \lambda_2 W + [\mathbf{t}_1 \ldots \mathbf{t}_m] \text{ with}$$

$$\mathbf{t}_i = -\rho X_i^T\left(\frac{\boldsymbol{\theta}_{\boldsymbol{i}}^{k-1}}{\rho} - \mathbf{z}_i^{k-1} + \mathbf{y}_i - X_i\mathbf{w}_i\right).$$

(2) Update $\mathbf{z}$:

$$\begin{aligned} \mathbf{z}^k = \arg\min_{\mathbf{z}} &\left\{\sum_{i=1}^{m}\|\mathbf{z}_i\| + \sum_{i=1}^{m}\Big[(\boldsymbol{\theta}_i^{k-1})^T(\mathbf{y}_i - \mathbf{z}_i - X_i\mathbf{w}_i^k)\right. \\ &\left. + \frac{\rho}{2}\|\mathbf{y}_i - \mathbf{z}_i - X_i\mathbf{w}_i^k\|^2\Big]\right\} \\ = \arg\min_{\mathbf{z}} &\sum_{i=1}^{m}\left\{\|\mathbf{z}_i\| + \frac{\rho}{2}\left\|\frac{\boldsymbol{\theta}_{\boldsymbol{i}}^{k-1}}{\rho} + \mathbf{y}_i - \mathbf{z}_i - X_i\mathbf{w}_i^k\right\|^2\right\}. \end{aligned}$$

For the above problem, the optimal solution is given by the following closed-form:

$$\begin{aligned} \mathbf{z}_i^k &= \arg\min_{\mathbf{z}_i}\left\{\frac{1}{\rho}\|\mathbf{z}_i\| + \frac{1}{2}\left\|\mathbf{z}_i - \left(\frac{\boldsymbol{\theta}_i^{k-1}}{\rho} + \mathbf{y}_i - X_i\mathbf{w}_i^k\right)\right\|^2\right\} \\ &= \max\left(0, 1 - \frac{1}{\rho\|\mathbf{v}_i\|}\right)\mathbf{v}_i, \end{aligned}$$

where $\mathbf{v}_i = \dfrac{\boldsymbol{\theta}_i^{k-1}}{\rho} + \mathbf{y}_i - X_i\mathbf{w}_i^k$.

(3) Update $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_i^k = \boldsymbol{\theta}_i^{k-1} + \rho\left(\mathbf{y}_i - \mathbf{z}_i^k - X_i\mathbf{w}_i^k\right).$$