Robust Multi-Task Feature Learning

Pinghua Gong[†], Jieping Ye[‡], Changshui Zhang[†]
[†]State Key Laboratory on Intelligent Technology and Systems
[†]Tsinghua National Laboratory for Information Science and Technology (TNList)
[†]Department of Automation, Tsinghua University, Beijing 100084, China
[‡]Computer Science and Engineering, Center for Evolutionary Medicine and Informatics,
[‡]The Biodesign Institute, Arizona State University, Tempe, AZ 85287
[†]{gph08@mails, zcs@mail}.tsinghua.edu.cn, [‡]Jieping.Ye@asu.edu

ABSTRACT

Multi-task learning (MTL) aims to improve the performance of multiple related tasks by exploiting the intrinsic relationships among them. Recently, multi-task feature learning algorithms have received increasing attention and they have been successfully applied to many applications involving high-dimensional data. However, they assume that all tasks share a common set of features, which is too restrictive and may not hold in real-world applications, since outlier tasks often exist. In this paper, we propose a Robust Multi-Task Feature Learning algorithm (rMTFL) which simultaneously captures a common set of features among relevant tasks and identifies outlier tasks. Specifically, we decompose the weight (model) matrix for all tasks into two components. We impose the well-known group Lasso penalty on row groups of the first component for capturing the shared features among relevant tasks. To simultaneously identify the outlier tasks, we impose the same group Lasso penalty but on column groups of the second component. We propose to employ the accelerated gradient descent to efficiently solve the optimization problem in rMTFL, and show that the proposed algorithm is scalable to large-size problems. In addition, we provide a detailed theoretical analysis on the proposed rMTFL formulation. Specifically, we present a theoretical bound to measure how well our proposed rMTFL approximates the true evaluation, and provide bounds to measure the error between the estimated weights of rMTFL and the underlying true weights. Moreover, by assuming that the underlying true weights are above the noise level, we present a sound theoretical result to show how to obtain the underlying true shared features and outlier tasks (sparsity patterns). Empirical studies on both synthetic and real-world data demonstrate that our proposed rMTFL is capable of simultaneously capturing shared features among tasks and identifying outlier tasks.

KDD'12, August 12–16, 2012, Beijing, China.

Categories and Subject Descriptors

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

General Terms

Algorithm

Keywords

Multi-task learning, feature selection, outlier tasks detection

1. INTRODUCTION

Multi-task learning [8] aims to improve the performance of multiple related tasks by utilizing the intrinsic relationships among these tasks. Multi-task learning has been applied successfully in a wide range of applications including object recognition [8], speech recognition [28], handwritten digits recognition [30] and disease progression prediction [44]. A critical ingredient in these applications is how to model the shared structures among tasks. Existing algorithms can be broadly classified into two categories: explicit parameter sharing and implicit structure sharing.

Under explicit parameter sharing, all the tasks explicitly share some common parameters; examples include hidden units in neural networks [8, 5], prior in hierarchical Bayesian models [4, 31, 36, 38], parameters of Gaussian process [18], feature mapping matrix [1], classification weight [11] and similarity metric [28, 40]. On the contrary, algorithms under implicit structure sharing do not explicitly impose all tasks to share certain parameters, but they implicitly capture some common structures; for example, the algorithms in [29, 24] constrain all tasks to share a common low rank subspace and the algorithms in [27, 2, 23, 19, 22, 17, 35, 41] constrain all tasks to share a common set of features.

One key assumption of the above multi-task learning algorithms for both categories is that all tasks are related to each other by the presumed structures. However, this may not hold in real-world applications, as outlier tasks often exist. Thus, simply assuming that all tasks share a certain structure may degrade the performance. This motivates the development of several recent multi-task learning algorithms for discovering the inherent relationship among tasks. For example, some multi-task learning algorithms [32, 34, 14, 42, 16] cluster the given tasks into different groups and impose the tasks in the same groups to share a certain common structure. Multi-task learning algorithms with a composite

regularization [15, 9, 10] have been proposed to capture different types of relationships using regularization.

In this paper, we consider the multi-task learning setting where the relevant tasks share a common set of features while outlier tasks exist. We propose a Robust Multi-Task Feature Learning algorithm (rMTFL) which simultaneously captures the shared features among relevant tasks and detects outlier tasks. Specifically, we decompose the weight matrix W consisting of the prediction models of all tasks into the sum of two components P and Q. We employ the well-known group Lasso penalty on row groups of P such that the relevant tasks capture a common set of features. In addition, we employ the same group Lasso penalty but on column groups of Q to simultaneously identify the outlier tasks. The main contributions of this paper include:

(1) We propose a Robust Multi-Task Feature Learning formulation (rMTFL) which simultaneously captures a common set of features among relevant tasks and identifies outlier tasks. We propose to employ accelerated gradient descent to efficiently solve the optimization problem involved in rMTFL, and show that the proposed algorithm is scalable to large-size problems.

(2) We present a theoretical bound to measure how well rMTFL can approximate the underlying true evaluation, and give bounds to measure the error between the weights estimated from rMTFL and the underlying true weights. Moreover, by assuming that the underlying true weights are above the noise level, we present a sound theoretical result to show how we can obtain the underlying true shared features and outlier tasks (sparsity patterns).

(3) We perform empirical studies using both synthetic and real-world data. Our experiments demonstrate the efficiency of the proposed algorithm. Results also demonstrate the effectiveness of rMTFL for capturing shared features among tasks and identifying outlier tasks simultaneously.

Organization: The remainder of this paper is organized as follows: In Section 2, we introduce our proposed rMTFL formulation. In Section 3, we present the proposed optimization algorithm for rMTFL. In Section 4, we provide a detailed theoretical analysis on rMTFL. In Section 5, we discuss related work. Experimental results are presented in Section 6 and we conclude the paper in Section 7.

Notations: Scalars, vectors, matrices and sets are denoted by lower case letters, bold face lower case letters, capital letters and calligraphic capital letters, respectively. x_i and x_{ij} denote the i-th entry of a vector $\mathbf{x}$ and the (i,j)-th entry of a matrix X. $\mathbf{x}^i$ ($\mathbf{x}_i$) denotes the i-th row (column) of a matrix X. $X^{\mathcal{J}}$ denotes a submatrix composed of the rows of X indexed by $\mathcal{J}$. $x_{jk}^{(i)}$ and $\mathbf{x}_j^{(i)}$ denote the (j,k)-th entry and the j-th column of a matrix X_i. Euclidean and Frobenius norms are denoted by $\|\cdot\|$ and $\|\cdot\|_F$. $\ell_{p,q}$-norm of a matrix X is defined as $\|X\|_{p,q} = \left(\sum_i \left((\sum_j x_{ij}^q)^{1/q}\right)^p\right)^{1/p}$ and the inner product of X and Y is denoted by $\langle X, Y\rangle$. $\mathbb{N}_m$ is defined as the set $\{1, \cdots, m\}$ and $N(\mu, \sigma^2)$ denotes a normal distribution with mean μ and standard deviation σ.

2. THE PROPOSED FORMULATION

Assume that we are given m learning tasks associated with the training data $\{(X_1, \mathbf{y}_1), \cdots, (X_m, \mathbf{y}_m)\}$, where $X_i \in \mathbb{R}^{d\times n_i}$ is the data matrix of the i-th task with each column as a sample; $\mathbf{y}_i \in \mathbb{R}^{n_i}$ is the response of the i-th task (y_i has continuous values for regression and discrete values for classification); d is the data dimensionality; n_i is the number of samples for the i-th task. The data has been normalized such that the (j,k)-th entry of X_i denoted as $x_{jk}^{(i)}$ satisfies

$$\sum_{k=1}^{n_i} \left(x_{jk}^{(i)}\right)^2 = 1, \forall j \in \mathbb{N}_d. \tag{1}$$

We consider learning a linear function

$$y_{ji} \approx f_i\left(\mathbf{x}_j^{(i)}\right) = \left(\mathbf{x}_j^{(i)}\right)^T \mathbf{w}_i, i \in \mathbb{N}_m, j \in \mathbb{N}_{n_i}$$

for each task and decomposing the weight matrix $W = [\mathbf{w}_1, \cdots, \mathbf{w}_m] \in \mathbb{R}^{d\times m}$ into the sum of two components P and Q (Please refer to Figure 1 for illustration). We make use of different regularization terms on P and Q to exploit relationships among tasks. Formally, our rMTFL model is formulated as:

$$\min_{W,P,Q} \sum_{i=1}^{m} \frac{1}{mn_i} \left\|X_i^T \mathbf{w}_i - \mathbf{y}_i\right\|^2 + \lambda_1 \|P\|_{1,2} + \lambda_2 \left\|Q^T\right\|_{1,2},$$
$$s.t.\ W = P + Q, \tag{2}$$

where the first regularization term on P captures the shared features among tasks and the second term on Q discovers the outlier tasks; λ_1 and λ_2 are nonnegative parameters to control these two terms. Specifically, the first regularization term is based on the well-known group Lasso penalty on row groups of P which restricts the rows of the optimal solution $P^\star$ to consist of all zero or nonzero elements [2]. Thus, all related tasks should select a common set of features. However, the assumption that all tasks share the same set of features may not hold in real applications, as outlier tasks often exist. To address this issue, we introduce the second regularization term based on the same group Lasso penalty but on column groups of Q to discover these outlier tasks. Similarly, the columns of the optimal solution $Q^\star$ consist of all zero or nonzero elements, with the nonzero columns corresponding to outlier tasks. Intuitively, if the i-th column of $Q^\star$ is nonzero, then the i-th column of $W^\star$ is also nonzero, thus the i-th task does not share a common set of features with other tasks, identified as an outlier task; meanwhile, for the remaining tasks corresponding to the zero columns of $Q^\star$, they share a common set of features captured by the nonzero rows of $P^\star$ (see Figure 1).

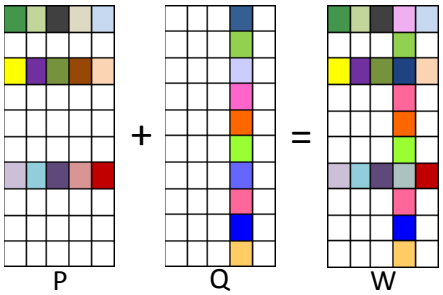

Figure 1: Illustration of weight matrix decomposition for rMTFL, where squares with white background denote zero entries. There are 5 tasks, where the fourth task is an outlier task. Please refer to the text for detailed explanation.

3. OPTIMIZATION ALGORITHM

In this section, we show how to solve the rMTFL formulation in Eq. (2) efficiently. Denote

$$l(P,Q) = \sum_{i=1}^{m} \frac{1}{mn_i} \left\| X_i^T(\mathbf{p}_i + \mathbf{q}_i) - \mathbf{y}_i \right\|^2,$$
$$r(P,Q) = \lambda_1 \|P\|_{1,2} + \lambda_2 \left\| Q^T \right\|_{1,2}, \quad (3)$$

where $l(P,Q)$ is the empirical loss function and $r(P,Q)$ is the regularization term. We note that the objective function in Eq. (2) is a composite function of a differential term $l(P,Q)$ and a non-differential term $r(P,Q)$. Denote

$$T_{R,S,\eta}(P,Q) = l(R,S) + \left\langle \frac{\partial l(R,S)}{\partial R}, P-R \right\rangle + \frac{\eta}{2}\|P-R\|_F^2$$
$$+ \left\langle \frac{\partial l(R,S)}{\partial S}, Q-S \right\rangle + \frac{\eta}{2}\|Q-S\|_F^2, \quad (4)$$

which is the first order Taylor expansion of $l(P,Q)$ at (R,S), with the squared Euclidean distance between (P,Q) and (R,S) as the regularization term. The traditional gradient descent algorithm obtains the solution at the k-th iteration ($k \geq 1$) by $(P^k, Q^k) = \arg\min_{P,Q} T_{P^{k-1},Q^{k-1},\eta_k}(P,Q) + r(P,Q)$ with a proper step size η_k. Here we propose to employ the accelerated gradient descent [25, 26] to solve the optimization problem, which generates the solution at the k-th iteration ($k \geq 1$) by computing the following proximal operator [20, 19, 21, 12, 37, 3]:

$$(P^k, Q^k) = \arg\min_{P,Q} T_{R^k,S^k,\eta_k}(P,Q) + r(P,Q), \quad (5)$$

where $R^1 = P^0, S^1 = Q^0$ and $R^{k+1} = P^k + \alpha_k(P^k - P^{k-1}), S^{k+1} = Q^k + \alpha_k(Q^k - Q^{k-1})$ for $k \geq 1$; η_k ($k \geq 1$) is set by finding the smallest nonnegative integer m_k such that with $\eta_k = 2^{m_k}\eta_{k-1}$:

$$l(P^k, Q^k) \leq T_{R^k,S^k,\eta_k}(P^k, Q^k). \quad (6)$$

We note that (R^{k+1}, S^{k+1}) is in fact a linear combination of (P^k, Q^k) and (P^{k-1}, Q^{k-1}). The coefficient α_k plays an important role in the convergence of the algorithm. As suggested by [6], we set $\alpha_k = (t_{k-1} - 1)/t_k$, where $t_0 = 1$ and $t_k = \left(1 + \sqrt{t_{k-1}^2 + 1}\right)/2$ for $k \geq 1$. According to the theoretical analysis in [6], we present the following convergence result for rMTFL:

THEOREM 1. *Let (P^k, Q^k) be generated by Eq. (5) with a properly chosen η_k satisfying Eq. (6). Then for any $k \geq 1$,*

$$f(P^k, Q^k) - f(P^\star, Q^\star) = O\left(\frac{1}{k^2}\right), \quad (7)$$

where $f(\cdot,\cdot)$ and $(P^\star, Q^\star)$ are respectively the objective function and the optimal solution in Eq. (2).

3.1 Implementation details

There are two issues that remain to be addressed: how to compute the proximal operator in Eq. (5) and how to select a proper initial value η_0.

Due to the decomposable property of Eq. (5), we can cast Eq. (5) into the following two separate proximal operator problems:

$$P^k = \arg\min_P \frac{1}{2}\left\| P - \left(R^k - \frac{1}{\eta_k}\nabla_R l(R^k, S^k)\right)\right\|_F^2 + \frac{\lambda_1}{\eta_k}\|P\|_{1,2}$$
$$Q^k = \arg\min_Q \frac{1}{2}\left\| Q - \left(S^k - \frac{1}{\eta_k}\nabla_S l(R^k, S^k)\right)\right\|_F^2 + \frac{\lambda_2}{\eta_k}\left\|Q^T\right\|_{1,2}$$

where $\nabla_R l(R^k, S^k)$ and $\nabla_S l(R^k, S^k)$ are the partial derivatives of $l(R,S)$ with respect to S and R at (R^k, S^k). The above proximal operator problems admit closed form solutions with time complexity of $O(dm)$ [19]:

$$\left(\mathbf{p}^{(k)}\right)^i = \max\left(0, 1 - \frac{\lambda_1}{\eta_k \left\|(\mathbf{u}^{(k)})^i\right\|}\right)\left(\mathbf{u}^{(k)}\right)^i, \ \forall i \in \mathbb{N}_d,$$
$$\mathbf{q}_j^{(k)} = \max\left(0, 1 - \frac{\lambda_2}{\eta_k \left\|\mathbf{v}_j^{(k)}\right\|}\right)\mathbf{v}_j^{(k)}, \ \forall j \in \mathbb{N}_m,$$

where $U^k = R^k - \frac{1}{\eta_k}\nabla_R l(R^k, S^k), V^k = S^k - \frac{1}{\eta_k}\nabla_S l(R^k, S^k)$; $\left(\mathbf{u}^{(k)}\right)^i$ and $\mathbf{v}_j^{(k)}$ denote the i-th row of U^k and the j-th column of V^k, respectively.

An appropriate choice for η_0 is the Lipschitz constant L of the gradient of $l(P,Q)$. However, the Lipschitz constant L is unknown and calculating it is computationally expensive. Next, we show how to estimate its lower and upper bounds. Denote by $D \in \mathbb{R}^{dm \times (\sum_{i=1}^m n_i)}$ a block diagonal matrix with $\sqrt{\frac{2}{mn_i}}X_i$ ($i \in \mathbb{N}_m$) as the i-th block. Then the Lipschitz constant L is just the squared maximum singular value of D. According to matrix norm properties [13], we can bound L as follows:

$$\min\left(\frac{\|D\|_{\infty,1}^2}{\sum_{i=1}^m n_i}, \frac{\left\|D^T\right\|_{\infty,1}^2}{dm}\right) \leq L \leq \|D\|_{\infty,1}\left\|D^T\right\|_{\infty,1}. \quad (8)$$

We note that D is a block diagonal matrix and it is sparse when m (the number of tasks) is large, which makes the bounds of L very tight. If we set η_0 as the upper bound of L, then we do not need line search, because when $\eta_k \geq L$, Eq. (6) is always satisfied [6]. Otherwise, line search is necessary. Although setting η_0 as the upper bound of L can eliminate line search, it may increase the outer iterative steps. On the contrary, it leads to a smaller outer iterative steps by setting η_0 as the lower bound of L. In our experiments, we use the lower bound of L to initialize η.

4. THEORETICAL ANALYSIS

4.1 Basic Assumption

We assume that the responses are given by a linear model plus Gaussian noise[1], i.e.,

$$y_{ji} = f_i^\star\left(\mathbf{x}_j^{(i)}\right) + \delta_{ji} = \left(\mathbf{x}_j^{(i)}\right)^T \mathbf{w}_i^\star + \delta_{ji}, i \in \mathbb{N}_m, j \in \mathbb{N}_n, \quad (9)$$

where $W^\star$ is the true weight matrix decomposed as the sum of two underlying true components $P^\star$ and $Q^\star$:

$$W^\star = [\mathbf{w}_1^\star, \cdots, \mathbf{w}_m^\star] = P^\star + Q^\star \in \mathbb{R}^{d \times m}; \quad (10)$$

[1]For notation simplicity, we assume that the number of training samples of all tasks are the same. However, the following theoretical analysis can be easily extended to the case with different training sample sizes for different tasks.

$$X_i = \left[\mathbf{x}_1^{(i)}, \cdots, \mathbf{x}_n^{(i)}\right]^T \in \mathbb{R}^{d\times n}, \mathbf{y}_i = [y_{1i}, \cdots, y_{ni}]^T \in \mathbb{R}^n \tag{11}$$

are respectively the training data and responses of the i-th task;

$$\boldsymbol{\delta}_i = [\delta_{1i}, \cdots, \delta_{ni}]^T \in \mathbb{R}^n, \delta_{ji} \sim N(0, \sigma^2), i \in \mathbb{N}_m, j \in \mathbb{N}_n, \tag{12}$$

$$\mathbf{f}_i^\star = X_i^T \mathbf{w}_i^\star = \left[f_i^\star\left(\mathbf{x}_1^{(i)}\right), \cdots, f_i^\star\left(\mathbf{x}_n^{(i)}\right)\right]^T \in \mathbb{R}^n \tag{13}$$

are the i.i.d. normal noise and the true evaluation, respectively. Thus, we have

$$\mathbf{y}_i = \mathbf{f}_i^\star + \boldsymbol{\delta}_i,\ i \in \mathbb{N}_m. \tag{14}$$

We also define

$$\mathcal{J}(P) = \{i \mid \mathbf{p}^i \neq \mathbf{0}\},\ \mathcal{J}_\perp(P) = \{i \mid \mathbf{p}^i = \mathbf{0}\} \tag{15}$$

as the index sets for the nonzero and zero rows of P.

4.2 Theoretical Bounds

The following theorem provides a key property of the optimal solution of Eq. (2), which is critical for our subsequent theoretical analysis:

THEOREM 2. *Let $(\hat{P}, \hat{Q})$ be an optimal solution of Eq. (2) for $m \geq 2$ and $n, d \geq 1$. Let X_i and $\mathbf{y}_i$ be defined in Eq. (11); let $\boldsymbol{\delta}_i$ and $\mathbf{f}_i^\star$ be defined in Eq. (12) and Eq. (13), respectively. We assume that the data is normalized as in Eq. (1). Choose the regularization parameters λ_1 and λ_2 as*

$$\lambda_1, \lambda_2 \geq \alpha,\ \alpha = \frac{2\sigma}{mn}\sqrt{dm + t}, \tag{16}$$

where t is a positive scalar. Then with probability of at least $1 - \exp\left(-\frac{1}{2}\left(t - dm\log\left(1 + \frac{t}{dm}\right)\right)\right)$, for any $P, Q \in \mathbb{R}^{d\times m}$, we have

$$\sum_{i=1}^m \frac{1}{mn}\left\|X_i^T(\hat{\mathbf{p}}_i + \hat{\mathbf{q}}_i) - \mathbf{f}_i^\star\right\|^2 \leq \sum_{i=1}^m \frac{1}{mn}\left\|X_i^T(\mathbf{p}_i + \mathbf{q}_i) - \mathbf{f}_i^\star\right\|^2 + 2\lambda_1\left\|(\hat{P} - P)^{\mathcal{J}(P)}\right\|_{1,2} + 2\lambda_2\left\|(\hat{Q}^T - Q^T)^{\mathcal{J}(Q^T)}\right\|_{1,2}. \tag{17}$$

Based on Theorem 2, we present some performance bounds of our rMTFL model in Eq. (2). We first introduce some notations to unclutter the equations. Let $X \in \mathbb{R}^{dm\times mn}$ be a block diagonal matrix with $X_i \in \mathbb{R}^{d\times n}$ $(i \in \mathbb{N}_m)$ as the i-th block. Define a vectorization operator 'vec' over an arbitrary matrix $A \in \mathbb{R}^{d\times m}$ such that $\mathrm{vec}(A) = [\mathbf{a}_1^T, \cdots, \mathbf{a}_m^T]^T$. Then, Eq. (17) can be rewritten as

$$\begin{aligned}&\frac{1}{mn}\left\|X^T\mathrm{vec}(\hat{P} + \hat{Q}) - \mathrm{vec}(F^\star)\right\|^2 \\ &\leq \frac{1}{mn}\left\|X^T\mathrm{vec}(P + Q) - \mathrm{vec}(F^\star)\right\|^2 \\ &+ 2\lambda_1\left\|(\hat{P} - P)^{\mathcal{J}(P)}\right\|_{1,2} + 2\lambda_2\left\|(\hat{Q}^T - Q^T)^{\mathcal{J}(Q^T)}\right\|_{1,2},\end{aligned} \tag{18}$$

where $F^\star = [\mathbf{f}_1^\star, \cdots, \mathbf{f}_m^\star] \in \mathbb{R}^{n\times m}$. Next, we make the following assumption about the training data and the weight matrix, which generalizes the restricted eigenvalue assumption in [7].

ASSUMPTION 1. *For a matrix pair $\Gamma_P \in \mathbb{R}^{d\times m}$ and $\Gamma_Q \in \mathbb{R}^{d\times m}$, let r and c $(1 \leq r \leq d, 1 \leq c \leq m)$ be the upper bounds of $|\mathcal{J}(P^\star)|$ and $|\mathcal{J}(Q^{\star T})|$, respectively, and let β_1 and β_2 be positive scalars. We assume that there exist positive scalars $\kappa_1(r)$ and $\kappa_2(c)$ such that*

$$\kappa_1(r) = \min_{\Gamma_P, \Gamma_Q \in \mathcal{R}(r,c)} \frac{\left\|X^T\mathrm{vec}(\Gamma_P + \Gamma_Q)\right\|}{\sqrt{mn}\left\|(\Gamma_P)^{\mathcal{J}(P)}\right\|_F} \tag{19}$$

$$\kappa_2(c) = \min_{\Gamma_P, \Gamma_Q \in \mathcal{R}(r,c)} \frac{\left\|X^T\mathrm{vec}(\Gamma_P + \Gamma_Q)\right\|}{\sqrt{mn}\left\|(\Gamma_Q^T)^{\mathcal{J}(Q^T)}\right\|_F}, \tag{20}$$

where the set $\mathcal{R}(r, c)$ is defined as

$$\begin{aligned}\mathcal{R}(r,c) = \Big\{&\Gamma_P, \Gamma_Q \in \mathbb{R}^{d\times m} \mid \Gamma_P \neq 0, \Gamma_Q \neq 0, |\mathcal{J}(P)| \leq r, \\ &\left|\mathcal{J}(Q^T)\right| \leq c, \left\|(\Gamma_P)^{\mathcal{J}_\perp(P)}\right\|_{1,2} \leq \beta_1\left\|(\Gamma_P)^{\mathcal{J}(P)}\right\|_{1,2}, \\ &\left\|(\Gamma_Q^T)^{\mathcal{J}_\perp(Q^T)}\right\|_{1,2} \leq \beta_2\left\|\left(\Gamma_Q^T\right)^{\mathcal{J}(Q^T)}\right\|_{1,2}\Big\},\end{aligned}$$

$\mathcal{J}(\cdot)$ is defined in Eq. (15) and $|\mathcal{J}|$ denotes the number of elements in the set $\mathcal{J}$.

Note that Assumption 1 is related to the restricted eigenvalue assumption which is a critical condition in [7]. Some previous studies on multi-task learning [22, 10] also make use of similar assumptions. Our main theoretical result is summarized in the following theorem for performance bounds.

THEOREM 3. *Let $(\hat{P}, \hat{Q})$ be an optimal solution of Eq. (2) for $m \geq 2$ and $n, d \geq 1$ and take the regularization parameters λ_1 and λ_2 as in Eq. (16). Then under Assumption 1, the following results hold with probability of at least $1 - \exp\left(-\frac{1}{2}\left(t - dm\log\left(1 + \frac{t}{dm}\right)\right)\right)$ $(t > 0)$:*

$$\frac{1}{mn}\left\|X^T\mathrm{vec}(\hat{P} + \hat{Q}) - \mathrm{vec}(F^\star)\right\|^2 \leq \left(\frac{2\lambda_1\sqrt{r}}{\kappa_1(r)} + \frac{2\lambda_2\sqrt{c}}{\kappa_2(c)}\right)^2, \tag{21}$$

$$\left\|\hat{P} - P^\star\right\|_{1,2} \leq \frac{(\beta_1 + 1)\sqrt{r}}{\kappa_1(r)}\left(\frac{2\lambda_1\sqrt{r}}{\kappa_1(r)} + \frac{2\lambda_2\sqrt{c}}{\kappa_2(c)}\right), \tag{22}$$

$$\left\|\hat{Q}^T - Q^{\star T}\right\|_{1,2} \leq \frac{(\beta_2 + 1)\sqrt{c}}{\kappa_2(c)}\left(\frac{2\lambda_1\sqrt{r}}{\kappa_1(r)} + \frac{2\lambda_2\sqrt{c}}{\kappa_2(c)}\right). \tag{23}$$

If in addition, the following conditions hold:

$$\min_{j\in\mathcal{J}(P^\star)}\left\|(\mathbf{p}^\star)^j\right\| > \frac{2(\beta_1 + 1)\sqrt{r}}{\kappa_1(r)}\left(\frac{2\lambda_1\sqrt{r}}{\kappa_1(r)} + \frac{2\lambda_2\sqrt{c}}{\kappa_2(c)}\right), \tag{24}$$

$$\min_{j\in\mathcal{J}(Q^{\star T})}\|(\mathbf{q}^\star)_j\| > \frac{2(\beta_2 + 1)\sqrt{c}}{\kappa_2(c)}\left(\frac{2\lambda_1\sqrt{r}}{\kappa_1(r)} + \frac{2\lambda_2\sqrt{c}}{\kappa_2(c)}\right), \tag{25}$$

then with the same probability, the following two sets

$$\hat{\mathcal{J}}_1 = \left\{j \mid \left\|\hat{\mathbf{p}}^j\right\| > \frac{(\beta_1 + 1)\sqrt{r}}{\kappa_1(r)}\left(\frac{2\lambda_1\sqrt{r}}{\kappa_1(r)} + \frac{2\lambda_2\sqrt{c}}{\kappa_2(c)}\right)\right\}, \tag{26}$$

$$\hat{\mathcal{J}}_2 = \left\{j \mid \|\hat{\mathbf{q}}_j\| > \frac{(\beta_2 + 1)\sqrt{c}}{\kappa_2(c)}\left(\frac{2\lambda_1\sqrt{r}}{\kappa_1(r)} + \frac{2\lambda_2\sqrt{c}}{\kappa_2(c)}\right)\right\} \tag{27}$$

estimate the true sparsity pattern $\mathcal{J}(P^\star)$ and $\mathcal{J}(Q^{\star T})$, re-

spectively. That is,

$$\hat{\mathcal{J}}_1 = \mathcal{J}(P^\star), \tag{28}$$

$$\hat{\mathcal{J}}_2 = \mathcal{J}(Q^{\star T}). \tag{29}$$

Theorem 3 provides important theoretical guarantee for rMTFL. Specifically, these bounds not only measure how well our rMTFL model can approximate the true evaluation values defined in Eq. (9) [Eq. (21)], but also measure how well our rMTFL model can approximate the true weight matrices ($P^\star, Q^\star, W^\star = P^\star + Q^\star$) [Eq. (22) and Eq. (23)]. Moreover, under the assumption that the underlying true weights are above the noise level [Eq. (24) and Eq. (24)], we can also estimate the true sparsity patterns (i.e., $\mathcal{J}(P^\star), \mathcal{J}(Q^{\star T})$) with high probability [Eq. (26) and Eq. (27)].

5. RELATED WORK

Previous studies in [15, 9, 10] also decompose the weight matrix into two components; rMTFL differs from these work in several aspects:

(1) rMTFL employs different regularization terms from the algorithms in [15, 9, 10]. The regularization terms in rMTFL not only have intuitive explanations for feature selection and outlier tasks detection (see Figure 1 and detailed explanation in Section 2), but also have sound theoretical guarantee (see Section 4).

(2) rMTFL has the mechanism of detecting outlier tasks, unlike the algorithms in [15, 9]. Although the algorithm in [10] has the ability to detect the outlier, it focuses on capturing the low rank structure among tasks, while rMTFL has advantages on high dimensional multi-task feature learning problems. Specifically, in terms of the evaluation performance, the main difference between our Theorem 2 and Lemma 4.3 in [10] is that the bound of rMTFL is based on $\|(\hat{P} - P)^{\mathcal{J}(P)}\|_{1,2}$, while the bound of RMTL in [10] is based on $\|\mathcal{Q}(\hat{L} - L)\|_{\mathrm{tr}}$. In practical multi-task learning problems, the dimensionality is often high and the underlying selected features are few, that is, the number of elements in $\mathcal{J}(P)$ can be small, which indicates that $\|(\hat{P} - P)^{\mathcal{J}(P)}\|_{1,2}$ is small, leading to a tight bound in Theorem 2. However, in the scenario of a large number of tasks, the relevant tasks may share a low rank subspace, resulting in a small value of $\|\mathcal{Q}(\hat{L} - L)\|_{\mathrm{tr}}$. Therefore, rMTFL has an advantage of identifying a few shared features for high dimensional data, while RMTL in [10] focuses on discovering low rank subspace among a large number of tasks. Our experimental results in Section 6.4 demonstrate that rMTFL outperforms RMTL in the high dimensional scenario, and RMTL is preferred when the number of tasks is large but the dimensionality is low.

(3) Unlike the analysis in [10], we provide theoretical bounds to measure the error between the estimated weights of rMTFL and the underlying true weights. Moreover, we have theoretically shown under what conditions we can obtain the underlying true shared features and outlier tasks (sparsity patterns). In addition, both Theorem 2 and Theorem 3 work with probability of at least $1-\exp\left(-\frac{1}{2}\left(t - dm\log\left(1+\frac{t}{dm}\right)\right)\right)$ which is higher than $1 - m\exp\left(-\frac{1}{2}\left(t - d\log\left(1+\frac{t}{d}\right)\right)\right)$ presented in Lemma 4.3 and Theorem 4.1 of [10].

(4) Each step of the optimization method in [10] involves SVD operation with a time complexity of $O(\min(d^2m, m^2d))$, thus it does not scale to large-size problems (e.g., the number of tasks and the dimensionality are large). As we show in Section 3.1, the optimization method of rMTFL has a much lower time complexity of $O(dm)$ and hence can be applied to large-size problems.

6. EXPERIMENTS

6.1 Competing Algorithms and Data Sets

Competing Algorithms: We compare our rMTFL algorithm on multi-task regression problems with seven representative algorithms: ridge multi-task regression (ridge), ℓ_1-norm multi-task regression (lasso), trace-norm multi-task regression (trace), $\ell_{1,2}$-norm multi-task regression (L1,2), dirty model multi-task regression (DirtyMTL) [15], sparse structures and low rank multi-task regression (SLR) [9] and robust multi-task regression (RMTL) [10]. All eight algorithms employ a quadratic loss function. Matlab codes of the rMTFL algorithm are available online [43].

Synthetic data: The synthetic data is generated as follows: we set the number of tasks $m = 30$ and each task has $n_i = 200$ samples in $d = 200$ dimension; each entry of the data matrix $X_i \in \mathbb{R}^{d\times n_i}(i \in \mathbb{N}_m)$ is sampled from the distribution $N(0, 25)$ and it is normalized such that Eq. (1) is satisfied; each entry of the ground truth weight matrices $P \in \mathbb{R}^{d\times m}$ and $Q \in \mathbb{R}^{d\times m}$ is generated from the distribution $N(0, 64)$; we set the first 160 rows of P and the first 20 columns of Q as zero vectors; the elements of the noise vector $\boldsymbol{\delta}_i \in \mathbb{R}^{n_i}(i \in \mathbb{N}_m)$ are sampled from the distribution $N(0, 1)$; the response $\mathbf{y}_i \in \mathbb{R}^{n_i}(i \in \mathbb{N}_m)$ is computed via $\mathbf{y}_i = X_i^T(P + Q) + \boldsymbol{\delta}_i$. Under this setting, we have constructed 20 related tasks and 10 outlier tasks.

Real-world Data: We adopt two data sets for our multi-task regression evaluation: School data[2] and MRI data.

The School data set is from the Inner London Education Authority (ILEA), consisting of examination records of 15362 students (samples) from 139 secondary schools in years 1985, 1986 and 1987. Each sample is represented by 27 binary attributes which include year, gender, examination score, etc., plus 1 bias attribute (In our experiments, the bias attribute is not used). The response (target) is the examination score. So we have 139 tasks with each task corresponding to one school.

The MRI data set is from the ANDI database. It contains MRI data of 675 patients preprocessed using FreeSurfer[3]. The MRI data include 306 features which can be categorized into 5 types: cortical thickness average, cortical thickness standard deviation, volume of cortical parcellation, volume of white matter parcellation, and surface area. The response (target) is the Mini Mental State Examination (MMSE) score coming from 6 different time points: M06, M12, M18, M24, M36, and M48. We remove the samples which fail the MRI quality controls and with missing entries. After the preprocessing above, we have 6 tasks with each task corresponding to a time point and the sample sizes corresponding to 6 tasks are 648, 642, 293, 569, 389 and 87, respectively.

6.2 Experimental Setting

In our experiments, we terminate all the algorithms when the relative change of the two consecutive objective func-

[2] `http://www.cs.ucl.ac.uk/staff/a.argyriou/code/`
[3] `www.loni.ucla.edu/ADNI/`

Table 1: Comparison of eight multi-task regression algorithms on the MRI data set in terms of averaged nMSE and aMSE.

measure	traning ratio	ridge	lasso	trace	L1,2	DirtyMTL	SLR	RMTL	rMTFL
nMSE	0.15	0.9494	0.6469	0.6889	0.6445	0.6355	0.6905	0.6930	**0.5743**
	0.20	0.9355	0.6242	0.6629	0.6555	0.6231	0.6648	0.6557	**0.5700**
	0.25	0.9151	0.6015	0.6230	0.6446	0.6082	0.6244	0.6239	**0.5498**
aMSE	0.15	0.0270	0.0188	0.0195	0.0184	0.0177	0.0196	0.0196	**0.0168**
	0.20	0.0262	0.0177	0.0184	0.0184	0.0172	0.0185	0.0182	**0.0163**
	0.25	0.0255	0.0170	0.0171	0.0181	0.0167	0.0171	0.0171	**0.0157**

Table 2: Comparison of eight multi-task regression algorithms on the School data set in terms of averaged nMSE and aMSE.

measure	traning ratio	ridge	lasso	trace	L1,2	DirtyMTL	SLR	RMTL	rMTFL
nMSE	0.16	1.2325	1.0457	0.7829	0.9236	0.8989	0.7812	**0.7804**	0.8628
	0.24	1.0734	0.9441	**0.7606**	0.9017	0.8413	0.7608	0.7613	0.8173
	0.32	0.9996	0.8875	0.7506	0.8972	0.8019	0.7507	**0.7504**	0.7874
aMSE	0.16	0.3154	0.2735	0.2048	0.2395	0.2358	0.2044	**0.2041**	0.2252
	0.24	0.2773	0.2469	**0.1993**	0.2341	0.2203	0.1993	0.1995	0.2135
	0.32	0.2580	0.2312	0.1958	0.2316	0.2088	0.1958	**0.1958**	0.2049

tion values is less than 10^{-5}. We randomly split the samples (both synthetic and real-world data sets) from each task into training and test samples with different training ratios. We evaluate eight multi-task regression algorithms on the test data set, using normalized mean squared error (nMSE) and averaged means squared error (aMSE) as the regression performance measures [39, 10, 42]. For each training ratio, both nMSE and aMSE are averaged over 10 random splittings of training and test sets. All parameters of the eight algorithms are tuned via 3-fold cross validation.

6.3 Synthetic Data Experiments

We set the training ratio of synthetic data generated in Section 6.1 as 20% and 30%, respectively. Experimental results (averaged nMSE and aMSE) are shown in Figure 2. We observe that rMTFL outperforms all the other competing algorithms, which demonstrates the effectiveness of rMTFL for high dimensional problems with outlier tasks.

6.3.1 *Illustration of Outlier Tasks Detection*

Next, we further demonstrate the outlier tasks detection capability of rMTFL. We firstly generate another synthetic data set following the same procedure in Section 6.1, except that each task has $n_i = 20$ samples. Then, we set $\lambda_1 = 2\sqrt{dm+t}/\sum_{i=1}^{m} n_i, \lambda_2 = 6\sqrt{dm+t}/\sum_{i=1}^{m} n_i, t = 10^{-10}$ and run rMTFL on this synthetic data until the relative change of the two consecutive objective function values is less than 10^{-5}. Figure 3 shows the results of P and Q obtained by rMTFL. Specifically, there are 164 zero rows in P and 21 zero columns in Q. These results demonstrate the capability of rMTFL in simultaneously capturing the shared features among tasks (the nonzero rows of P) and discovering outlier tasks (the nonzero columns of Q).

6.3.2 *Scalability Studies on rMTFL*

We conduct scalability studies on two algorithms which are capable of identifying outlier tasks: rMTFL and RMTL [10], when the dimension and the number of tasks increase. We firstly fix $m = 20$ and let the dimension d increase as $\{50^i\}, i = 0, \cdots, 5$. Then we run rMTFL and RMTL on the synthetic data generated following the same procedure in Section 6.1. The computational time (CPU time) vs. dimension plot is shown in the left subfigure of Figure 4.

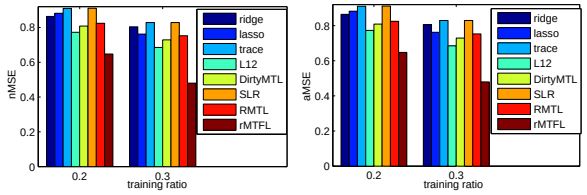

Figure 2: Averaged test error (nMSE and aMSE) vs. training ratio for synthetic data.

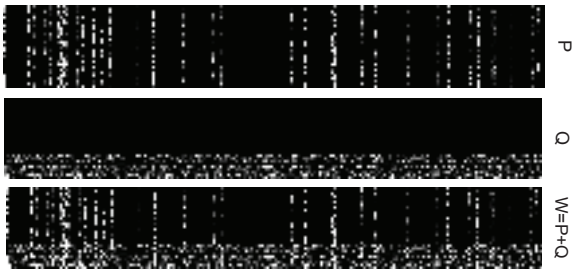

Figure 3: Figures of P, Q and $W = P + Q$ generated from rMTFL on the synthetic data. Black points correspond to zero entries. Note that the figures are clockwise rotated 90 degrees.

Similarly, we fix $d = 100$ and let the number of tasks m increase as $\{10^i\}, i = 0, \cdots, 5$. Then we run rMTFL and RMTL and show the computational time (CPU time) vs. the number of tasks plot as in the right subfigure of Figure 4. We observe that, the CPU time of both rMTFL and RMTL increases when the dimension d (or the number of tasks m) increases. However, the CPU time of RMTL increases significantly faster than rMTFL. Because the SVD computation with a time complexity of $O(\min(d^2m, m^2d))$ involved at each step of RMTL is computationally much more expensive than the computation of the $\ell_{1,2}$-norm proximal operator with a time complexity of $O(dm)$ involved at each step of rMTFL. This demonstrates the superior scalability of rMTFL over RMTL.

6.4 Real-world Data Experiments

For the School data set, we respectively set the training ratio as 16%, 24%, 32%, and for the MRI data set, we respectively set the training ratio as 15%, 20%, 25%. Table 1 and

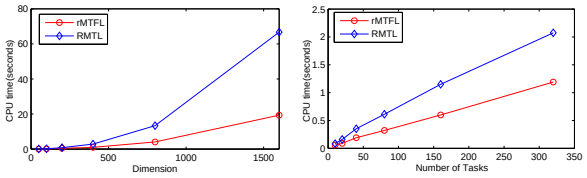

Figure 4: CPU time vs. dimension (left) and number of tasks (right) plots for rMTFL and RMTL. The CPU time is averaged over 10 independent runs.

Table 2 show the experimental results in terms of averaged nMSE and aMSE.

From these results, we have the following observations: (1) rMTFL outperforms all the other algorithms on the MRI data set. This may be due to the fact that for the MRI data set, the dimension ($d = 306$) is high especially when compared with the number of tasks ($m = 6$). (2) For the School data set, the multi-task learning algorithms based on trace norm (low rank) regularization (trace, SLR, RMTL) outperform the multi-task learning algorithms based on $\ell_{1,q}$-norm (feature selection) regularization (lasso, L1,2, DirtyMTL, rMTFL). This may be due to the fact that the number of tasks ($m = 139$) of the School data set is larger compared with dimension ($d = 27$). In this case, restricting all tasks to share a low rank subspace is more reasonable than restricting all tasks to share a few common features. (3) On both data sets, the performance of rMTFL is the best among the feature selection based multi-task learning algorithms (lasso, L1,2, DirtyMTL, rMTFL). This may be due to rMTFL's capability of simultaneously discovering the shared features and identifying outlier tasks.

6.4.1 *Outlier Tasks Detection*

The proposed rMTFL algorithm is capable of capturing the shared features among tasks and detecting outlier tasks. We next evaluate the outlier tasks detection performance on the MRI data set. Firstly, we run rMTFL on the whole MRI data set and observe that the fourth task is identified as an outlier task. Then, we remove the fourth task, obtaining a new multi-task regression problem with the remaining 5 tasks. Finally, we run $\ell_{1,2}$-norm multi-task regression on this 'clean' multi-task regression problem. We call this two-stage procedure as 2S-rMTFL. The test errors (nMSE and aMSE) on the MRI data set are shown in Figure 5. We can clearly see that after removing the outlier tasks, 2S-rMTFL outperforms L1,2 and rMTFL, which demonstrates the effectiveness of rMTFL in detecting outlier tasks.

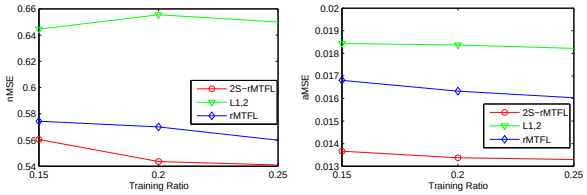

Figure 5: Test errors (nMSE and aMSE) on the MRI data set. See the texts for more details.

7. CONCLUSIONS

In this paper, we propose a Robust Multi-Task Feature Learning algorithm (rMTFL) to simultaneously capture the shared features among multiple related tasks and detect outlier tasks. We analyze the theoretical properties of rMTFL. Our analysis shows how well rMTFL can approximate the true evaluation, and measure how well rMTFL can approximate the underlying true weights. Moreover, we show that rMTFL can obtain the true sparsity patterns if the underlying true weights are above the noise level. In addition, the optimization problem involved in rMTFL can be solved efficiently, and rMTFL scales to large-size problems. In the future work, we will extend our rMTFL algorithm to multi-task learning problems with general loss functions and apply rMTFL to other real-world applications.

Acknowledgements

This work is supported in part by NSFC (Grant No. 60835002, 61075004 and 91120301), NIH (R01 LM010730) and NSF (IIS-0953662, CCF-1025177).

8. REFERENCES

[1] R. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6:1817–1853, 2005.

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

[3] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Arxiv preprint arXiv:1108.0775*, 2011.

[4] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *JMLR*, 4:83–99, 2003.

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

[6] 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.

[7] P. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

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

[9] J. Chen, J. Liu, and J. Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. In *SIGKDD*, pages 1179–1188, 2010.

[10] J. Chen, J. Zhou, and J. Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *SIGKDD*, pages 42–50, 2011.

[11] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *SIGKDD*, pages 109–117, 2004.

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

[13] R. Horn and C. Johnson. *Matrix analysis*. Cambridge University Press, 1990.

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

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

[16] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *ICML*, 2011.

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

[18] N. Lawrence and J. Platt. Learning to learn with the informative vector machine. In *ICML*, 2004.

[19] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *UAI*, pages 339–348, 2009.

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

[21] J. Liu, L. Yuan, and J. Ye. An efficient algorithm for a class of fused lasso problems. In *SIGKDD*, pages 323–332, 2010.

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

[23] S. Negahban and M. Wainwright. Joint support recovery under high-dimensional scaling: Benefits and perils of $\ell_{1,\infty}$-regularization. *NIPS*, 21, 2008.

[24] S. Negahban and M. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 39(2):1069–1097, 2011.

[25] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer Netherlands, 2004.

[26] Y. Nesterov. Gradient methods for minimizing composite objective function. *Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Tech. Rep*, 76, 2007.

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

[28] S. Parameswaran and K. Weinberger. Large margin multi-task metric learning. *NIPS*, 23:1867–1875, 2010.

[29] T. Pong, P. Tseng, S. Ji, and J. Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20(6):3465–3489, 2010.

[30] N. Quadrianto, A. Smola, T. Caetano, S. Vishwanathan, and J. Petterson. Multitask learning without label correspondences. *NIPS*, 2010.

[31] A. Schwaighofer, V. Tresp, and K. Yu. Learning gaussian process kernels via hierarchical bayes. *NIPS*, 17:1209–1216, 2005.

[32] S. Thrun and J. O'Sullivan. Discovering structure in multiple learning tasks: The tc algorithm. In *ICML*, pages 489–497, 1996.

[33] D. Wallace. Bounds on normal approximations to student's and the chi-square distributions. *The Annals of Mathematical Statistics*, pages 1121–1130, 1959.

[34] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *JMLR*, 8:35–63, 2007.

[35] X. Yang, S. Kim, and E. Xing. Heterogeneous multitask learning with joint sparsity constraints. *NIPS*, 23, 2009.

[36] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *ICML*, pages 1012–1019, 2005.

[37] L. Yuan, J. Liu, and J. Ye. Efficient methods for overlapping group lasso. *NIPS*, 2011.

[38] J. Zhang, Z. Ghahramani, and Y. Yang. Learning multiple related tasks using latent independent component analysis. *NIPS*, 18:1585–1592, 2006.

[39] Y. Zhang and D. Yeung. Multi-task learning using generalized t process. In *AISTATS*, 2010.

[40] Y. Zhang and D. Yeung. Transfer metric learning by learning task relationships. In *SIGKDD*, pages 1199–1208, 2010.

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

[42] J. Zhou, J. Chen, and J. Ye. Clustered multi-task learning via alternating structure optimization. *NIPS*, 2011.

[43] J. Zhou, J. Chen, and J. Ye. *MALSAR: Multi-tAsk Learning via StructurAl Regularization*. Arizona State University, 2012.

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

APPENDIX

To prove the theorems presented in Section 4.2, we first provide some basic lemmas and then give the detailed proof.

LEMMA 1. *For any matrix pair $P, \hat{P} \in \mathbb{R}^{d\times m}$, we have the following inequality:*

$$\left\|\hat{P}-P\right\|_{1,2}+\|P\|_{1,2}-\left\|\hat{P}\right\|_{1,2}\leq 2\left\|(\hat{P}-P)^{\mathcal{J}(P)}\right\|_{1,2}, \tag{30}$$

PROOF. According to Eq. (15), we have

$$P^{\mathcal{J}_\perp(P)}=0,\ \ \left\|(\hat{P}-P)^{\mathcal{J}_\perp(P)}\right\|_{1,2}=\left\|\hat{P}^{\mathcal{J}_\perp(P)}\right\|_{1,2}.$$

It follows that

$$\begin{aligned}&\left\|(\hat{P}-P)^{\mathcal{J}_\perp(P)}\right\|_{1,2}+\|P\|_{1,2}-\left\|\hat{P}\right\|_{1,2}\\ =&\left\|\hat{P}^{\mathcal{J}_\perp(P)}\right\|_{1,2}+\|P\|_{1,2}-\left\|\hat{P}\right\|_{1,2}\\ \leq&\left\|(\hat{P}-P)^{\mathcal{J}(P)}\right\|_{1,2}\end{aligned} \tag{31}$$

We note that $\left\|\hat{P}-P\right\|_{1,2}=\left\|(\hat{P}-P)^{\mathcal{J}(P)}\right\|_{1,2}+\left\|(\hat{P}-P)^{\mathcal{J}_\perp(P)}\right\|_{1,2}$. Substituting Eq. (31) into Eq. (30), we verify Lemma 1. □

LEMMA 2. *Let δ_i be I.I.D. random variables with $\delta_i \sim N(0,\sigma^2), i\in\mathbb{N}_n$ and $\sum_{i=1}^n \alpha_i^2=1$. Then we have*

$$v=\frac{1}{\sigma}\sum_{i=1}^n \alpha_i\delta_i$$

is a standard normal random variable, i.e., $v\sim N(0,1)$.

PROOF. Since δ_i are I.I.D. random variables with $\delta_i \sim N(0,\sigma^2), i\in\mathbb{N}_n$, v must be a normal random variable. Next, we need to show that the mean ($\mathbb{E}$) and variance ($\mathbb{V}$) of v are respectively 0 and 1, as given below:

$$\mathbb{E}(v)=\mathbb{E}\left(\frac{1}{\sigma}\sum_{i=1}^n\alpha_i\delta_i\right)=\frac{1}{\sigma}\sum_{i=1}^n\alpha_i\mathbb{E}(\delta_i)=0,$$

$$\mathbb{V}(v)=\mathbb{E}\left(v^2\right)=\frac{1}{\sigma^2}\sum_{i=1}^n\alpha_i^2\mathbb{E}(\delta_i^2)=1.$$

□

LEMMA 3. *Let $\chi^2(d)$ be a chi-squared random variable with d degrees of freedom. Then, the following holds:*

$$\Pr\left(\chi^2(d)\geq d+t\right)\leq\exp\left(-\frac{1}{2}\left(t-d\log\left(1+\frac{t}{d}\right)\right)\right), t>0.$$

PROOF. By the Wallace inequality [33], we obtain

$$\Pr\left(\chi^2(d)\geq d+t\right)\leq\Pr\left(N>z(t)\right), t>0, \tag{32}$$

where N is a standard normal random variable and $z(t)=\sqrt{t-d\log(1+\frac{t}{d})}$. Lemma 3 follows directly from Eq. (32) and inequality $\Pr\left(N\geq z(t)\right)\leq\exp\left(-\frac{z(t)^2}{2}\right)$. □

Proof of Theorem 2:

PROOF. Since $(\hat{P},\hat{Q})$ is an optimal solution of Eq. (2), the following holds for any P and Q:

$$\begin{aligned}&\sum_{i=1}^m\frac{1}{mn}\left\|X_i^T(\hat{\mathbf{p}}_i+\hat{\mathbf{q}}_i)-\mathbf{y}_i\right\|^2\leq\sum_{i=1}^m\frac{1}{mn}\left\|X_i^T(\mathbf{p}_i+\mathbf{q}_i)-\mathbf{y}_i\right\|^2\\ &+\lambda_1\left(\|P\|_{1,2}-\left\|\hat{P}\right\|_{1,2}\right)+\lambda_2\left(\left\|Q^T\right\|_{1,2}-\left\|\hat{Q}^T\right\|_{1,2}\right).\end{aligned}$$

Substituting Eq. (14) into the above inequality, we have

$$\sum_{i=1}^{m}\frac{1}{mn}\left\|X_i^T(\hat{\mathbf{p}}_i+\hat{\mathbf{q}}_i)-\mathbf{f}_i^\star\right\|^2 \le \sum_{i=1}^{m}\frac{1}{mn}\left\|X_i^T(\mathbf{p}_i+\mathbf{q}_i)-\mathbf{f}_i^\star\right\|^2 + \lambda_1\left(\|P\|_{1,2}-\left\|\hat{P}\right\|_{1,2}\right)+\lambda_2\left(\left\|Q^T\right\|_{1,2}-\left\|\hat{Q}^T\right\|_{1,2}\right) + \frac{2}{mn}\left\langle Z,\hat{P}-P\right\rangle+\frac{2}{mn}\left\langle Z,\hat{Q}-Q\right\rangle, \quad (33)$$

where $Z=[X_1\boldsymbol{\delta}_1,\cdots,X_m\boldsymbol{\delta}_m]\in\mathbb{R}^{d\times m}$ with its (j,i)-th entry given by

$$z_{ji}=\mathbf{x}_j^{(i)}\boldsymbol{\delta}_i=\sum_{k=1}^{n}x_{jk}^{(i)}\delta_{ki}, i\in\mathbb{N}_m, j\in\mathbb{N}_d, \quad (34)$$

and $x_{jk}^{(i)}$ denotes the (j,k)-th entry of data matrix X_i for the i-th task. It follows from Eq. (1) that

$$v_{ji}=\frac{1}{\sigma}z_{ji}=\frac{1}{\sigma}\sum_{k=1}^{n}x_{jk}^{(i)}\delta_{ki}\ (i\in\mathbb{N}_m, j\in\mathbb{N}_d) \quad (35)$$

are i.i.d. standard normal random variables (see Lemma 2), i.e., $v_{ji}\sim N(0,1)$. Thus,

$$u=\sum_{j=1}^{d}\sum_{i=1}^{m}v_{ji}^2=\frac{1}{\sigma^2}\|Z\|_F^2$$

is a chi-squared random variable with dm degrees of freedom, and it follows from Lemma 3:

$$\Pr\left(\frac{2}{mn}\|Z\|_F\ge\alpha\right)=\Pr\left(u\ge\frac{\alpha^2n^2m^2}{4}\right) \le \exp\left(-\frac{1}{2}\left(t-dm\log\left(1+\frac{t}{dm}\right)\right)\right)\ (t>0),$$

which is equivalent to the following:

$$\Pr\left(\frac{2}{mn}\|Z\|_F\le\alpha\right)\ge 1-\exp\left(-\frac{1}{2}\left(t-dm\log\left(1+\frac{t}{dm}\right)\right)\right). \quad (36)$$

Under the event in Eq. (36), we bound $\frac{2}{mn}\langle Z,\hat{P}-P\rangle$ as

$$\frac{2}{mn}\left\langle Z,\hat{P}-P\right\rangle\le\frac{2}{mn}\|Z\|_F\left\|\hat{P}-P\right\|_F \le\alpha\left\|\hat{P}-P\right\|_F\le\alpha\left\|\hat{P}-P\right\|_{1,2}. \quad (37)$$

Similarly, under the event in Eq. (36), we have

$$\frac{2}{mn}\left\langle Z,\hat{Q}-Q\right\rangle\le\frac{2}{mn}\|Z\|_F\left\|\hat{Q}-Q\right\|_F \le\alpha\left\|\hat{Q}-Q\right\|_F\le\alpha\left\|\hat{Q}^T-Q^T\right\|_{1,2}. \quad (38)$$

Combine Eq. (33), Eq. (37), Eq. (38) and Lemma 1, we verify Theorem 2. □

Proof of Theorem 3:

Proof. Let $\Gamma_P=\hat{P}-P$, $\Gamma_Q=\hat{Q}-Q$. Setting $P=P^\star$, $Q=Q^\star$, $P^\star$ and $Q^\star$ are the true weight matrices in Eq. (10), we have $X^T\mathrm{vec}(P+Q)=X^T\mathrm{vec}(P^\star+Q^\star)=\mathrm{vec}(F^\star)$. Following Eq. (18), we obtain

$$\frac{1}{mn}\left\|X^T\mathrm{vec}(\hat{P}+\hat{Q})-\mathrm{vec}(F^\star)\right\|^2\le 2\lambda_1\left\|(\hat{P}-P^\star)^{\mathcal{J}(P^\star)}\right\|_{1,2} + 2\lambda_2\left\|(\hat{Q}^T-Q^{\star T})^{\mathcal{J}(Q^{\star T})}\right\|_{1,2}. \quad (39)$$

Under Assumption 1, we have

$$\left\|(\hat{P}-P^\star)^{\mathcal{J}(P^\star)}\right\|_{1,2}\le\sqrt{r}\left\|(\hat{P}-P^\star)^{\mathcal{J}(P^\star)}\right\|_F \le\frac{\sqrt{r}}{\kappa_1(r)\sqrt{mn}}\left\|X^T\mathrm{vec}(\hat{P}+\hat{Q})-\mathrm{vec}(F^\star)\right\|, \quad (40)$$

$$\left\|(\hat{Q}^T-Q^{\star T})^{\mathcal{J}(Q^{\star T})}\right\|_{1,2}\le\sqrt{c}\left\|(\hat{Q}^T-Q^{\star T})^{\mathcal{J}(Q^{\star T})}\right\|_F \le\frac{\sqrt{c}}{\kappa_2(c)\sqrt{mn}}\left\|X^T\mathrm{vec}(\hat{P}+\hat{Q})-\mathrm{vec}(F^\star)\right\|. \quad (41)$$

Substituting Eq. (40) and Eq. (41) into Eq. (39), we obtain

$$\left\|X^T\mathrm{vec}(\hat{P}+\hat{Q})-\mathrm{vec}(F^\star)\right\|\le\sqrt{mn}\left(\frac{2\lambda_1\sqrt{r}}{\kappa_1(r)}+\frac{2\lambda_2\sqrt{c}}{\kappa_2(c)}\right),$$

which directly leads to Eq. (21).

Following Assumption 1, we have

$$\left\|(\Gamma_P)^{\mathcal{J}_\perp(P^\star)}\right\|_{1,2}\le\beta_1\left\|(\Gamma_P)^{\mathcal{J}(P^\star)}\right\|_{1,2},$$
$$\left\|(\Gamma_Q^T)^{\mathcal{J}_\perp(Q^{\star T})}\right\|_{1,2}\le\beta_2\left\|(\Gamma_Q^T)^{\mathcal{J}(Q^{\star T})}\right\|_{1,2},$$

which imply that

$$\left\|\hat{P}-P^\star\right\|_{1,2}\le(\beta_1+1)\left\|(\hat{P}-P^\star)^{\mathcal{J}(P^\star)}\right\|_{1,2}, \quad (42)$$

$$\left\|\hat{Q}^T-Q^{\star T}\right\|_{1,2}\le(\beta_2+1)\left\|(\hat{Q}^T-Q^{\star T})^{\mathcal{J}(Q^{\star T})}\right\|_{1,2}. \quad (43)$$

Substituting Eq. (42) and Eq. (43)) into Eq. (40) and Eq. (41), and considering Eq. (21), we can easily verify Eq. (22) and Eq. (23).

To prove Eq. (28), we need to show the following two:

$$(a)\ \forall j\in\hat{\mathcal{J}}_1\Rightarrow j\in\mathcal{J}(P^\star), \quad (44)$$

$$(b)\ \forall j\in\mathcal{J}(P^\star)\Rightarrow j\in\hat{\mathcal{J}}_1. \quad (45)$$

We first prove (a) by contradiction. Assume there exists a j_1 such that $j_1\in\hat{\mathcal{J}}_1, j_1\notin\mathcal{J}(P^\star)$. Then according to the definitions of $\hat{\mathcal{J}}_1$ and $\mathcal{J}(P^\star)$, we have

$$\left\|\hat{\mathbf{p}}^{j_1}-(\mathbf{p}^\star)^{j_1}\right\|=\left\|\hat{\mathbf{p}}^{j_1}\right\|>\frac{(\beta_1+1)\sqrt{r}}{\kappa_1(r)}\left(\frac{2\lambda_1\sqrt{r}}{\kappa_1(r)}+\frac{2\lambda_2\sqrt{c}}{\kappa_2(c)}\right),$$

which contradicts with the following fact:

$$\left\|\hat{P}-P^\star\right\|_{\infty,2}\le\left\|\hat{P}-P^\star\right\|_{1,2} \le\frac{(\beta_1+1)\sqrt{r}}{\kappa_1(r)}\left(\frac{2\lambda_1\sqrt{r}}{\kappa_1(r)}+\frac{2\lambda_2\sqrt{c}}{\kappa_2(c)}\right). \quad (46)$$

We thus verify (a). Similarly, if we assume there exists a j_2 such that $j_2\in\mathcal{J}(P^\star), j_2\notin\hat{\mathcal{J}}_1$, then using the condition in Eq. (24) and the definition of $\hat{\mathcal{J}}_1$ in Eq. (26), we have

$$\left\|\hat{\mathbf{p}}^{j_2}-(\mathbf{p}^\star)^{j_2}\right\|\ge\left\|(\mathbf{p}^\star)^{j_2}\right\|-\left\|\hat{\mathbf{p}}^{j_2}\right\| >\frac{(\beta_1+1)\sqrt{r}}{\kappa_1(r)}\left(\frac{2\lambda_1\sqrt{r}}{\kappa_1(r)}+\frac{2\lambda_2\sqrt{c}}{\kappa_2(c)}\right),$$

which contradicts with Eq. (46), thus (b) holds. Combining (a) and (b), we verify Eq. (28). Similarly, we can prove Eq. (29). □