Forward Backward Greedy Algorithms for Multi-Task Learning with Faster Rates

Lu Tian
Department of Systems and Information Engineering
University of Virginia

Pan Xu
Department of Systems and Information Engineering
University of Virginia

Quanquan Gu
Department of Systems and Information Engineering
University of Virginia

Abstract

A large body of algorithms have been proposed for multi-task learning. However, the effectiveness of many multi-task learning algorithms highly depends on the structural regularization, which incurs bias in the resulting estimators and leads to slower convergence rate. In this paper, we aim at developing a multi-task learning algorithm with faster convergence rate. In particular, we propose a general estimator for multi-task learning with row sparsity constraint on the parameter matrix, i.e., the number of nonzero rows in the parameter matrix being small. The proposed estimator is a nonconvex optimization problem. In order to solve it, we develop a forward backward greedy algorithm with provable guarantee. More specifically, we prove that the output of the greedy algorithm attains a sharper estimation error bound than many state-of-the-art multi-task learning methods. Moreover, our estimator enjoys model selection consistency under a mild condition. Thorough experiments on both synthetic and real-world data demonstrate the effectiveness of our method and back up our theory.

1 INTRODUCTION

Multi-task learning (MTL) (Caruana, 1997) has witnessed increasing attention in machine learning and statistics in the past decades. In multi-task learning, one deals with a number of related learning tasks simultaneously, with the goal to improve the generalization performance by utilizing the intrinsic relationship among these tasks. It has been successfully applied to a wide range of applications including object recognition (Caruana, 1997), speech recognition (Parameswaran and Weinberger, 2010), handwritten digits recognition (Quadrianto et al., 2010), and disease progression prediction (Zhou et al., 2011).

The fundamental problem in multi-task learning is how to characterize the relationship among tasks. Representative methods include learning hidden units in neural networks (Caruana, 1997; Baxter, 2000), sharing prior in hierarchical Bayesian models (Bakker and Heskes, 2003; Schwaighofer et al., 2004; Yu et al., 2005; Zhang et al., 2005) and Gaussian processes (Lawrence and Platt, 2004), learning a shared feature mapping matrix in multiple regression (Ando and Zhang, 2005; Evgeniou and Pontil, 2004). Some other works also proposed to learn the task relations (Zhang and Yeung, 2012, 2013; Han and Zhang, 2015), to mention a few. In this study, we focus on a large family of multi-task learning algorithms, which assume that all tasks share a common set of features (Obozinski et al., 2006; Argyriou et al., 2008; Negahban and Wainwright, 2008; Liu et al., 2009; Lounici et al., 2009; Yang et al., 2009; Zhang et al., 2010; Lounici et al., 2011), because they serve as the basis for many other sophisticated multi-task learning algorithms. Note that our method and theory can be extended to those sophisticated multi-task learning settings (Jacob et al., 2009; Kim and Xing, 2010; Kang et al., 2011; Zhang and Yeung, 2012; Gong et al., 2012; Zhang and Yeung, 2013; Han and Zhang, 2015) straightforwardly.

In detail, the multiple task learning setting (Obozinski et al., 2006; Argyriou et al., 2008; Negahban and Wainwright, 2008; Liu et al., 2009; Lounici et al., 2009; Yang et al., 2009; Zhang et al., 2010) considered in this paper is as follows: Given a set of observations $\{\mathbf{X}^{(i)}, \mathbf{y}^{(i)}\}, i = 1, \cdots, m$ from m tasks, where $\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_{n_i}^{(i)}]^\top \in \mathbb{R}^{n_i \times d}, i = 1, \cdots, m$ are the design matrices for each task, and $\mathbf{y}^{(i)} = [y_1^{(i)}, \ldots, y_{n_i}^{(i)}]^\top \in \mathbb{R}^{n_i}, i = 1, \cdots, m$ are corresponding vectors of response variables. It is often useful to represent the parameters in multiple tasks via a matrix, where each column corresponds to a task, and each row to a feature, i.e., $\boldsymbol{\Theta}^* = [\boldsymbol{\theta}_1^*, \cdots, \boldsymbol{\theta}_m^*] \in \mathbb{R}^{d \times m}$, where $\boldsymbol{\theta}_i^* \in \mathbb{R}^d$ is the parameter vector of the i-th task. We assume that, conditioned on the covariate (feature) vector $\mathbf{x}_j^{(i)}$, the response variable $y_j^{(i)}$ for each task depends on the same subset of features. In other words, $\boldsymbol{\theta}_i^*$'s are sparse and share the

same support. This corresponds to the matrix $\mathbf{\Theta}^*$ being " row-sparse": each row is either all zero or mostly nonzero, and the number of nonzero rows is relatively small. More specifically, the number of nonzero rows in $\mathbf{\Theta}^*$ is denoted by $s^* = \|\mathbf{\Theta}^*\|_{0,2}$. A lot of recent research in this setting used $\ell_{1,q}(q > 1)$ norm regularizations that encourage the parameter matrix to have such row-sparse structure. Particular examples include the $\ell_{1,\infty}$ norm regularization (Turlach et al., 2005; Zhang and Huang, 2008; Negahban and Wainwright, 2008), the $\ell_{1,2}$ norm regularization (Lounici et al., 2009; Obozinski et al., 2011), and the mixture of $\ell_{1,\infty}$ norm and $\ell_{1,1}$ norm regularizations (Jalali et al., 2013).

However, all the methods mentioned above for multi-task learning are based on convex regularization, i.e., $\ell_{1,q}, q > 1$ norm regularization. Recent studies (Fan and Li, 2001; Zhang, 2010) have shown that convex regularization based estimators suffer from the bias. To remedy this problem, one can choose nonconvex regularization alternatively such as the smoothly clipped absolute deviation (SCAD) penalty (Fan and Li, 2001) and the mimimax concave penalty (MCP) (Zhang, 2010). However, the empirical performance of nonconvex penalty regularized estimator highly relies on the parameters of the nonconvex penalty, which are difficult to tune in practice. One can also use the debiasing method proposed in Javanmard and Montanari (2014) to cancel the bias. However, the debiasing method will result in a non-sparse estimation result. In order to sparsify the result, a truncation step is needed, which introduces an extra tuning parameter. Moreover, the estimation error bound of debiased estimator is no better than that of convex relaxation based estimator.

In this paper, we aim at developing a new estimator which is able to get rid of the bias, attain faster convergence rate, and easy to implement in practice. In detail, we propose a general estimator for multi-task learning with row-sparsity constraint on the parameter matrix. Due to the nonconvex $\ell_{0,2}$ norm constraint, the estimator is a non-convex optimization problem and finding its global optimal solution is generally NP-hard. We propose a greedy algorithms to attain an approximate solution with provable guarantee. At the core of the proposed greedy algorithm is a forward backward feature selection strategy. We prove that the output of our algorithm attains a sharp statistical estimation error bound. As a special example of the proposed general estimator, we consider the multivariate linear regression model $\mathbf{y}^{(i)} = \mathbf{X}^{(i)}\boldsymbol{\theta}_i^* + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is the zero mean noise vector. We show that its estimation error bound is $O(\sqrt{s^*m/n} + \sqrt{s^* \log s^*/n})$ in terms of the Frobenius norm with $n = \min_i\{n_i\}$, which is sharper than the state-of-the-art results Jalali et al. (2013); Gong et al. (2013). Furthermore, in order to achieve model selection consistency, most existing work for the square loss function (Jalali et al., 2013; Wainwright, 2009; Zhao and Yu, 2006) relies on the very stringent incoherence condition. In sharp contrast, our estimator enjoys model selection consistency under a mild condition on the ℓ_2 norms of the nonzero rows in $\mathbf{\Theta}^*$. Thorough experiments on both simulated data and real data show that the proposed method outperforms the state-of-the-art methods.

The remainder of this paper is organized as follows. In Section 2, we propose a general estimator for multi-task learning with row sparsity constraint, followed by a greedy algorithm with forward-backward feature selection strategy. In Section 3, we prove the convergence of the greedy algorithm, as well as the statistical estimation error bound for the output of the greedy algorithm. We report the experimental results in Section 4 and conclude our work in Section 5.

Notation We use bold capitals to denote matrices, bold lowercase letters for vectors, and lowercase letters for scalars. The j-th natural basis in $\mathbb{R}^d$ is denoted as $\mathbf{e}_j$. For matrices $\mathbf{A}$ and $\mathbf{B}$ with commensurate dimensions, we use $\langle \mathbf{A}, \mathbf{B} \rangle$ to denote their trace inner product, i.e. $\langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{tr}(\mathbf{A}^\top \mathbf{B})$. Given a matrix $\mathbf{\Theta}$, its i-th row is denoted by $\mathbf{\Theta}_{i*}$ and its j-th column is denoted by $\mathbf{\Theta}_{*j}$. The $\ell_{p,q}$ norm of a matrix $\mathbf{\Theta}$ is defined as $\|\mathbf{\Theta}\|_{p,q} = \big\{ \sum_i [(\sum_j |\Theta_{ij}|^q)^{1/q}]^p \big\}^{1/p}$, and the Frobenius norm of $\mathbf{\Theta}$ is $\|\mathbf{\Theta}\|_F = \sqrt{\langle \mathbf{\Theta}, \mathbf{\Theta} \rangle} = \|\mathbf{\Theta}\|_{2,2}$. For a matrix $\mathbf{\Theta}$, we use $F(\mathbf{\Theta})$ to denote the index set of the non-zero rows in $\mathbf{\Theta}$. For a row index set F, we denote by $\mathbf{\Theta}_F$ to be the matrix that its i-th row is the same as the i-th row of $\mathbf{\Theta}$ if $i \in F$, and its i-th row is a zero vector if $i \notin F$.

2 THE PROPOSED METHOD

In this section, we first introduce the underlying model for multi-task learning, followed by a general estimator. Then we propose a greedy algorithm to solve the estimator.

2.1 THE MODEL AND ESTIMATOR

Suppose that we have observations $\{(\mathbf{X}^{(1)}, \mathbf{y}^{(1)}), \ldots, (\mathbf{X}^{(m)}, \mathbf{y}^{(m)})\}$ from m tasks, where $\mathbf{X}^{(i)} \in \mathbb{R}^{n_i \times d}$ is the design matrix of the i-th task, $\mathbf{y}_i \in \mathbb{R}^{n_i}$ is the vector of response variables for the i-th task. We assume that the observations in each task are generated from generalized linear models

$$\mathbb{P}(y_j^{(i)} | \mathbf{x}_j^{(i)}, \boldsymbol{\theta}_i^*, \sigma_i) = \exp\left\{ \frac{y_j^{(i)} \langle \boldsymbol{\theta}_i^*, \mathbf{x}_j^{(i)} \rangle - \Phi(\boldsymbol{\theta}_i^{*\top} \mathbf{x}_j^{(i)})}{c(\sigma_i)} \right\}, \quad i = 1, \ldots, m, j = 1, \ldots, n_i,$$

where $\Phi(\cdot) : \mathbb{R} \to \mathbb{R}$ is a link function, $\mathbf{x}_j^{(i)}$ is the j-th row of $\mathbf{X}^{(i)}$, $y_j^{(i)}$ is the j-th coordinate of $\mathbf{y}^{(i)}$, $\boldsymbol{\theta}_i^* \in \mathbb{R}^d$ is the parameter of i-th task, and $c(\sigma_i) \in \mathbb{R}$ is fixed and known scale parameter of the i-th task. A special example

of generalized linear model is the linear regression model where the distribution of the response variable conditioned on the covariates is a normal distribution. That is, when $c(\sigma)$ is chosen as σ^2 and $\Phi(t) = t^2$. Logistic regression is another special case of the generalized linear model, where $\Phi(t) = \log(1 + \exp(t)), c(\sigma) = 1$ and $y_j^{(i)} \in \{0, 1\}$.

Our goal is to recover the unknown $\boldsymbol{\theta}_i^*$'s given the observations from m tasks. A general estimator for multi-task learning is based on minimizing the negative log likelihood, under the $\ell_{0,2}$ constraint on the parameter matrix $\boldsymbol{\Theta}$. This gives rise to:

$$\min_{\boldsymbol{\Theta} \in \mathbb{R}^{d \times m}} \mathcal{L}(\boldsymbol{\Theta}) \quad \text{subject to } \|\boldsymbol{\Theta}\|_{0,2} \leq s, \tag{2.1}$$

where s is a tuning parameter which controls the row sparsity of $\boldsymbol{\Theta}$, $\mathcal{L}(\boldsymbol{\Theta})$ is the sum of the negative log likelihood over all the tasks, which is given by

$$\mathcal{L}(\boldsymbol{\Theta}) = -\sum_{i=1}^{m} \big[1/(2n_i) \sum_{j=1}^{n_i} y_j^{(i)} \boldsymbol{\Theta}_{*i}^\top \mathbf{x}_j^{(i)} + \Phi(\boldsymbol{\Theta}_{*i}^\top \mathbf{x}_j^{(i)})\big], \tag{2.2}$$

where $\boldsymbol{\Theta}_{*i}$ is the i-th column of $\boldsymbol{\Theta}$. In particular, when $c(\sigma) = \sigma^2$ and $\Phi(t) = t^2$, the negative log likelihood function of the exponential family distribution in (2.2) reduces to the square loss function, which is shown as follows:

$$\mathcal{L}(\boldsymbol{\Theta}) = \sum_{i=1}^{m} \frac{1}{2n_i} \|\mathbf{y}^{(i)} - \mathbf{X}^{(i)} \boldsymbol{\Theta}_{*i}\|_2^2. \tag{2.3}$$

Note that the square loss function has been used in Obozinski et al. (2006); Argyriou et al. (2008); Negahban and Wainwright (2008); Lounici et al. (2009); Zhang et al. (2010) for simplicity.

In addition, the optimization problem in (2.1) is nonconvex, because the constraint set $\|\boldsymbol{\Theta}\|_{0,2} \leq s$ is nonconvex. In fact, due to the combinatorial nature of this constraint, finding its global optimal solution is actually NP-hard. In the next subsection, we will propose a greedy algorithm to solve (2.1) approximately, yet with provable guarantee.

2.2 THE PROPOSED ALGORITHM

In order to get a good estimation of $\boldsymbol{\Theta}^*$, a vital problem is to get the row support of $\boldsymbol{\Theta}^*$. In order to get a good estimation of the row support of $\boldsymbol{\Theta}^*$, we designed the following algorithm. The formal description of the algorithm is summarized in Algorithm 1. In detail, we use the forward-backward strategy (Zhang, 2009) to select the feature set iteratively. In particular, we start from an empty feature set. The proposed algorithm adds the feature that will decrease the loss function most greatly into the current selected feature set in each iteration (The "Forward" strategy). Since the ℓ_2 norm of each row of the gradient characterize the decrease rate of the loss function, we use the row with largest ℓ_2 norm which is indexed by $i^{(t+1)}$ during the forward step (The seventh line of Algorithm 1).After each feature set updating, we also update the coefficients for each feature just as an ordinary regression problem. However, the forward strategy is too greedy because it only permits the entrance of new features but prohibits the deletion of irrelevant features. Hence we introduce the "Backward" mechanism to help the algorithm get rid of the bad local optima. That is, in each iteration, we not only add a new feature but also remove one or more irrelevant features from the feature set. The goodness of a feature is measured by the increase of the loss function when the feature is removed from the feature set. Similarly, the coefficient of each selected features is updated when there is a modification of feature set.

Algorithm 1 Forward Backward Greedy Algorithm for Multi-Task Learning (MultiFoBa)

1: **Require:** $\epsilon > 0$
2: **Initialize:** $\boldsymbol{\Theta}^{(0)} = 0, t = 0, F^{(0)} = \emptyset$
3: **while** TRUE **do**
4: **if** $\|\nabla \mathcal{L}(\boldsymbol{\Theta}^{(t)})\|_{\infty,2} < \epsilon$ **then**
5: break
6: **end if**
7: $i^{(t+1)} = \operatorname{argmax}_{i \notin F^{(t)}} \left\|\nabla \mathcal{L}(\boldsymbol{\Theta}^{(t)})_i\right\|_2$
8: $F^{(t+1)} = F^{(t)} \cup \{i^{(t+1)}\}$
9: $\boldsymbol{\Theta}^{(t+1)} = \operatorname{argmin}_{\boldsymbol{\Theta}_{i*}=\mathbf{0}, i \notin F^{(t+1)}} \mathcal{L}(\boldsymbol{\Theta})$
10: $\delta^{(t+1)} = \mathcal{L}(\boldsymbol{\Theta}^{(t)}) - \mathcal{L}(\boldsymbol{\Theta}^{(t+1)})$
11: $t = t + 1$
12: **while** TRUE **do**
13: **if** $\min_{i \in F^{(t)}} \mathcal{L}(\boldsymbol{\Theta}^{(t)} - \mathbf{e}_i \boldsymbol{\Theta}_{i*}^{(t)}) - \mathcal{L}(\boldsymbol{\Theta}^{(t)}) \geq \delta^{(t)}/2$ **then**
14: break
15: **end if**
16: $i^{(t)} = \operatorname{argmin}_{i \in F^{(t)}} \mathcal{L}(\boldsymbol{\Theta}^{(t)} - \mathbf{e}_i \boldsymbol{\Theta}_{i*}^{(t)})$
17: $F^{(t-1)} = F^{(t)} \setminus \{i^{(t)}\}$
18: $\boldsymbol{\Theta}^{(t-1)} = \operatorname{argmin}_{\boldsymbol{\Theta}_{i*}=\mathbf{0}, i \notin F^{(t-1)}} \mathcal{L}(\boldsymbol{\Theta})$
19: $t = t - 1$
20: **end while**
21: **end while**
22: **Output:** $\boldsymbol{\Theta}^{(t)}$

Note that we only delete those features which cause an increase of loss function by less than $\delta^{(t)}/2$, where $\delta^{(t)}$ is the decrease of the loss function when the last feature is added. This guarantees that the loss function will not increase when the cardinality of the feature set returns back. And this implies that the algorithm will not stuck in an infinite loop. In contrast to the forward feature selection algorithm, our algorithm employs a backward feature elimination step, which is able to help avoid the local optima. As we will show in the next section, the backward step is essential in achieving the model selection consistency under mild conditions.

We now analyze the time complexity of the algorithm.

The main computational overhead in the "forward" step is the 9-th line in Algorithm 1. For square loss, there exist a closed-form solution to the optimization problem in this line. In detail, for the i-th task, the solution is given by $\mathbf{\Theta}_{*i}^{(t)} = \left(\mathbf{X}_{F^{(t)}}^{(i)\top}\mathbf{X}_{F^{(t)}}^{(i)}\right)^{-1}\mathbf{X}_{F^{(t)}}^{(i)\top}\mathbf{y}^{(i)}$. The time complexity is $O(|F^{(t)}|^2\sum_{i=1}^m n_i)$ in the t-th iteration of the loop. Similarly, in the "backward" step, the main workload falls in the 18-th line and the time complexity is also $O(|F^{(t)}|^2\sum_{i=1}^m n_i)$. Since $|F^{(t)}|$ is much smaller than d, the computational cost is not expensive. Furthermore, we will prove in Section 3 that Algorithm 1 will terminate after finite steps. Therefore, the total time complexity of our algorithm is $O(|F^{(t)}|^2\sum_{i=1}^m n_i)$ times the number of steps. Overall, Algorithm 1 is efficient.

3 MAIN THEORETICAL RESULTS

In this section we will analyze the practicability of the algorithm and prove the finite-sample statistical rate of the proposed estimator. The detailed proofs of all the theory are deferred in the supplemental material.

For the ease of statistical analysis, we consider an oracle estimator $\widehat{\mathbf{\Theta}}_O$ which is obtained by restricting the row support of the estimator onto the row support of the unknown true parameter matrix $\mathbf{\Theta}^*$. More specifically, let $F(\mathbf{\Theta}) \subset \{1, 2, \ldots, d\}$ denote the index set of nonzero rows of $\mathbf{\Theta}$. Then $\widehat{\mathbf{\Theta}}_O$ is the optimal solution to the following problem:

$$\widehat{\mathbf{\Theta}}_O = \underset{\mathbf{\Theta}\in\mathbb{R}^{d\times m}}{\operatorname{argmin}} \mathcal{L}(\mathbf{\Theta}) \quad \text{subject to } F(\mathbf{\Theta}) = F(\mathbf{\Theta}^*). \tag{3.1}$$

Note that $\widehat{\mathbf{\Theta}}_O$ is not a practical estimator but a reference estimator used for theoretical analysis only. To simplify notation, let $F^* \equiv F(\mathbf{\Theta}^*) = F(\widehat{\mathbf{\Theta}}_O)$ and $F^{(t)} \equiv F(\mathbf{\Theta}^{(t)})$. We use $F^{(t)} - F^*$ to denote the set difference. For the matrix $\mathbf{\Theta}^{(t)}_{F^{(t)}-F^*} \in \mathbb{R}^{d\times m}$, its i-th row is the same as the i-th row of $\mathbf{\Theta}^{(t)}$ if $i \in F^{(t)} - F^*$. For $i \notin F^{(t)} - F^*$, the i-th row of $\mathbf{\Theta}^{(t)}_{F^{(t)}-F^*}$ is a zero vector. Note that according to this definition, $\mathbf{\Theta}^{(t)}_{F^{(t)}-F^*}$ is equal to $[\mathbf{\Theta}^{(t)} - \widehat{\mathbf{\Theta}}_O]_{F^{(t)}-F^*}$.

To concisely characterize the property of $\mathcal{L}(\mathbf{\Theta})$, we first introduce the definition of sparse eigenvalues, which is the extension of sparse eigenvalue for sparse regression (Zhang, 2009; Jalali et al., 2011; Liu et al., 2013; Rao et al., 2015). Similar extension has been used in Gong et al. (2013).

Definition 3.1 (Sparse Eigenvalues)**.** The smallest and largest s-sparse eigenvalues of $\nabla^2\mathcal{L}(\mathbf{\Theta})$ are

$$\rho_+(s) = \max_{1\le i\le m} \sup\big\{\mathbf{u}^\top\nabla^2_{\boldsymbol{\theta}_i}\mathcal{L}(\mathbf{\Theta})\mathbf{u} : \|\mathbf{u}\|_0 \le s, \|\mathbf{u}\|_2 = 1, \mathbf{\Theta}\in\mathbb{R}^{d\times m}\big\},$$
$$\rho_-(s) = \min_{1\le i\le m} \inf\big\{\mathbf{u}^\top\nabla^2_{\boldsymbol{\theta}_i}\mathcal{L}(\mathbf{\Theta})\mathbf{u} : \|\mathbf{u}\|_0 \le s, \|\mathbf{u}\|_2 = 1, \mathbf{\Theta}\in\mathbb{R}^{d\times m}\big\}.$$

Remark 3.2. The definition of $\rho_-(\cdot)$ is highly related to the definition of restricted strong convexity in Negahban et al. (2009). Previous studies (Zhang et al., 2009; Negahban et al., 2009) have shown that the assumption $\rho_-(s) > 0$ can be satisfied for different forms of $\mathcal{L}(\mathbf{\Theta})$. This is often referred to as sparse eigenvalue condition. For example, Zhang et al. (2009) proved that when the model is a linear regression model and $\mathcal{L}(\mathbf{\Theta})$ is a square loss, $\nabla^2_{\boldsymbol{\theta}_i}\mathcal{L}(\mathbf{\Theta})$ satisfies the sparse eigenvalue condition with high probability. Therefore, when we choose square loss in (2.3), it is easy to show that $\rho_-(s) > 0$ holds with high probability analogously. Another example is the generalized linear model. Negahban et al. (2009) proved that with high probability the loss function corresponding to generalized linear model satisfies the restricted strong convexity, which also implies that $\rho_-(s) > 0$.

Without loss of generality, we make the following assumption on the structure of the loss function $\mathcal{L}(\mathbf{\Theta})$.

Assumption 3.3 (Decomposable Loss Function)**.** The loss function can be decomposed into the sum of loss functions on different tasks. By formulation, we have

$$\mathcal{L}(\mathbf{\Theta}) = \sum_{i=1}^m \ell_i(\mathbf{\Theta}_{*i}),$$

where ℓ_i is the loss function defined on the i-th task.

Assumption 3.3 can be verified for many types of loss functions, including the loss functions in (2.2) and (2.3).

Combining Assumption 3.3 with the definition of sparse eigenvalues, it is easy to show that

$$\frac{\rho_-(s)}{2}\|\mathbf{\Delta}\|_F^2 \le \mathcal{L}(\mathbf{\Theta}+\mathbf{\Delta}) - \mathcal{L}(\mathbf{\Theta}) - \langle\nabla\mathcal{L}(\mathbf{\Theta}), \mathbf{\Delta}\rangle \le \frac{\rho_+(s)}{2}\|\mathbf{\Delta}\|_F^2, \quad \text{for all} \quad \|\mathbf{\Delta}\|_{0,2} \le s. \tag{3.2}$$

These two inequalities in (3.2) are frequently used in the proof in order to bound the difference between $\mathcal{L}(\mathbf{\Theta})$ and $\mathcal{L}(\mathbf{\Theta}+\mathbf{\Delta})$. In fact, it is highly related to the restricted strong convexity and smoothness condition proposed in Negahban et al. (2009). The key difference is here the inequality holds in the sparse subspace rather than a cone.

The first question we are going to address is whether and when Algorithm 1 will terminate. The following theorem guarantees that the proposed algorithm terminates in finite steps.

Theorem 3.4. Suppose that the loss function $\mathcal{L}(\mathbf{\Theta})$ satisfies Assumption 3.3. Let s be any integer satisfying $\rho_-(s) > 0$ and the following condition:

$$s \geq \big(s^* + 1\big)\bigg\{\bigg[\bigg(\sqrt{\frac{\rho_+(s)}{\rho_-(s)}} + 1\bigg)\frac{\sqrt{2}\rho_+(1)}{\rho_-(s)}\bigg]^2 + 1\bigg\}, \tag{3.3}$$

and take $\epsilon > 2\sqrt{2}\big\|\nabla\mathcal{L}(\widehat{\mathbf{\Theta}}_O)\big\|_{\infty,2}\ \rho_+(1)/\rho_-(s)$ in Algorithm 1. Then the algorithm terminates at some $t \leq s - s^*$.

Next we will introduce some theoretical results about the estimation error bounds of the output of Algorithm 1.

Theorem 3.5. Suppose that the loss function $\mathcal{L}(\mathbf{\Theta})$ satisfies Assumption 3.3. Let s be any integer satisfying (3.3) and $\rho_-(s) > 0$. Take

$$\epsilon > \frac{2\sqrt{2}\rho_+(1)}{\rho_-(s)}\ \|\nabla\mathcal{L}(\widehat{\mathbf{\Theta}}_O)\|_{\infty,2}, \tag{3.4}$$

then the output of Algorithm 1 satisfies

$$\big\|\mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*\big\|_F \leq \frac{2\sqrt{2}\epsilon}{\rho_-(s)}\sqrt{s_2^*} + \frac{2\sqrt{s^*}}{\rho_-(s^*)}\big\|[\nabla\mathcal{L}(\mathbf{\Theta}^*)]_{F^*}\big\|_{\infty,2}, \tag{3.5}$$

$$\frac{\rho_-(s)^2}{8\rho_+(1)^2}\big|F^{(t)} - F^*\big| \leq \big|F^* - F^{(t)}\big| \leq 2s_2^*, \tag{3.6}$$

where s_2^* is defined as

$$s_2^* := \big|\big\{i \in F^* - F^{(t)} : \big\|\mathbf{\Theta}_{i*}^*\big\|_2 < 2\sqrt{2}\epsilon/\rho_-(s) + \|[\widehat{\mathbf{\Theta}}_O - \mathbf{\Theta}^*]_{F^*}\|_{\infty,2}\big\}\big|. \tag{3.7}$$

Note in Theorem 3.5 that s_2^* denotes the number of nonzero rows in $\mathbf{\Theta}^*$ whose ℓ_2 norms are small. Those correspond to the rows which are difficult to recover. It is easy to verify that if $s_2^* = 0$, we have $|F^{(t)} - F^*| = |F^* - F^{(t)}| = 0$ by (3.6), which implies that $F^{(t)} = F^*$.

In the following corollary, we show that if the ℓ_2 norms of all the nonzero rows are sufficiently large, i.e., $s_2^* = 0$, we can achieve a sharper estimation error bound, together with model selection consistency.

Corollary 3.6. Under the same conditions as Theorem 3.5, if $s_2^* = 0$, i.e., the ℓ_2 norm of each row of $\widehat{\mathbf{\Theta}}_O$ is sufficiently large, then the estimation error of the output of Algorithm 1 is bounded by

$$\big\|\mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*\big\|_F \leq \frac{2\sqrt{s^*}}{\rho_-(s^*)}\big\|[\nabla\mathcal{L}(\mathbf{\Theta}^*)]_{F^*}\big\|_{\infty,2}, \tag{3.8}$$

and the model selection consistency can be obtained, i.e., $F^{(t)} = F^*$.

3.1 HIGH PROBABILITY RESULTS FOR SQUARE LOSS

In this subsection, we present the high probability result for a specific example, i.e., the square loss case. Similar high probability results can be proved for the general loss function in (2.2) with more involved arguments.

For the sake of simplicity, we assume that every task has the same number of observations, i.e., $n_1 = \ldots = n_m = n$. Then the square loss function in (2.3) can be further reduced to

$$\mathcal{L}(\mathbf{\Theta}) = \frac{1}{2n}\sum_{i=1}^{m}\|\mathbf{X}^{(i)}\boldsymbol{\theta}_i - \mathbf{y}^{(i)}\|_2^2. \tag{3.9}$$

Our analysis can be easily extended to the general square loss in (2.3) where different tasks may have different number of observations.

Without loss of generality, we make the following assumption on the design matrices $\mathbf{X}^{(i)}$'s.

Assumption 3.7. For all columns in $\mathbf{X}^{(i)}$, we have $\|\mathbf{X}_{*j}^{(i)}\|_2 \leq \sqrt{n}$, where $\mathbf{X}_{*j}^{(i)}$ is the j-th column of $\mathbf{X}^{(i)}$.

Note that Assumption 3.7 is often made in the analysis of Lasso estimator (Negahban et al., 2009; Zhang et al., 2009).

The estimation error bound of the output of Algorithm 1 is shown in the following theorem.

Theorem 3.8. Under the same conditions as Theorem 3.5, when the loss function is the square loss in (3.9) and satisfies Assumption 3.7, we have with probability at least $1 - 1/d - 2/s^*$ that

$$\|\mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*\|_F \leq \frac{10\rho_+(1)\sigma}{\rho_-^2(s)}\sqrt{\frac{s^*m}{n}} + \frac{16\rho_+(1)\sigma}{\rho_-^2(s)}\sqrt{\frac{s_2^*\log d}{n}} + \frac{4\sigma}{\rho_-(s^*)}\sqrt{\frac{s^*\log s^*}{n}}, \tag{3.10}$$

where s_2^* is defined as

$$s_2^* = \bigg|\bigg\{i \in F^* - F^{(t)} : \big\|\mathbf{\Theta}_{i*}^*\big\|_2 \leq \frac{9\rho_+(1)\sigma}{\rho_-^2(s)}\bigg(\sqrt{\frac{m}{n}} + 2\sqrt{\frac{\log d}{n}}\bigg)\bigg\}\bigg|.$$

Remark 3.9. Theorem 3.5 suggests that the statistical estimation rate of our algorithm is

$$O\bigg(\sqrt{\frac{s^*m}{n}} + \sqrt{\frac{s^*\log s^*}{n}} + \sqrt{\frac{s_2^*\log d}{n}}\bigg),$$

which is sharper than the statistical rate of convex relaxation based methods (Lounici et al., 2009; Obozinski et al., 2011), i.e., $O(\sqrt{s^*m/n} + \sqrt{s^*\log d/n})$, since s_2^* could be much smaller than s^*, and $\log s^*$ is much smaller than

$\log d$. From the sample complexity point of view, Theorem 3.5 implies the sample complexity of our algorithm is $O(s^*m + s^* \log s^* + s_2^* \log d)$. When s_2^* is sufficiently smaller than s^*, our sample complexity is tighter than the existing best sample complexity for group sparse signal recovery (Baraniuk et al., 2010; Rao et al., 2012), i.e., $O(s^* \log d + s^* m)$.

Corollary 3.10. Under the same conditions as Theorem 3.8, when $s_2^* = 0$, we have with probability at least $1 - 1/d - 2/s^*$ that

$$\|\mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*\|_F \leq \frac{10\rho_+(1)\sigma}{\rho_-^2(s)}\sqrt{\frac{s^*m}{n}} + \frac{4\sigma}{\rho_-(s^*)}\sqrt{\frac{s^* \log s^*}{n}},$$

and the model selection consistency can be obtained with probability at least $1 - 1/d - 2/s^*$, i.e., $F^{(t)} = F^*$.

Remark 3.11. From Corollary 3.10, we know that our algorithm can get a even faster convergence rate in terms of Frobenius norm as follows

$$O\left(\sqrt{\frac{s^*m}{n}} + \sqrt{\frac{s^* \log s^*}{n}}\right). \tag{3.11}$$

In addition, the sufficient condition for our algorithm to achieve model selection consistency is as follows

$$\left\|\mathbf{\Theta}^*_{i*}\right\|_2 \gtrsim \sqrt{\frac{m}{n}} + \sqrt{\frac{\log d}{n}} \quad \text{for all} \quad i \in F^* - F^{(t)}, \tag{3.12}$$

which is implied by $s_2^* = 0$. In other words, we need all non-zero rows of $\mathbf{\Theta}^*$ in the row index set $F^* - F^{(t)}$ are big enough in terms of ℓ_2 norm. This thanks to the $\ell_{0,2}$ constraint on the parameter matrix, which does not introduce bias when the nonzero rows of the parameter matrix are of large magnitude in terms of ℓ_2 norm. In fact, our analysis can be directly applied to the original forward backward algorithm in Zhang (2009), and delivers a sharper bound for single task sparse regression. This can be seen as a by-product of our technical contribution.

Remark 3.12. One may be curious that why our bound in (3.11) beats the minimax lower bound for group sparse recovery Lounici et al. (2011). Note that when $s_2^* = 0$, our algorithm fully recovers the support of $\mathbf{\Theta}^*$, and our estimator is identical to the multivariate regression estimator that is restricted on the true support F^* of the parameter matrix. In this case, our estimator reduces to a multivariate regression in the classical regime rather than in the high dimensional regime, i.e., the oracle estimator in (3.1). Therefore, the minimax lower bound that characterizes the information theoretic limit is no longer the one in the high dimensional regime Lounici et al. (2011), but the one for the multivariate regression in the classical regime, i.e., $O(\sqrt{s^*m/n})$. As we can see, the upper bound in (3.11) achieved by our algorithm matches the minimax lower bound of multivariate regression in the classical regime up to $\log(s^*)$. This is one of our major contributions in this paper and what we referred to as "faster rate".

Remark 3.13. It is interesting to compare our result in Corollary 3.10 with the main result in Gong et al. (2013). Gong et al. (2013) proposed multi-stage multi-task learning method and proved an estimation error bound as follows

$$\|\mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*\|_F \lesssim \sqrt{\frac{s^*m}{n}} + \sqrt{\frac{m \log d}{n}}. \tag{3.13}$$

The sufficient condition for their method to achieve model selection consistency is

$$\|\mathbf{\Theta}^*_{i*}\|_2 \gtrsim \sqrt{\frac{m(\log d + \log m)}{n}} \quad \text{for all} \quad i \in F^*. \tag{3.14}$$

By comparing (3.11) with (3.13), it is clear that our estimator attains a much shaper estimation error bound than Gong et al. (2013).

Furthermore, by comparing (3.12) with (3.14), we can see that the sufficient condition of model selection consistency for our algorithm is much milder than their method. In detail, our sufficient condition only takes into account those dimensions that fall in $F^* - F^{(t)}$ rather than the whole F^*. Moreover, the magnitude condition in (3.12) is also in a much smaller order than (3.14). This clearly demonstrates that the sufficient condition of the model selection consistency for our algorithm is substantially milder than Gong et al. (2013).

4 EXPERIMENTS

In this section, we conduct extensive empirical study on both synthetic and real-world datasets, to verify the effectiveness of the proposed method.

4.1 COMPARED ALGORITHMS

We present the empirical study by comparing the results of the following algorithms: **Lasso**: we apply Lasso (Tibshirani, 1996) to each task individually; **FoBa**: a forward backward algorithm for sparse regression (Zhang, 2009). Similar to Lasso, we apply FoBa to each task individually; **L1,2**: multi-task feature learning based on $\ell_{2,1}$-norm regularization (Liu et al., 2009); **MSMTFL**: the Multi-Stage Multi-Task learning method proposed by Gong et al. (2013); **DirtyMTL**: a dirty statistical model based multi-task learning algorithm with regularizer $\lambda_1\|\mathbf{P}\|_{1,1} + \lambda_2\|\mathbf{Q}\|_{1,\infty}$ ($\mathbf{\Theta} = \mathbf{P} + \mathbf{Q}$) (Jalali et al., 2013); **rRMTL**: a robust multi-task learning algorithm employing $\lambda_1\|\mathbf{P}\|_{2,1} + \lambda_2\|\mathbf{Q}^\top\|_{2,1}$ as the regularizer ($\mathbf{\Theta} = \mathbf{P} + \mathbf{Q}$) (Gong et al., 2012); **MultiFoBa**: This is our proposed algorithm, which employs the forward-backward strategy to select features under feature set cardinality constraint. We implement the proposed algorithm by MATLAB. For other algorithms, we use the implementation in

Table 1: The estimation error in terms of Frobenius norm of different algorithms on synthetic datasets.

	Lasso	FoBa	L1,2	MSMTFL	DirtyMTL	rMTFL	MultiFoBa
Dataset 1	7.61±0.38	5.79±0.39	1.59±0.10	1.80±0.09	6.30±0.19	1.89±0.46	**0.72±0.09**
Dataset 2	11.14±0.84	7.45±0.85	2.25±0.17	2.77±0.12	8.11±0.75	5.22±1.39	**1.04±0.09**
Dataset 3	11.35±0.59	9.27±0.59	3.12±0.49	3.36±0.21	8.56±0.74	6.29±1.01	**1.66±0.17**

Table 2: The F_1 scores of support recovery of different algorithms on synthetic datasets.

	Lasso	FoBa	L1,2	MSMTFL	DirtyMTL	rMTFL	MultiFoBa
Dataset 1	0.99±0.01	1.00±0.00	1.00±0.00	1.00±0.00	0.98±0.02	0.93±0.07	**1.00±0.00**
Dataset 2	0.60±0.11	1.00±0.00	0.98±0.02	0.99±0.01	0.86±0.11	1.00±0.00	**1.00±0.00**
Dataset 3	0.54±0.11	0.80±0.00	0.82±0.17	0.96±0.04	0.76±0.08	0.84±0.04	**0.95±0.02**

the software package MALSAR[1] In the experiments, the quadratic loss function in (3.9) is employed for all the compared algorithms. For MSMTFL, we use the capped-ℓ_1 regularizer. Note that the proposed algorithm has only one parameter ϵ, which controls the termination of the algorithm.

4.2 SYNTHETIC DATA

The synthetic data are generated by setting the number of tasks as m, where each task has n samples and of dimensionality d. Each sample is drawn from a multivariate normal distribution $N(0, \mathbf{I})$ where $\mathbf{I}$ is a $d \times d$ identity matrix. Then we normalize all columns of each data matrix $\mathbf{X}^{(i)} \in \mathbb{R}^{n\times d}$ to length one. Each entry of the underlying parameter matrix $\mathbf{\Theta}^*$ is sampled i.i.d. from the uniform distribution over the interval $[-10, 10]$. To simulate sparsity, we randomly set $d - s^*$ rows of $\mathbf{\Theta}^*$ to zero vectors. The response vector is generated by $\mathbf{y}^{(i)} = \mathbf{X}^{(i)}\boldsymbol{\theta}_i^* + \boldsymbol{\epsilon}^{(i)}$, where each entry of $\boldsymbol{\epsilon}^{(i)}$ is drawn i.i.d. from the normal distribution $N(0, \sigma_i^2)$. We choose $\sigma_i = 0.1$ for all i. In detail, we generate two synthetic datasets as follows. The parameter settings are $d = 256, m = 10, s^* = 5, n = 100$ for "Dataset 1"; and $d = 512, m = 10, s^* = 10, n = 100$ for "Dataset 2". In addition, we generate a more challenging synthetic dataset ("Dataset 3") to test the support recovery ability of different algorithms when there are nonzero rows with small ℓ_2 norm in $\mathbf{\Theta}^*$. "Dataset 3" is generated differently. Firstly, we generate a $d \times m$ matrix in which each element is sampled i.i.d. from the uniform distribution in the interval $[-10, 10]$. Then we randomly set $d - s^*$ rows as zero vectors. Among the other s^* nonzero rows, we randomly select s_w^* rows and divide each element in these rows by 20 to simulate the small norm. Other procedures of the data generation are the same as "Dataset 1" and "Dataset 2". We set $d = 512, m = 10, n = 100, s^* = 15, s_w^* = 5$ for "Dataset 3".

All algorithms in the comparative study are employed to estimate $\widehat{\mathbf{\Theta}}$ given $\mathbf{X}^{(i)}$'s and $\mathbf{y}^{(i)}$'s. Since all the algorithms have one or several parameters, we tune the parameters by 5-fold cross validation on each synthetic data. The estimation error of the parameter matrix in terms of Frobenius norm $\|\mathbf{\Theta}^{(t)} - \mathbf{\Theta}^*\|_F$ is reported in Table 1.

In order to evaluate the support recovery results of different algorithms, we use F_1 score defined as follows

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},$$

where $\text{precision} = |\text{supp}(\mathbf{\Theta}^*) \cap \text{supp}(\widehat{\mathbf{\Theta}})|/|\text{supp}(\widehat{\mathbf{\Theta}})|$ and $\text{recall} = |\text{supp}(\mathbf{\Theta}^*) \cap \text{supp}(\widehat{\mathbf{\Theta}})|/|\text{supp}(\mathbf{\Theta}^*)|$. Note that for some algorithms (such as Dirty and rMTFL) they not only output the estimator $\widehat{\mathbf{\Theta}}$, but also output two intermediate estimators $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{Q}}$, where $\widehat{\mathbf{\Theta}} = \widehat{\mathbf{P}} + \widehat{\mathbf{Q}}$. By empirical study, we found that the estimator $\widehat{\mathbf{P}}$ is more suitable for support recovery than the estimator $\widehat{\mathbf{\Theta}}$ (i.e., $\widehat{\mathbf{P}}$ achieves higher F_1 score than $\widehat{\mathbf{\Theta}}$), because in these algorithms $\widehat{\mathbf{P}}$ is a sparse or row-sparse matrix. Hence for Dirty and rMTFL algorithms, we use $\widehat{\mathbf{P}}$ to evaluate the feature selection (i.e., support recovery) performance. The F_1 score of support recovery is reported in Table 2.

From Tables 1 and 2, it can be seen that when all nonzero rows are with large ℓ_2 norms (Dataset 1 and 2), our algorithm can exactly recover the supports of $\mathbf{\Theta}^*$ and attain a small estimation error. While some other algorithms can also recover most supports (with a high F_1 score), they suffer from larger estimation error than ours. This is reasonable because our algorithm employs the support cardinality constraint, which is unbiased when the supports are recovered correctly. In contrast, many other algorithms employ some kinds of convex penalties, which lead to biased estimators. In order to recover the support correctly, they have to use a large penalty parameter λ, which makes the estimators more biased.

When there are nonzero rows with small ℓ_2 norm (Dataset 3), our algorithm can still recover the support of $\mathbf{\Theta}^*$ with high accuracy. This is consistent with our theory. In con-

[1] https://github.com/jiayuzhou/MALSAR

Table 3: The nMSE of different algorithms on school dataset.

	Lasso	FoBa	L1,2	MSMTFL	DirtyMTL	rMTFL	MultiFoBa
20%	0.903±0.012	0.832±0.009	0.924±0.022	0.804±0.009	0.803±0.009	0.802±0.010	**0.762±0.022**
30%	0.859±0.014	0.766±0.013	0.911±0.043	0.765±0.009	0.749±0.009	0.750±0.008	**0.727±0.027**

Table 4: The nMSE of different algorithms on SARCOS dataset.

	Lasso	FoBa	L1,2	MSMTFL	DirtyMTL	rMTFL	MultiFoBa
50	0.093±0.035	0.083±0.012	0.073±0.012	0.077±0.011	0.082±0.037	0.072±0.010	**0.067±0.010**
100	0.075±0.013	0.055±0.006	0.055±0.010	0.053±0.005	0.071±0.038	0.050±0.005	**0.045±0.003**
150	0.068±0.034	0.051±0.005	0.049±0.006	0.047±0.002	0.063±0.037	0.044±0.002	**0.040±0.001**

trast, the other algorithms achieve even worse recovery results when nonzero rows with small ℓ_2 norms exist.

4.3 REAL DATA

We use the School data[2] and the SARCOS data[3] to verify the effectiveness of the proposed algorithm on real datasets.

The School dataset consists of information of students from 139 secondary schools, as well as their exam scores. Each student is described by their 27 attributes, such as gender and ethnic group. The student exam score predicting problem can be cast as a multi-task regression problem: each school is considered as a task, each task as different number of data points, the attributes of students are input variables and their scores are responses. We randomly choose 20% and 30% samples from each task to form the training set and the rest samples as the test set. We tune the parameters of all the algorithms by 5-fold cross validation on the training data. We use the normalized Mean Square Error (nMSE), i.e., the mean squared error divided by the variance of ground-truth output, to measure the performance of all algorithms. Experiment results averaged over 20 repetitions are reported in Table 3.

The SARCOS data is collected for an inverse dynamic prediction problem for a anthropomorphic arm with 7 degrees of freedom. The data contains the training part and the testing part. The training part consists of 44,484 samples and the testing part 4,449 samples. Each sample is described by 21 attributes such as joint positions and velocities. There are also 7 responses attached to each sample, representing 7 torques. Our goal is predicting the responses based on the attributes. This problem can be casted as a multi-task regression problem, where the prediction of each response is regarded as a task, and all tasks share the same design matrix. We randomly choose 50, 100 and 150 samples from the training data of the original dataset to form 3 training sets and accordingly select 2000 samples from the testing data of the original dataset to form 3 testing sets. The experiment results averaged over 20 repetitions are summarized in Table 4.

From both Table 3 and Table 4, we can observe that the proposed algorithm outperforms the other algorithms greatly under different training/test splits on both datasets. This is due to the unbiased property of our estimator under mild conditions, as well as the faster statistical rate of our proposed estimator.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a general estimator for multi-task learning with row sparsity constraint on the parameter matrix. In order to solve it, we develop a forward backward greedy algorithm, whose output attains a sharper estimation error bound than many state-of-the-art multi-task learning methods. Moreover, the output of the proposed greedy algorithm enjoys model selection consistency under a mild condition. Thorough experiments on both synthetic and real-world data back up our theory.

We notice that the $\ell_{0,2}$ constrained nonconvex optimization problem in (2.1) can be potentially solved by the extensions of iterative hard thresholding (Jain et al., 2014) and Frank-Wolfe algorithms (Jaggi, 2013; Lacoste-Julien and Jaggi, 2013). We will investigate these algorithms in the future.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. Research was sponsored by Quanquan Gu's startup funding at Department of Systems and Information Engineering, University of Virginia.

References

ANDO, R. K. and ZHANG, T. (2005). A framework for learning predictive structures from multiple tasks and

[2] http://ttic.uchicago.edu/~argyriou/code/

[3] http://www.gaussianprocess.org/gpml/data/

unlabeled data. *The Journal of Machine Learning Research* **6** 1817–1853.

ARGYRIOU, A., EVGENIOU, T. and PONTIL, M. (2008). Convex multi-task feature learning. *Machine Learning* **73** 243–272.

BAKKER, B. and HESKES, T. (2003). Task clustering and gating for bayesian multitask learning. *The Journal of Machine Learning Research* **4** 83–99.

BARANIUK, R. G., CEVHER, V., DUARTE, M. F. and HEGDE, C. (2010). Model-based compressive sensing. *Information Theory, IEEE Transactions on* **56** 1982–2001.

BAXTER, J. (2000). A model of inductive bias learning. *J. Artif. Intell. Res.(JAIR)* **12** 149–198.

BOUCHERON, S., LUGOSI, G. and MASSART, P. (2013). *Concentration inequalities: A nonasymptotic theory of independence*. OUP Oxford.

CARUANA, R. (1997). Multitask learning. *Machine learning* **28** 41–75.

EVGENIOU, T. and PONTIL, M. (2004). Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

FAN, J. and LI, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association* **96** 1348–1360.

GONG, P., YE, J. and ZHANG, C. (2012). Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

GONG, P., YE, J. and ZHANG, C. (2013). Multi-stage multi-task feature learning. *The Journal of Machine Learning Research* **14** 2979–3010.

HAN, L. and ZHANG, Y. (2015). Learning multi-level task groups in multi-task learning .

JACOB, L., VERT, J.-P. and BACH, F. R. (2009). Clustered multi-task learning: A convex formulation. In *Advances in neural information processing systems*.

JAGGI, M. (2013). Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*.

JAIN, P., TEWARI, A. and KAR, P. (2014). On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*.

JALALI, A., JOHNSON, C. C. and RAVIKUMAR, P. K. (2011). On learning discrete graphical models using greedy methods. In *Advances in Neural Information Processing Systems*.

JALALI, A., RAVIKUMAR, P. and SANGHAVI, S. (2013). A dirty model for multiple sparse regression. *Information Theory, IEEE Transactions on* **59** 7947–7968.

JAVANMARD, A. and MONTANARI, A. (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research* **15** 2869–2909.

KANG, Z., GRAUMAN, K. and SHA, F. (2011). Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*.

KIM, S. and XING, E. P. (2010). Tree-guided group lasso for multi-task regression with structured sparsity .

LACOSTE-JULIEN, S. and JAGGI, M. (2013). An affine invariant linear convergence analysis for frank-wolfe algorithms. *arXiv preprint arXiv:1312.7864* .

LAWRENCE, N. D. and PLATT, J. C. (2004). Learning to learn with the informative vector machine. In *Proceedings of the twenty-first international conference on Machine learning*. ACM.

LIU, J., FUJIMAKI, R. and YE, J. (2013). Forward-backward greedy algorithms for general convex smooth functions over a cardinality constraint. *arXiv preprint arXiv:1401.0086* .

LIU, J., JI, S. and YE, J. (2009). Multi-task feature learning via efficient l 2, 1-norm minimization. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press.

LOUNICI, K., PONTIL, M., TSYBAKOV, A. B. and VAN DE GEER, S. (2009). Taking advantage of sparsity in multi-task learning. *arXiv preprint arXiv:0903.1468* .

LOUNICI, K., PONTIL, M., VAN DE GEER, S. and TSYBAKOV, A. B. (2011). Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics* 2164–2204.

NEGAHBAN, S. and WAINWRIGHT, M. J. (2008). Joint support recovery under high-dimensional scaling: Benefits and perils of l1,-regularization. *Advances in Neural Information Processing Systems* **21** 1161–1168.

NEGAHBAN, S., YU, B., WAINWRIGHT, M. J. and RAVIKUMAR, P. K. (2009). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*.

NESTEROV, Y. (2004). *Introductory lectures on convex optimization*, vol. 87. Springer Science & Business Media.

OBOZINSKI, G., TASKAR, B. and JORDAN, M. (2006). Multi-task feature selection. *Statistics Department, UC Berkeley, Tech. Rep* .

OBOZINSKI, G., WAINWRIGHT, M. J. and JORDAN, M. I. (2011). Support union recovery in high-dimensional multivariate regression. *The Annals of Statistics* 1–47.

PARAMESWARAN, S. and WEINBERGER, K. Q. (2010). Large margin multi-task metric learning. In *Advances in neural information processing systems*.

QUADRIANTO, N., PETTERSON, J., CAETANO, T. S., SMOLA, A. J. and VISHWANATHAN, S. (2010). Multitask learning without label correspondences. In *Advances in Neural Information Processing Systems*.

RAO, N., SHAH, P. and WRIGHT, S. (2015). Forward–backward greedy algorithms for atomic norm regularization. *Signal Processing, IEEE Transactions on* **63** 5798–5811.

RAO, N. S., RECHT, B. and NOWAK, R. D. (2012). Universal measurement bounds for structured sparse signal recovery. In *International Conference on Artificial Intelligence and Statistics*.

SCHWAIGHOFER, A., TRESP, V. and YU, K. (2004). Learning gaussian process kernels via hierarchical bayes. In *Advances in Neural Information Processing Systems*.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

TURLACH, B. A., VENABLES, W. N. and WRIGHT, S. J. (2005). Simultaneous variable selection. *Technometrics* **47** 349–363.

WAINWRIGHT, M. (2009). Sharp thresholds for noisy and high-dimensional recovery of sparsity using 1-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory* **55** 2183–2202.

YANG, X., KIM, S. and XING, E. P. (2009). Heterogeneous multitask learning with joint sparsity constraints. In *Advances in neural information processing systems*.

YU, K., TRESP, V. and SCHWAIGHOFER, A. (2005). Learning gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine learning*. ACM.

ZHANG, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics* 894–942.

ZHANG, C.-H. and HUANG, J. (2008). The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics* 1567–1594.

ZHANG, J., GHAHRAMANI, Z. and YANG, Y. (2005). Learning multiple related tasks using latent independent component analysis. In *Advances in neural information processing systems*.

ZHANG, T. (2009). Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Advances in Neural Information Processing Systems*.

ZHANG, T. ET AL. (2009). Some sharp performance bounds for least squares regression with l1 regularization. *The Annals of Statistics* **37** 2109–2144.

ZHANG, Y. and YEUNG, D.-Y. (2012). A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536* .

ZHANG, Y. and YEUNG, D.-Y. (2013). Learning high-order task relationships in multi-task learning. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press.

ZHANG, Y., YEUNG, D.-Y. and XU, Q. (2010). Probabilistic multi-task feature selection. In *Advances in neural information processing systems*.

ZHAO, P. and YU, B. (2006). On model selection consistency of lasso. *The Journal of Machine Learning Research* **7** 2541–2563.

ZHOU, J., YUAN, L., LIU, J. and YE, J. (2011). A multi-task learning formulation for predicting disease progression. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.