Adaptive Multi-task Sparse Learning with an Application to fMRI Study

Xi Chen * Jinghui He † Rick Lawrence† Jaime G. Carbonell*

Abstract
In this paper, we consider the multi-task sparse learning problem under the assumption that the dimensionality diverges with the sample size. The traditional l_1/l_2 multi-task lasso does not enjoy the oracle property unless a rather strong condition is enforced. Inspired by adaptive lasso, we propose a multi-stage procedure, adaptive multi-task lasso, to simultaneously conduct model estimation and variable selection across different tasks. Motivated by adaptive elastic-net, we further propose the adaptive multi-task elastic-net by adding another quadratic penalty to address the problem of collinearity. When the number of tasks is fixed, under weak assumptions, we establish the asymptotic oracle property for the proposed adaptive multi-task sparse learning methods including both adaptive multi-task lasso and elastic-net. In addition to the desirable asymptotic property, we show by simulations that adaptive sparse learning methods also achieve much improved finite sample performance. As a case study, we apply adaptive multi-task elastic-net to a cognitive science problem, where one wants to discover a compact semantic basis for predicting fMRI images. We show that adaptive multi-task sparse learning methods achieve superior performance and provide some insights into how the brain represents meanings of words.

1 Introduction

The traditional learning problem can often be cast to the estimation of a function $f : \mathcal{X} \mapsto \mathcal{Y}$, where $\mathcal{X} \in \mathbb{R}^p$ is the input space and $\mathcal{Y} \in \mathbb{R}$ is the output space. For many applications, the entire learning task can often be divided into several sub-tasks. When sub-tasks are related, it can be advantageous to learn all tasks simultaneously instead of learning each task independently. More formally, given K related tasks, the objective of *multi-task learning* [23, 5] is to *jointly* estimate K functions $f^{(k)} : \mathcal{X}^{(k)} \mapsto \mathcal{Y}^{(k)}$ for $1 \leq k \leq K$. Multi-task learning has been applied to many practical problems, including computer vision [21], natural language processing [1], computational biology [19] and neuroscience [11]. In multi-task learning, the basic assumption to be made is how different tasks are related to each other. Popular ways of modeling the relatedness include assuming that all tasks share a common latent feature representation [2] (e.g., sparsity-pattern); or parameters for different tasks are close to each other [7] or share a common prior [25]. We also note that when different tasks share the same input space but different output spaces, the corresponding learning problem is often referred to as *multi-response* learning, which can be viewed as a special case of multi-task learning.

For high-dimensional data, variable selection is of great importance to improve both prediction accuracy and model interpretability. The task of conducting variable selection can always be achieved via learning the sparsity pattern of parameters. In the multi-task learning setting, it is often assumed that parameters for different tasks share the same sparsity pattern [2, 18]. To achieve such an effect, a popular approach is to adopt a joint sparsity regularization to encourage group-wise sparsity across multiple tasks. In particular, one can adopt the l_1/l_q mixed-norm penalty with $q > 1$ [26, 19, 16]. Given K tasks, the l_1/l_q mixed-norm penalty is defined as:

$$\lambda_1 \sum_{j=1}^{p} \|\boldsymbol{\beta}_j\|_q, \tag{1.1}$$

where $\boldsymbol{\beta}_j = (\beta_j^{(1)}, \beta_j^{(2)}, \ldots, \beta_j^{(K)}) \in \mathbb{R}^K$ is the coefficient vector to be estimated for the j-th variable, λ_1 is a positive regularization parameter and p denotes the dimensionality of the input space. In this paper, we focus on the widely used l_1/l_2 mixed-norm penalty which encourages the joint sparsity pattern among different tasks.

The traditional l_1/l_2 mixed-norm penalty mainly suffers from two problems: (1) each l_2-norm on the coefficient vector shares the same amount of regularization (i.e., λ_1). This condition might be too restrictive for practical applications. A natural way to address this issue is to use a different weight w_j for the j-th variable, i.e., to define the penalty as $\lambda_1 \sum_{j=1}^{p} w_j \|\boldsymbol{\beta}_j\|_2$.

*School of Computer Science, Carnegie Mellon University
†IBM T.J. Watson Research Center

When there is a prior on the importance of each variable (e.g., extracted from biological domain knowledge as in [10] [1]), the weight w_j can be determined based on the prior knowledge. However, when the prior knowledge of the weight is unavailable, a natural question is how to automatically estimate w_j from the data. (2) From a statistical point-of-view, according to [8, 9], a good estimation procedure for sparse learning should have the following asymptotical oracle property: model selection consistency and asymptotic normality ($\sqrt{n}$-estimation consistency). When the dimensionality p diverges with the sample size, one fundamental limitation of the l_1/l_2-regularized multi-task lasso is that it does not have oracle property unless the design matrix satisfies a rather strong condition [15, 3].[2]

To address these problems, inspired by adaptive single-task lasso and its extensions [27, 24, 3, 29], we propose a multi-stage adaptive estimation procedure for multi-task sparse learning. More precisely, we first estimate the initial coefficients from the ordinary multi-task lasso. Then, we construct adaptive weights for each variable from the estimated coefficients. As the last step, final coefficients are estimated by another l_1/l_2-regularized multi-task lasso with the constructed adaptive weights. We establish the oracle property for the proposed *adaptive multi-task lasso.*

In addition, it is known that when the correlation between predictors is high, lasso leads to unstable variable selection performance. To address the problem of collinearity, Zou et al. [28] proposed the so-called elastic-net penalty by adding another quadratic penalty on top of the sparsity-inducing l_1 penalty. In this paper, we apply the elastic-net penalty to the multi-task learning setting by adding the quadratic penalty on top of l_1/l_2 mixed-norm penalty. We show that the proposed *adaptive multi-task elastic-net*, as a generalization of adaptive multi-task lasso, also achieves the oracle property under the assumption that the number of variables diverges with the sample size. In addition to the asymptotic property, we demonstrate via simulations that adaptive multi-task elastic-net leads to much better empirical performance for finite sample case. We note that the proof of oracle property for both adaptive lasso [27] and adaptive group lasso [24, 3] assumes that the dimensionality p is fixed and hence cannot be applied here. Our proof directly follows the proof for adaptive single-task elastic-net in [29].

As an important application, we apply adaptive multi-task learning methods to a cognitive neuroscience problem [14, 11], where we are interested in simultaneously predicting the functional magnetic resonance images (fMRI) from the presented word and selecting the corresponding semantic knowledge basis. We show that the proposed adaptive multi-task elastic-net achieves superior results.

The rest of this paper is organized as follows. In Section 2, we overview multi-task lasso and elastic-net penalty. In Section 3, we propose the adaptive multi-task learning algorithm. In Section 4, we discuss computational issues. In Section 5, we establish the oracle property of the proposed adaptive multi-task learning methods. In Section 6, we present numerical results on both simulated and real fMRI datasets. We conclude the paper in Section 7 with a discussion of possible future work.

2 Background

In this section, we introduce the background of the multi-task lasso. Consider a K-task linear regression model:

$$\begin{aligned} y^{(1)} &= X^{(1)}(\boldsymbol{\beta}^{(1)})^* + \boldsymbol{\epsilon}^{(1)} \\ y^{(2)} &= X^{(2)}(\boldsymbol{\beta}^{(2)})^* + \boldsymbol{\epsilon}^{(2)} \\ &\vdots \\ y^{(K)} &= X^{(K)}(\boldsymbol{\beta}^{(K)})^* + \boldsymbol{\epsilon}^{(K)}, \end{aligned} \tag{2.2}$$

where for each task $k = 1, \ldots, K$, let $X^{(k)}$ be the prescribed $n^{(k)} \times p$ design matrix, $(\boldsymbol{\beta}^{(k)})^*$ the true regression coefficients, $y^{(k)}$ the $n^{(k)}$-dimensional outputs and $\boldsymbol{\epsilon}^{(1)}, \ldots, \boldsymbol{\epsilon}^{(K)}$ *i.i.d.* random noises. We assume that for each task k and dimension j, predictors are standardized to mean zero and l_2-norm one:

$$\sum_{i=1}^{n^{(k)}} x_{ij}^{(k)} = 0 \quad \text{and} \quad \sum_{i=1}^{n^{(k)}} (x_{ij}^{(k)})^2 = 1. \tag{2.3}$$

We further assume that the noise has mean 0 and variance σ^2, i.e., $\mathbb{E}(\epsilon_j^{(k)}) = 0$ and $\mathrm{Var}(\epsilon_j^{(k)}) = \sigma^2$.

For the notation simplicity, we re-write Eq. (2.2) in a more compact form:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\epsilon}, \tag{2.4}$$

where $\mathbf{y}$ and $\boldsymbol{\epsilon}$ are $\sum_{k=1}^K n^{(k)}$-dimensional random vectors formed by stacking $y^{(1)}, \ldots, y^{(K)}$ and $\boldsymbol{\epsilon}^{(1)}, \ldots, \boldsymbol{\epsilon}^{(K)}$.

[1] Although the work in [10] also follows the name "adaptive multi-task lasso", our work distinguishes from [10] in that we automatically construct the prior weights purely from the data instead of relying on any prior knowledge. The method in [10] defines the weight as a linear combination of the data features from the prior knowledge and jointly optimizes the regression parameters and linear combination coefficients. Hence, the optimization is not only computationally heavy but also has many local minima.

[2] The finite sample properties of l_1/l_2-regularized multi-task lasso have been studied in [13].

Similarly, $\boldsymbol{\beta}^*$ denotes the vector obtained by stacking $\{(\beta^{(1)})^*, \ldots, (\beta^{(K)})^*\}$. The design matrix $\mathbf{X}$ is a block diagonal matrix with $X^{(k)}$ being the k-th block.

Furthermore, we introduce $\boldsymbol{\beta}_j \equiv (\beta_j^{(k)} : k \in \{1, \ldots, K\})$ for $1 \le j \le p$, that is, the vector formed by the regression coefficients corresponding to the j-th variable and let $\boldsymbol{\beta}$ denote the vector obtained by stacking $\{(\boldsymbol{\beta}^{(1)}), \ldots, (\boldsymbol{\beta}^{(K)})\}$. The l_1/l_2 mixed-norm of $\boldsymbol{\beta}$ is defined as:

$$\|\boldsymbol{\beta}\|_{2,1} = \sum_{j=1}^{p} \|\boldsymbol{\beta}_j\|_2, \tag{2.5}$$

where $\|\boldsymbol{\beta}_j\|_2 \equiv \sqrt{\sum_{k=1}^{K} (\beta_j^{(k)})^2}$ has the effect to enforce the elements in $\boldsymbol{\beta}_j$ to achieve zeros simultaneously.

The multi-task lasso is formulated by minimizing the squared loss with the l_1/l_2 mixed-norm of $\boldsymbol{\beta}$:

$$\begin{aligned} \widehat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} \Big\{ \sum_{k=1}^{K} \|y^{(k)} - \sum_{j=1}^{p} \beta_j^{(k)} X_j^{(k)}\|_2^2 + \lambda_1 \sum_{j=1}^{p} \|\boldsymbol{\beta}_j\|_2 \Big\} \\ &\equiv \arg\min_{\boldsymbol{\beta}} \Big\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1 \sum_{j=1}^{p} \|\boldsymbol{\beta}_j\|_2 \Big\}. \end{aligned} \tag{2.6}$$

To address the problem of collinearity among variables, similar to single-task elastic-net [28], one can add another quadratic penalty $\sum_{j=1}^{p} \|\boldsymbol{\beta}_j\|_2^2$ on top of the l_1/l_2-regularization and the corresponding multi-task elastic-net can be formulated as:

$$\widehat{\boldsymbol{\beta}} = (1 + \lambda_2) \arg\min_{\boldsymbol{\beta}} \Big\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1 \sum_{j=1}^{p} \|\boldsymbol{\beta}_j\|_2 + \lambda_2 \|\boldsymbol{\beta}\|_2^2 \Big\}. \tag{2.7}$$

The motivation for the $(1 + \lambda_2)$-scaling is to correct the extra bias introduced by the quadratic penalty $\lambda_2 \sum_{j=1}^{p} \|\boldsymbol{\beta}_j\|_2^2$. The readers may refer to [28] for more details on this scaling parameter.

As we discussed in the introduction part, it is desirable to have different regularization weights $\{w_j\}_{j=1}^{p}$ for different variables. When there is no prior knowledge for constructing such weights, it is impractical to tune each w_j individually. Inspired by the adaptive single-task lasso [28] and adaptive single-task elastic-net [29], we propose our adaptive multi-task learning methods in the next section which use a data-driven method to automatically construct the regularization weights.

3 Adaptive Multi-task Sparse Learning

In this section, we present the proposed adaptive multi-task elastic-net in Algorithm 1. The algorithm has three stages. In the first stage, we estimate the initial regression coefficients $\widehat{\boldsymbol{\beta}}$ via the multi-task elastic-net with uniform weight for each variable. Then we construct the adaptive weights $\{\widehat{w}_j\}_{j=1}^{p}$ from the initial estimated coefficients $\widehat{\boldsymbol{\beta}}$ as in Eq. (3.9). As the last step, we obtain the final coefficients via the multi-task elastic-net with the adaptive weights $\{\widehat{w}_j\}_{j=1}^{p}$.

Algorithm 1 Adaptive Multi-task Elastic-Net

Input: Input and response for K tasks $\{y^{(k)}, X^{(k)}\}_{k=1}^{K}$, tuning parameters λ_1, λ_1^*, λ_2, and the predefined positive constant γ for constructing adaptive weights.

1. $$\widehat{\boldsymbol{\beta}} = (1+\lambda_2) \arg\min_{\boldsymbol{\beta}} \Big\{ \|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1 \sum_{j=1}^{p} \|\boldsymbol{\beta}_j\|_2 + \lambda_2 \|\boldsymbol{\beta}\|_2^2 \Big\}. \tag{3.8}$$
2. $$\widehat{w}_j = (\|\widehat{\boldsymbol{\beta}}_j\|_2)^{-\gamma}, \quad \text{for } j = 1, \ldots, p \tag{3.9}$$
3. $$\widehat{\boldsymbol{\beta}}^* = (1+\lambda_2) \arg\min_{\boldsymbol{\beta}} \Big\{ \|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1^* \sum_{j=1}^{p} \widehat{w}_j \|\boldsymbol{\beta}_j\|_2 + \lambda_2 \|\boldsymbol{\beta}\|_2^2 \Big\}. \tag{3.10}$$

Output: The final estimated coefficients $\widehat{\boldsymbol{\beta}}^*$.

We first note that if λ_2 is set to zero, this procedure reduces to *adaptive multi-task lasso*. Therefore, we can view adaptive multi-task lasso as a special case of adaptive multi-task elastic-net with $\lambda_2 = 0$.

We also note that for the ease of tuning parameters, Step 1 (Eq. (3.8)) and Step 3 (Eq. (3.10)) share the same regularization parameter λ_2 for the quadratic penalty. According to our practical experience, using different regularization parameters for the quadratic penalty has very limited improvement on the performance but makes the tuning process much more time-consuming. In addition, as we show in the next section, the (asymptotic) oracle property can be established without assuming two different λ_2s for Step 1 and 3. But for λ_1 and λ_1^*, both of them have to be tuned to guarantee empirical performance and statistical property. We will discuss the choice of the parameters and the constant γ in more details in Section 5.

4 Computation

As for the optimization problems in Step 1 and 3, due to the simple structure of l_1/l_2 mixed-norm penalty, the proximal operator associated with the l_1/l_2 penalty can be solved in a closed-form. Therefore, one can easily adopt the Nesterov's composite gradient methods [17] (e.g., fast iterative shrinkage thresholding algorithm

Algorithm 2 FISTA for solving Multi-task Elastic-Net with Adaptive Weights

Input: $\{X^{(k)}\}_{k=1}^K$, $\{y^{(k)}\}_{k=1}^K$, $\{\widehat{w}_j\}_{j=1}^p$, λ_1^*, λ_2.
Initialization: $\theta_0 = 1$, $\mathbf{v}^0 = \boldsymbol{\beta}^0$,
$L = 2\ \max_{k=1}^K \sigma_{\max}(X^{(k)}) + 2\lambda_2$
Iterate For $t = 0, 1, 2, \ldots$, until convergence of $\boldsymbol{\beta}^t$:

1. Compute $\nabla h(\mathbf{v}^t)$ according to (4.11).
2. Solve the proximal operator associated with the l_1/l_2 mixed norm penalty:

$$\begin{aligned}\boldsymbol{\beta}^{t+1} = \arg\min_{\boldsymbol{\beta}} \langle \boldsymbol{\beta}, \nabla h(\mathbf{v}^t)\rangle + \frac{L}{2}\|\boldsymbol{\beta} - \mathbf{v}^t\|_2^2 \\ +\lambda_1^* \sum_{j=1}^p \widehat{w}_j \|\boldsymbol{\beta}_j\|_2 \end{aligned} \tag{4.12}$$

3. Set $\theta_{t+1} = \frac{1+\sqrt{1+4\theta_t^2}}{2}$.
4. Set $\mathbf{v}^{t+1} = \boldsymbol{\beta}^{t+1} + \frac{\theta_t - 1}{\theta_{t+1}}(\boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t)$.

Output: $\widehat{\boldsymbol{\beta}}^* = (1+\lambda_2)\boldsymbol{\beta}^{t+1}$.

(FISTA) [4] or a variant of FISTA with line-search in [12]) to solve the corresponding optimization problems. For the purpose of completeness, we present the specialization of FISTA [4] for solving Step 3 (Step 1 can be viewed as a special case of Step 3) in Algorithm 2.

Let
$$h(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_2\|\boldsymbol{\beta}\|_2^2$$
be the smooth part of the objective function in Eq. (3.10) with gradient:

$$\nabla h(\boldsymbol{\beta}) = 2\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - 2\mathbf{X}^T\mathbf{y} + 2\lambda_2\boldsymbol{\beta}. \tag{4.11}$$

The Lipschitz constant L for $\nabla h(\boldsymbol{\beta})$ is defined as follows: for any $\boldsymbol{\beta}^1$ and $\boldsymbol{\beta}^2$, we always have $\|\nabla h(\boldsymbol{\beta}^1) - \nabla h(\boldsymbol{\beta}^2)\|_2 \le L\|\boldsymbol{\beta}^1 - \boldsymbol{\beta}^2\|_2$. The closed form of L can be easily derived:

$$\begin{aligned} L &= 2\ \sigma_{\max}(X) + 2\lambda_2, \\ &= 2\ \max_{k=1}^{K} \sigma_{\max}(X^{(k)}) + 2\lambda_2 \end{aligned}$$

where $\sigma_{\max}(X)$ is the maximum singular value of X.

The proximal operator in Eq. (4.12) can be solved in a closed form as shown in [6, 12]. More specifically, rewrite Eq. (4.12):

$$\boldsymbol{\beta}^{t+1} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta} - (\mathbf{v}^t - \frac{1}{L}\nabla h(\mathbf{v}^t))\|_2^2 + \frac{\lambda_1^*}{L}\sum_{j=1}^p \widehat{w}_j\|\boldsymbol{\beta}_j\|_2$$

Let $\boldsymbol{\alpha} = \mathbf{v}^t - \frac{1}{L}\nabla h(\mathbf{v}^t)$. Then we have:

$$\boldsymbol{\beta}_j^{t+1} = \begin{cases} (1 - \frac{\lambda_1^* \widehat{w}_j}{L\|\boldsymbol{\alpha}_j\|_2})\boldsymbol{\alpha}_j & \text{if} \quad \|\boldsymbol{\alpha}_j\|_2 > \frac{\lambda_1^*\widehat{w}_j}{L} \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{4.13}$$

According to [4], Algorithm 2 has a convergence rate of $O(\frac{1}{T^2})$, where T is the total number of iterations and the per-iteration complexity is $O(\min(p,n)\cdot p \cdot K)$.

We note that the computational cost for Step 3 is much cheaper than that for Step 1 since one only needs to conduct estimation on the variables selected from Step 1. More specially, recall that $\widehat{\boldsymbol{\beta}}$ is the initial sparse estimate obtained from Step 1, let $\widehat{\mathcal{A}} = \{j : \widehat{\boldsymbol{\beta}}_j \neq \mathbf{0}\}$ and $\widehat{\mathcal{A}}^c$ be the complement set of $\widehat{\mathcal{A}}$. For those $j \in \widehat{\mathcal{A}}^c$, $\widehat{w}_j = \infty$ and hence the final estimate $\widehat{\boldsymbol{\beta}}_j^* = \mathbf{0}$. Therefore, for Step 3, instead of solving the full problem, we can first set $\widehat{\boldsymbol{\beta}}^*_{\widehat{\mathcal{A}}^c} = \mathbf{0}$ and then estimate the remaining coefficients by:

$$\begin{aligned}\widehat{\boldsymbol{\beta}}^*_{\widehat{\mathcal{A}}} = (1+\lambda_2)\arg\min_{\boldsymbol{\beta}} \Big\{ \sum_{k=1}^K \|y^{(k)} - \sum_{j\in\widehat{\mathcal{A}}} \beta_j^{(k)} X_j^{(k)}\|_2^2 \\ +\lambda_1^* \sum_{j\in\widehat{\mathcal{A}}} \widehat{w}_j\|\boldsymbol{\beta}_j\|_2 + \lambda_2 \sum_{j\in\widehat{\mathcal{A}}} \|\boldsymbol{\beta}_j\|_2^2 \Big\}. \end{aligned} \tag{4.14}$$

When using Algorithm 2 to solve Eq.(4.14), the per-iteration complexity reduces to $O\left(\min(|\widehat{\mathcal{A}}|, n)\cdot|\widehat{\mathcal{A}}|\cdot K\right)$ as compared to $O(\min(p,n)\cdot p\cdot K)$ for solving the full problem. Since we often have $|\widehat{\mathcal{A}}| \ll p$, there is only a little extra computational cost for adaptive methods.

5 Statistical Property

In this section, we discuss the statistical property of adaptive multi-task elastic-net and lasso. Using the same proof technique for adaptive single-task elastic-net [29], we show that asymptotically adaptive multi-task elastic-net has the oracle property, that is, the estimated $\widehat{\boldsymbol{\beta}}^*$ satisfies model selection consistency and asymptotic normality.

We first introduce some necessary notations. We denote the Gram matrix of $\mathbf{X}$ by $\Psi = \frac{1}{n}\mathbf{X}^T\mathbf{X}$, which is a block-diagonal matrix with $\frac{1}{n}(X^{(k)})^T(X^{(k)})$ as its k-th block. Let $\mathcal{A}$ be the set of true relevant variable, i.e., $\mathcal{A} = \{j : \boldsymbol{\beta}_j^* \neq \mathbf{0}\}$ with $|\mathcal{A}| = p_0 < p$. Let $X_j^{(k)}$ be the j-th column of $X^{(k)}$ and $X_{\mathcal{A}}^{(k)}$ be the sub-matrix of $X^{(k)}$ with the indices of columns in $\mathcal{A}$. Then we define $\mathbf{X}_j$ to be the block diagonal matrix with $X_j^{(k)}$ as its k-th block, and $\mathbf{X}_{\mathcal{A}}$ with $X_{\mathcal{A}}^{(k)}$ as its k-th block. And we denote $\Sigma_{\mathcal{A}}$ as $\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}$. In addition, let $\boldsymbol{\beta}^*_{\mathcal{A}}$ be the p_0K sub-vector of $\boldsymbol{\beta}^*$ formed by stacking $\{(\beta_{\mathcal{A}}^{(1)})^*, \ldots, (\beta_{\mathcal{A}}^{(K)})^*\}$. For the notation simplicity, we assume that the sample size n for each task is the same.

To establish the oracle property under the fixed K scenarios, we make the following assumptions:

(A1) $\lambda_{\min}(\Psi) \geq b$, where b is a positive constant.

(A2) $p = O(n^\nu)$ for some $0 \leq \nu < 1$

(A3)

$$\lim_{n\to\infty} \frac{\max_{i=1}^n \sum_{k=1}^K \sum_{j=1}^p (x_{ij}^{(k)})^2}{n} = 0$$

(A4) There exists $\delta > 0$ such that for any task k and variable j: $\mathbb{E}(|\epsilon_j^{(k)}|^{2+\delta}) < \infty$

The first condition (A1) assumes the positive definiteness of the Gram matrix. The second one assumes that p can diverge with n while the last two assumptions are used for proving the asymptotic normality.

To establish the oracle property, we choose the fixed constant $\gamma > \frac{2}{1-\nu}$ for constructing the adaptive weights. The other parameters should be set according to the following conditions:

(B1)

$$\lim_{n\to\infty} \frac{\lambda_1}{\sqrt{n}} = 0; \quad \lim_{n\to\infty} \frac{\lambda_1^*}{\sqrt{n}} = 0; \quad \lim_{n\to\infty} \frac{\lambda_2}{\sqrt{n}} = 0;$$

(B2) In addition, let $\eta = \min_{j\in\mathcal{A}}(\|\boldsymbol{\beta}_j^*\|_2)$, we assume that λ_1^* and λ_2 satisfy the following conditions:

$$\lim_{n\to\infty} \frac{\lambda_2\|\boldsymbol{\beta}^*\|_2}{\sqrt{n}} = 0; \quad \lim_{n\to\infty} \left(\frac{n}{p\lambda_1^*}\right)^{\frac{1}{\gamma}} \eta = \infty$$

The oracle property which contains model selection consistency and asymptotic normality is stated in the next theorem:

THEOREM 5.1. *Let $\widehat{\boldsymbol{\beta}}^*$ be the estimator obtained from adaptive multi-task elastic-net in Algorithm 1, under the assumptions (A1)–(A4) and (B1)–(B2), we have*

1. *Let $\widehat{\mathcal{A}}^* = \{j : \widehat{\boldsymbol{\beta}}_j^* \neq \mathbf{0}\}$ be the set of estimated relevant variables,* $\Pr(\widehat{\mathcal{A}}^* = \mathcal{A}) \to 1$.

2. *There exists $\boldsymbol{\alpha}$ with $\|\boldsymbol{\alpha}\|_2 = 1$, such that*

$$\boldsymbol{\alpha}^T(\mathbf{I} + \lambda_2\Sigma_{\mathcal{A}}^{-1})\Sigma_{\mathcal{A}}^{1/2}\left(\widehat{\boldsymbol{\beta}}_{\mathcal{A}}^* - \boldsymbol{\beta}_{\mathcal{A}}^*\right)$$
$$\to_d N(0, (1+\lambda_2)^2\sigma^2),$$

where $\Sigma_{\mathcal{A}} = \mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}$.

We prove the oracle property by extending the proof for adaptive single-task elastic-net in [29] to the multi-task case. The proof of asymptotic normality is based on Lyapunov central limit theorem as in [29]. The detailed proof is presented in Appendix.

REMARK 1. *Since the assumptions involving λ_2 only include $\lim_{n\to\infty} \frac{\lambda_2}{\sqrt{n}} = 0$ and $\lim_{n\to\infty} \frac{\lambda_2\|\boldsymbol{\beta}^*\|_2}{\sqrt{n}} = 0$, adaptive multi-task lasso with $\lambda_2 = 0$ automatically satisfies these assumptions. Therefore, as a special case of adaptive multi-task elastic-net, adaptive multi-task lasso also enjoys the oracle property.*

6 Experiment

In this section, we demonstrate the performance of adaptive multi-task sparse learning methods by both simulated data and a fMRI case study.

6.1 Simulated Study We generate data from multi-task linear model with K tasks as in Eq.(2.2). More specially, each $X^{(k)}$ for $1 \leq k \leq K$ follows a p-dimensional standard multivariate Gaussian distribution. The true coefficients are $\boldsymbol{\beta}^* = (\beta_1^*, \ldots, \beta_{|\mathcal{A}|}^*, 0, \ldots, 0)^T$ where each β_j for $1 \leq j \leq |\mathcal{A}|$ is drawn from $N(3, 0.1^2)$. We compare multi-task lasso (lasso), multi-task elastic-net (enet), adaptive multi-task lasso (ada-lasso) and adaptive multi-task elastic-net (ada-enet). We set the sample size n for each task $n = 200$, $p = 4\lceil\sqrt{n}\rceil - 5 = 55$, $p_0 \equiv |\mathcal{A}| = \lceil p/3\rceil = 19$, $K = 5$ or $K = 10$ and the noise level $\sigma = 2$ or $\sigma = 4$. Since $\nu = 1/2$, we set $\gamma = \frac{2}{1-\nu} = 4$ according to the theory. According to our experience, the result is not very sensitive to the choice of λ_2 as long as it falls into a certain range. Therefore, for the ease of tuning parameters, we directly set $\lambda_2 = 1$ for elastic-net. We tune other parameters λ_1 and λ_1^* using the same sized held-out validation data generated in the same way as the training data.

For each method, we report the mean squared error (MSE) defined by $\mathbb{E}[\sum_{k=1}^K((\widehat{\boldsymbol{\beta}}^{(k)})^* - (\boldsymbol{\beta}^{(k)})^*)^T((\widehat{\boldsymbol{\beta}}^{(k)})^* - (\boldsymbol{\beta}^{(k)})^*)]$ and the variable selection performance. The variable selection performance is measured by precision defined by $|\widehat{\mathcal{A}}^* \cap \mathcal{A}|/|\widehat{\mathcal{A}}^*|$, recall defined by $|\widehat{\mathcal{A}}^* \cap \mathcal{A}|/|\mathcal{A}|$ and F1-score by $2 \cdot \text{precision} \cdot \text{recall}/(\text{precision} + \text{recall})$. In addition, we report the mean and standard deviation of each measure based on 100 runs in Table 1.

From Table 1, we make following interesting observations:

1. For all different settings of K and σ, adaptive methods outperform non-adaptive methods in both model fitting and selection. When the number of task K increases, the advantage of adaptive procedures becomes more apparent.

Table 1: Simulation study for $n = 200$, $p = 55$, $p_0 = 19$.

K	σ	Method	MSE	F1-score	Precision	Recall
5	2	lasso	5.895 (1.731)	0.781 (0.154)	0.731 (0.216)	0.904 (0.084)
		enet	5.512 (1.663)	0.738 (0.152)	0.658 (0.206)	0.915 (0.091)
		ada-lasso	2.604 (0.647)	0.815 (0.068)	0.828 (0.105)	0.817 (0.099)
		ada-enet	2.580 (0.617)	0.812 (0.072)	0.814 (0.105)	0.823 (0.095)
5	4	lasso	11.116 (2.914)	0.577 (0.069)	0.524 (0.142)	0.754 (0.222)
		enet	10.879 (2.787)	0.579 (0.066)	0.522 (0.140)	0.763 (0.223)
		ada-lasso	10.554 (3.074)	0.606 (0.079)	0.706 (0.100)	0.541 (0.097)
		ada-enet	10.551 (3.220)	0.613 (0.078)	0.705 (0.101)	0.554 (0.098)
10	2	lasso	7.761 (1.403)	0.719 (0.039)	0.564 (0.047)	0.998 (0.010)
		enet	7.583 (1.438)	0.665 (0.035)	0.499 (0.040)	0.999 (0.007)
		ada-lasso	2.391 (0.731)	0.902 (0.057)	0.865 (0.081)	0.948 (0.058)
		ada-enet	2.401 (0.727)	0.896 (0.058)	0.850 (0.086)	0.954 (0.055)
10	4	lasso	16.923 (2.771)	0.644 (0.051)	0.495 (0.050)	0.928 (0.064)
		enet	16.479 (2.808)	0.632 (0.047)	0.477 (0.047)	0.942 (0.053)
		ada-lasso	13.554 (4.142)	0.743 (0.103)	0.792 (0.099)	0.707 (0.126)
		ada-enet	13.374 (3.970)	0.746 (0.103)	0.794 (0.101)	0.712 (0.127)

Table 2: Simulation study for $n = 200$, $p = 400$, $p_0 = 134$.

K	σ	Method	MSE	F1-score	Precision	Recall
5	2	lasso	8.991 (0.786)	0.633 (0.019)	0.507 (0.018)	0.843 (0.030)
		enet	6.571 (1.432)	0.579 (0.049)	0.431 (0.095)	0.934 (0.085)
		ada-lasso	8.515 (1.143)	0.632 (0.046)	0.767 (0.048)	0.543 (0.068)
		ada-enet	6.895 (0.986)	0.651 (0.035)	0.639 (0.073)	0.681 (0.085)
5	4	lasso	16.078 (2.017)	0.547 (0.035)	0.492 (0.064)	0.656 (0.135)
		enet	13.765 (1.363)	0.506 (0.039)	0.603 (0.047)	0.437 (0.044)
		ada-lasso	12.270 (1.442)	0.488 (0.039)	0.621 (0.048)	0.404 (0.039)
		ada-enet	12.392 (1.572)	0.549 (0.028)	0.488 (0.075)	0.676 (0.158)
10	2	lasso	13.261 (1.250)	0.703 (0.030)	0.555 (0.033)	0.960 (0.017)
		enet	12.637 (0.968)	0.561 (0.008)	0.391 (0.008)	0.990 (0.009)
		ada-lasso	8.614 (1.075)	0.809 (0.027)	0.881 (0.024)	0.749 (0.044)
		ada-enet	6.790 (0.897)	0.798 (0.025)	0.765 (0.046)	0.838 (0.053)
10	4	lasso	20.924 (1.443)	0.629 (0.022)	0.514 (0.024)	0.810 (0.039)
		enet	19.082 (1.320)	0.591 (0.015)	0.437 (0.013)	0.912 (0.025)
		ada-lasso	18.776 (2.050)	0.654 (0.038)	0.744 (0.036)	0.584 (0.046)
		ada-enet	16.306 (1.891)	0.666 (0.036)	0.725 (0.039)	0.618 (0.042)

2. When the noise level is low ($\sigma = 2$), the performance of adaptive multi-task lasso and adaptive multi-task elastic-net are similar. For larger noise, adaptive multi-task elastic-net outperforms adaptive multi-task lasso.

3. In terms of variable selection performance, we observe that the recall for adaptive procedures is lower than that for non-adaptive ones but the precision is much higher, which leads to higher F1-score. This observation indicates that non-adaptive procedures tend to select an overly dense model, thus leading to high recall but very low precision.

Now we study a more challenging case for $p > n$. We set $p = 2n = 400$ and $p_0 = |\mathcal{A}| = \lceil p/3 \rceil = 134$ and repeat the above experiments. Although the theory does not directly apply to the case when p grows faster than n asymptotically, for this experiment, we still set γ to 4 as in the previous example. In fact, we tune γ in the range $\{1, 2, \ldots, 5\}$ and observe that the performance is insensitive with respect to γ. The results are presented in Table 2.

From Table 2, we can see that when $p > n$, adaptive multi-task elastic-net is still the best for most cases in terms of both model fitting and selection. When K is small, adaptive multi-task lasso could be worse than

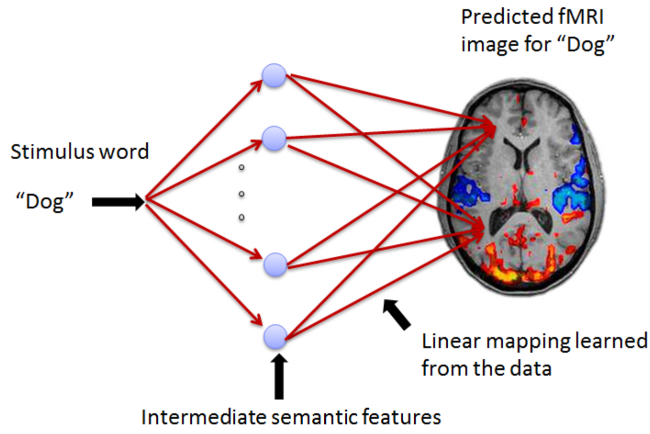

Figure 1: Model for predicting fMRI activation given a stimulus word

Table 3: The 60 stimulus words presented during the fMRI studies. Each row represents a category.

bear	cat	cow	dog	horse
arm	eye	foot	hand	leg
apartment	barn	church	house	igloo
arch	chimney	closet	door	window
coat	dress	pants	shirt	skirt
bed	chair	desk	dresser	table
ant	bee	beetle	butterfly	fly
bottle	cup	glass	knife	spoon
bell	key	refrigerator	telephone	watch
chisel	hammer	pliers	saw	screwdriver
carrot	celery	corn	lettuce	tomato
airplane	bicycle	car	train	truck

multi-task lasso or multi-task elastic-net. When we add another quadratic penalty (i.e., adaptive multi-task elastic-net), the performance will be greatly improved. For non-adaptive methods, when $p > n$, it is well known that adding the quadratic penalty leads to much better performance [28]. From our experiments, similar conclusions can also be drawn for adaptive methods.

6.2 Application to fMRI Study In this section, we present a case study of adaptive multi-task elastic-net by applying it to an important problem in cognitive neuroscience. Specifically, we consider the task of predicting a person's neural activity in response to an English word as described in [14, 11]. The goal is to predict the neural image recorded using *functional magnetic resonance imaging* (fMRI) when a person stares at and thinks about a given word. The experimental protocol is illustrated in Figure 1. In more details, given a stimulus word w, the first step encodes the meaning of w in terms of intermediate semantic features. The second step predicts the neural fMRI activation at each voxel [3] of the brain, as a sum of neural activations contributed by each of the intermediate semantic features. The training process uses a small number of words to learn a multi-task linear model that maps the intermediate semantic features to neural activation images where each task is defined by the activation at each voxel.

More specifically, the dataset contains 60 stimulus words which are composed of nouns from 12 categories with 5 exemplars per category. For example, a *bodypart* category includes Arm, Eye, Foot, Hand, Leg, a *tools* category includes the words Chisel, Hammer, Pliers, Saw, Screwdriver, and a *furniture* category includes Bed, Chair, Dresser, Desk, Table, etc. All the 60 words are presented in Table 3.

Then nine participants were presented with 60 different words and were asked to think about each word for several seconds while their neural activities were recorded. So that there are altogether $n = 60$ fMRI images taken for each participant[4]. A typical fMRI image contains activities in over 20,000 voxels. We select the top $K = 500$ voxel responses using the stability criterion score as described in [14]. By viewing the activation at each single voxel as a task, the output $y^{(k)}$ is the neural activation at the k-th voxel and there are in total 500 tasks.

As for the input, for each stimulus word, we adopt the semantic features from 218 questions as in [20][5]. These questions are related to the size, color, shape, property, usage of an object. Example questions include *IS IT BODY PART?* or *CAN YOU HOLD IT?*. Given a stimulus word, each question is rated from 1 to 5 (from definitely not to definitely yes). In other words, each stimulus word is mapped into a vector of length 218 which corresponds to the answers from 218 questions to this word. Therefore, in our problem, the design matrix $\mathbf{X}$ has $p = 218$ columns and is shared across all $K = 500$ tasks. These questions can be viewed as a set of sematic basis and the question which we try to answer in this experiment is: *What is the top 10 basis to best represent semantic meanings of the words from different categories* [6] *?*

[3] A voxel represents a 1-3 mm^3 volume in the brain and is the basic spatial unit of measurement in fMRI.

[4] Each image is actually the average of 6 different recordings.

[5] Our intermediate features are different from the ones used in [11].

[6] In addition to the top 10 basis, we also conduct experiments to select various numbers of basis. We observe that adaptive multi-

To automatically learn the semantic basis, we apply the proposed adaptive multi-task elastic-net on the fMRI data which can simultaneously predict the fMRI images and perform the basis selection. More specifically, our evaluation is based on the leave-two-out testing. For each trial, we select 2 words out of the 60 for testing and other 58 words for training. To evaluate the prediction performance, we convert this regression problem into a classification problem using the method in [14]. More specifically, let two testing images be y_1 and y_2, where each one is a 500×1 column vector and the predicted images be $\widehat{y}_1$ and $\widehat{y}_2$. If $\cos(y_1, \widehat{y}_1) + \cos(y_2, \widehat{y}_2) > \cos(y_1, \widehat{y}_2) + \cos(y_2, \widehat{y}_1)$, we say the prediction task for this trial is successful. We generate all $\binom{60}{2}$ possible pairs for 60 words (1,770 in total) and count the number of times that the joint labeling is correct. The accuracy is defined as the number of successes over 1770 trials.

For this experiment, lasso methods are always worse than the corresponding elastic-net methods. Therefore, we only compare adaptive multi-task elastic-net and multi-task elastic-net with λ_2 and γ set to one[7]. For multi-task elastic-net, we tune the regularization parameter so that 10 basis are selected. For adaptive multi-task elastic-net, λ_1 is tuned using leave-one-out cross validation on training set; while λ^* is tuned so that top 10 basis are included. In addition, there are in total 9 participants. Therefore, we have two choices of learning schemes. We can either treat each participant separately or combine fMRI from all participants (thereby yielding $500 \times 9 = 4500$ tasks). The comparison results are presented in Figure 2.

From Figure 2, we can see that for most participants, adaptive procedure significantly outperforms the non-adaptive procedure. The only exception is for the 3rd participant on the separated data and 4th participant on the combined data. The p-value of paired t-test between the results of adaptive and non-adaptive methods is $0.03884 < 0.05$ for the separated data and $0.008446 < 0.5$ for the combined data, which further indicates the adaptive method has the advantages over the non-adaptive method. From the box plot in Figure 2, we observe that although the median of the combined data does not have a notable improvement as compared to that of the separated data, the variance is much smaller. This indicates that the results obtained from the combined data are more stable [8].

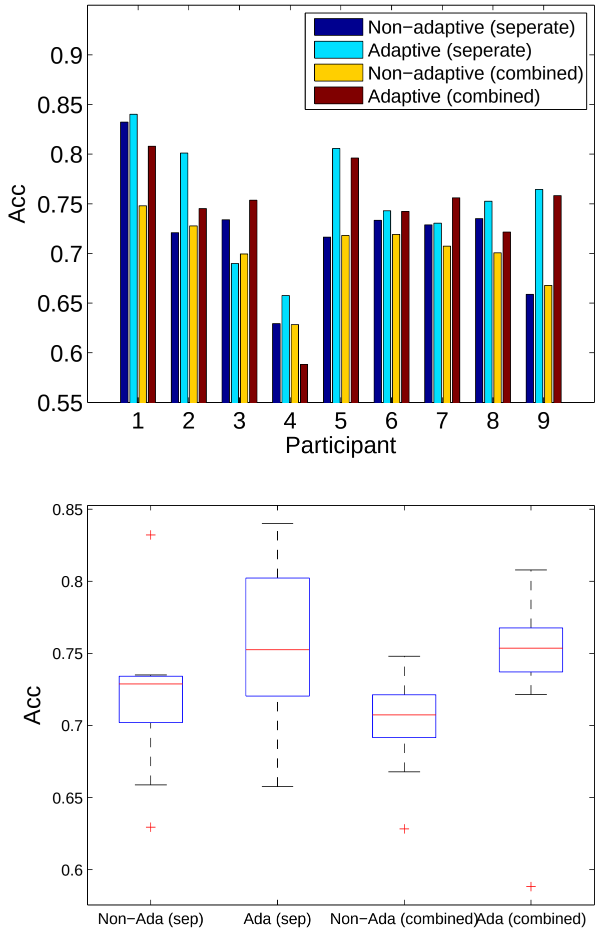

Figure 2: Bar and box plots for accuracies for 9 fMRI participants

In Table 4, we present one example of top 10 questions learned from adaptive multi-task elastic-net. As we can see, there is a close relationship between the selected semantic basis and 60 stimulus words. For example, *IS IT AN ANIMAL?* refers to words bear, cat, cow, dog, horse, ant, bee, beetle, butterfly, fly; *IS IT A BODY PART?* refers to arm, eye, foot, hand, leg; *IS IT MADE OF WOOD?* is related to the concept *furniture*, *IS IT MANMADE?* is related to many concepts, including *clothing*, *tools*, etc. Other interesting questions are related to the specific property of the objects, e.g., *CAN YOU EAT IT?* and *CAN*

task elastic-net always performs better than the non-adaptive methods. However, we omit the results due to space limitations.

[7] Similar to what we observe in the simulated study, the performance is insensitive to λ_2 and γ.

[8] The reported accuracies are lower than the ones in [20]. This is mainly because we learn a highly sparse model with only 10 semantic basis selected for the variable selection purpose; while [20] uses ridge regression, which utilizes all 218 features.

Table 4: An example of 10 learned semantic basis questions.

IS IT AN ANIMAL?
IS IT A BODY PART?
IS IT A BUILDING?
IS IT A BUILDING PART?
IS IT A TOOL ?
IS IT MANMADE?
CAN YOU EAT IT?
CAN YOU HOLD IT?
IS IT COLORFUL ?
DOES IT HAVE PARTS?

YOU HOLD IT?. We also point out that the correlated semantic basis *IS IT MANMADE?* and *IS IT A TOOL?* are selected simultaneously. It is mainly due to the "grouping effect" of the quadratic penalty in elastic-net which can simultaneously select highly correlated variables for the purpose of better interpretability.

7 Conclusion

In this paper, we propose adaptive multi-task lasso and elastic-net for multi-task sparse learning. Our methods can learn the regularization weight for each variable in a data-dependent manner and enjoy the asymptotic oracle property. We further apply the proposed method to an interesting fMRI study problem, which leads to superior performance in terms of predicting fMRI images from stimulus words.

As an immediate next step, we would like to apply the idea of adaptive learning to multi-task classification problems where the output space for each task is discrete. Theoretically, we would like to study the case where the number of tasks also goes to infinity with the sample size. In addition, we would like to explore another aspect of fMRI application: how to decode the stimulus word from a large set of possible words according to the recorded fMRI images.

8 Acknowledgement

We would like to thank Mark Palatucci for providing us the fMRI data that makes this experiment possible. We would also like to thank Mu Li and Qihang Lin for very helpful discussions. We thank anonymous reviewers for their constructive comments on improving the quality of the paper. Research was sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA) under the Anomaly Detection at Multiple Scales (ADAMS) program, Agreement Number W911NF-11-C-0200. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Defense Advanced Research Projects Agency or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

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

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–273, 2008.

[3] F. Bach. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 1179–1225:2008, 9.

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

[5] R. Caruana. Multitask learning. *Machine Learning Journal*, 28:41–75, 1997.

[6] J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, 2009.

[7] T. Evgeniou and M. Pontil. Regularized multitask learning. In *ACM SIGKDD*, 2004.

[8] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360, 2001.

[9] J. Fan and R. Li. Statistical challenges with high dimensionality: Feature selection in knowledge discovery. In *Proceedings of the Madrid International Congress of Mathematicians*, 2006.

[10] S. Lee, J. Zhu, and E. P. Xing. Adatpive multi-task lasso: with applications to eqtl detection. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[11] H. Liu, M. Palatucci, and J. Zhang. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *International Conference on Machine Learning*, 2009.

[12] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.

[13] K. Lounici, A. B. Tsybakov, M. Pontil, and S. A. V. D. Geer. Taking advantage of sparsity in multi-task learning. In *Conference on Learning Theory (COLT)*, 2009.

[14] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.-M. Chang, V. L. Malave, R. A. Mason, and M. A. Just. Predicting human brain activity associated with the meanings of nouns. *Science*, 320:1191, 2008.

[15] Y. Nardi and A. Rinaldo. On the asymptotic properties of the group lasso estimator for linear models. *Electronic Journal of Statistics*, 2:605–633, 2008.
[16] S. Negahban and M. J. Wainwright. Simultaneous support recovery in high dimensions: Benefits and perils of block ℓ_1/ℓ_∞-regularization. *IEEE Transactions on Information Theory*, 57 (6):3841–3863, 2011.
[17] Y. Nesterov. Gradient methods for minimizing composite objective function. Technical report, Universit catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007.
[18] G. Obozinski, B. Taskar, and M. I. Jordan. High-dimensional union support recovery in multivariate regression. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2008.
[19] G. Obozinski, B. Taskar, and M. I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20:231–252, 2010.
[20] M. Palatucci, D. Pomerleau, G. Hinton, and T. Mitchell. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.
[21] A. Quattoni, M. Collins, and D. Trevor. Transfer learning for image classification with sparse prototype representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
[22] R. Rockafellar. *Convex Analysis*. Princeton Univ. Press, 1996.
[23] S. Thrum and L. Pratt. *Learning to Learn*. Kluwer Academic Publishers, 1998.
[24] H. Wang and C. Leng. A note on adaptive group lasso. *Computational Statistics and Data Analysis*, 52:5277–5286, 2008.
[25] K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *International Conference on Machine Learning (ICML)*, 2005.
[26] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68:49–67, 2006.
[27] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.
[28] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.
[29] H. Zou and H. Zhang. On the adaptive elastic-net with a diverging number of parameters. *Annals of Statistics*, 37(4):1733–1751, 2009.

9 Appendix

In this section, we present the outline of the proof for model selection consistency in Theorem 5.1. Our proof directly follows the proof for adaptive single-task elastic-net [29] and extends it to the multi-task case. The asymptotic normality can be obtained from Lyapunov central limit theorem as shown in [29].

Before we go into the details of the proof for the model selection consistency, we first show the property of the estimator obtained by multi-task elastic-net with uniform weights (without $(1+\lambda_2)$-scaling):

LEMMA 9.1. *Let*

$$\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1) = \mathop{arg\,min}_{\boldsymbol{\beta}}\Big\{\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\sum_{j=1}^{p} w_j\|\boldsymbol{\beta}_j\|_2 + \lambda_2\|\boldsymbol{\beta}\|_2^2\Big\}, \tag{9.15}$$

then we have:

$$\mathbb{E}\left(\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)-\boldsymbol{\beta}^*\|_2^2\right) \le 4\frac{\lambda_2^2\|\boldsymbol{\beta}^*\|_2^2 + pK\sigma^2 + \lambda_1^2\sum_{j=1}^p w_j^2}{(bn+\lambda_2)^2}, \tag{9.16}$$

If $w_j = 1$ for all $1 \le j \le p$ (i.e., uniform weight), then we have:

$$\mathbb{E}\left(\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)-\boldsymbol{\beta}^*\|_2^2\right) \le 4\frac{\lambda_2^2\|\boldsymbol{\beta}^*\|_2^2 + pK\sigma^2 + \lambda_1^2 p}{(bn+\lambda_2)^2}, \tag{9.17}$$

Proof. The main idea of the proof follows [29] which introduces the ridge regression estimator:

$$\begin{aligned}\widehat{\boldsymbol{\beta}}(\lambda_2) &= \mathop{\arg\min}_{\boldsymbol{\beta}}\{\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_2\|\boldsymbol{\beta}\|_2^2\}\\ &= (\mathbf{X}^T\mathbf{X}+\lambda_2\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}\end{aligned}$$

We decompose $\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)-\boldsymbol{\beta}^*$ into $\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)-\widehat{\boldsymbol{\beta}}(\lambda_2)$ and $\widehat{\boldsymbol{\beta}}(\lambda_2)-\boldsymbol{\beta}^*$, and we can show that

$$\mathbb{E}\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)-\widehat{\boldsymbol{\beta}}(\lambda_2)\|_2^2 \le \frac{\lambda_1^2\sum_{j=1}^p w_j^2}{(bn+\lambda_2)^2} \tag{9.18}$$

$$\mathbb{E}(\|\widehat{\boldsymbol{\beta}}(\lambda_2)-\boldsymbol{\beta}^*\|_2^2) \le 2\frac{\lambda_2^2\|\boldsymbol{\beta}^*\|_2^2 + pK\sigma^2}{(bn+\lambda_2)^2} \tag{9.19}$$

By the fact that

$$\begin{aligned}\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)-\boldsymbol{\beta}^*\|_2^2 \le\ & 2\,\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)-\widehat{\boldsymbol{\beta}}(\lambda_2)\|_2^2\\ & +2\,\|\widehat{\boldsymbol{\beta}}(\lambda_2)-\boldsymbol{\beta}^*\|_2^2,\end{aligned}$$

we obtain the result in Eq. (9.17).

We decompose the proof of the model selection consistency into two parts: (1) for any irrelevant variable $j \in \mathcal{A}^c$, the probability that $\widehat{\boldsymbol{\beta}}^*_j = \mathbf{0}$ tends to be 1; (2) for all $j \in \mathcal{A}$, the probability that $\|\widehat{\boldsymbol{\beta}}^*_j\|_2 > 0$ tends to 1.

Now we present the first part of the model selection consistency in the following proposition:

PROPOSITION 9.1. *Let $\widetilde{\boldsymbol{\beta}}$ be the coefficients estimated by adaptive multi-task elastic-net without the $(1+\lambda_2)$-scaling:*

$$\widetilde{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \Big\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1^* \sum_{j=1}^{p} \widehat{w}_j \|\boldsymbol{\beta}_j\|_2 + \lambda_2 \|\boldsymbol{\beta}\|_2^2 \Big\}, \tag{9.20}$$

where $\widehat{w}_j = (\|\widehat{\boldsymbol{\beta}}_j\|_2)^{-\gamma}$ are the constructed adaptive weights. Then we have

$$\Pr(\forall j \in \mathcal{A}^c, \widetilde{\boldsymbol{\beta}}_j = \mathbf{0}) \to 1, \tag{9.21}$$

as $n \to \infty$.

We note that adaptive multi-task elastic-net with or without $(1+\lambda_2)$-scaling shares the same sparsity pattern. The reason why we consider adaptive multi-task elastic-net without $(1+\lambda_2)$-scaling here is mainly due to the simplicity of the notation.

Proof. For any vector $\mathbf{u}$, the subdifferential of $\|\mathbf{u}\|_2$ can be characterized as follows [22]:

$$\partial \|\cdot\|_2|_{\mathbf{u}} = \begin{cases} \{\xi : \|\xi\|_2 \le 1\} & \boldsymbol{\alpha} = \mathbf{0} \\ \frac{\mathbf{u}}{\|\mathbf{u}\|_2} & \mathbf{u} \neq \mathbf{0} \end{cases} \tag{9.22}$$

According to the Karush-Kuhn-Tucker condition of convex optimization problem in Eq. (9.20), we have that the event $\{\forall j \in \mathcal{A}^c, \widetilde{\boldsymbol{\beta}}_j = \mathbf{0}\}$ is the same as

$$-2X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}) + \lambda_1^* \widehat{w}_j \boldsymbol{\alpha}_j = 0, \quad \forall j \in \mathcal{A}^c, \tag{9.23}$$

where $\boldsymbol{\alpha}_j \in \partial \|\boldsymbol{\beta}\|_2|_{\boldsymbol{\beta}=\mathbf{0}}$. According to the property of the subdifferential of l_2-norm as in Eq. (9.22), the condition in Eq. (9.23) is equivalent to:

$$\|2X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2 \le \lambda_1^* \widehat{w}_j, \quad \forall j \in \mathcal{A}^c.$$

Therefore, Proposition 9.1 is equivalent to saying that:

$$\Pr(\forall j \in \mathcal{A}^c, \|2X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2 \le \lambda_1^* \widehat{w}_j) \to 1,$$

which is further equivalent to:

$$\Pr(\exists j \in \mathcal{A}^c, \|2X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2 > \lambda_1^* \widehat{w}_j) \to 0,$$

Let $\eta = \min_{j\in\mathcal{A}}(\|\boldsymbol{\beta}_j^*\|_2)$ and $\widehat{\eta} = \min_{j\in\mathcal{A}}(\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)\|_2)$, where $\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)$ is obtained from Eq. (9.15). By repeatedly using the union bound, we obtain:

$$\begin{aligned} &\Pr(\exists j \in \mathcal{A}^c, \|2X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2 > \lambda_1^* \widehat{w}_j) \\ \le\ & \sum_{j\in\mathcal{A}^c} \Pr(\|2X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2 > \lambda_1^* \widehat{w}_j) \\ \le\ & \sum_{j\in\mathcal{A}^c} \Pr(\|2X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2 > \lambda_1^* \widehat{w}_j, \widehat{\eta} > \eta/2) \\ & + \Pr(\widehat{\eta} \le \eta/2) \end{aligned} \tag{9.24}$$

We first analyze the last term $\Pr(\widehat{\eta} \le \eta/2)$. Let $\widehat{j} = \arg\min_{j\in\mathcal{A}}(\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)_j\|_2)$. The event $\widehat{\eta} \le \eta/2$ implies that

$$\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1) - \boldsymbol{\beta}^*\|_2 \ge \|\boldsymbol{\beta}^*_{\widehat{j}}\|_2 - \widehat{\eta} \ge \eta - \widehat{\eta} \ge \eta/2. \tag{9.25}$$

By Markov inequality and Lemma 9.1, we obtain that:

$$\begin{aligned} \Pr(\widehat{\eta} \le \eta/2) &\le \Pr(\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1) - \boldsymbol{\beta}^*\|_2 \ge \eta/2) \\ &\le \frac{\mathbb{E}(\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1) - \boldsymbol{\beta}^*\|_2^2)}{\eta^2/4} \\ &\le \frac{\lambda_2^2\|\boldsymbol{\beta}^*\|_2^2 + pK\sigma^2 + \lambda_1^2 p}{(bn+\lambda_2)^2} \cdot \frac{1}{\eta^2} \equiv K_3 \end{aligned} \tag{9.26}$$

Now we analyze the first term in Eq. (9.24). In order to bound $\widehat{w}_j$, we introduce $M = (\frac{\lambda_1^*}{n})^{\frac{1}{\gamma}}$ and consider two separate events $\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)_j\| \le M$ and $\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)_j\| > M$ for $j \in \mathcal{A}^c$ separately. More specifically, using the union bound, we obtain that:

$$\begin{aligned} &\textstyle\sum_{j\in\mathcal{A}^c} \Pr(\|2X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2 > \lambda_1^*\widehat{w}_j, \widehat{\eta} > \eta/2) \\ &\le \textstyle\sum_{j\in\mathcal{A}^c} \Pr(\|2X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2 > \lambda_1^*\widehat{w}_j, \widehat{\eta} > \eta/2, \\ &\qquad \|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)_j\| \le M) \\ &\quad + \textstyle\sum_{j\in\mathcal{A}^c} \Pr(\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)_j\|_2 > M) \end{aligned} \tag{9.27}$$

By Markov inequality, the last term in Eq. (9.27) can be easily bounded using the results from Lemma 9.1:

$$\begin{aligned} &\sum_{j\in\mathcal{A}^c} \Pr(\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)_j\|_2 > M) \\ \le\ & \frac{1}{M^2}\mathbb{E}\Big(\sum_{j\in\mathcal{A}^c} \|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)_j\|_2^2\Big) \\ \le\ & \frac{\mathbb{E}\left(\|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1) - \boldsymbol{\beta}^*\|_2^2\right)}{M^2} \\ \le\ & 4\frac{\lambda_2^2\|\boldsymbol{\beta}^*\|_2^2 + pK\sigma^2 + \lambda_1^2 p}{(bn+\lambda_2)^2} \cdot \frac{1}{M^2} \equiv K_2 \end{aligned} \tag{9.28}$$

As for the first term in (9.27), we obtain the bound also by Markov inequality with some algebraic derivations:

$$\begin{aligned} &\sum_{j\in\mathcal{A}^c} \Pr\Big(\|2X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2 > \lambda_1^*\widehat{w}_j, \\ &\qquad \widehat{\eta} > \eta/2, \|\widehat{\boldsymbol{\beta}}(\lambda_2,\lambda_1)_j\| \le M\Big) \\ &\le \sum_{j\in\mathcal{A}^c} \Pr(\|2X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2 > \lambda_1^* M^{-\gamma}, \widehat{\eta} > \eta/2) \\ &\le \frac{4M^{2\gamma}}{(\lambda_1^*)^2}\mathbf{E}\left(\sum_{j\in\mathcal{A}^c} \|X_j^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2^2 I(\widehat{\eta} > \eta/2\right) \\ &\le \frac{4M^{2\gamma}}{(\lambda_1^*)^2}\Big(8p^2K\frac{\lambda_2^2\|\boldsymbol{\beta}^*\|_2^2 + pK\sigma^2 + (\lambda_1^*)^2(\eta/2)^{-2\gamma}p}{(bn+\lambda_2)^2} \\ &\quad + 2npK\sigma^2\Big) \equiv K_1 \end{aligned} \tag{9.29}$$

Now combining Eq. (9.24) with Eq. (9.26), (9.28) and (9.29), we have

$$\Pr(\exists j \in \mathcal{A}^c, \|2X_j^T(\mathbf{y}-\mathbf{X}_{\mathcal{A}}\widetilde{\boldsymbol{\beta}}_{\mathcal{A}})\|_2 > \lambda_1^* \widehat{w}_j) \leq K_1+K_2+K_3.$$

Under the assumption (A2) and with the parameters satisfying (B1) and (B2) and $\gamma > \frac{2}{1-\nu}$, we have that K_1, K_2 and K_3 all go to zero as n goes to infinity. Therefore, we obtain that $\Pr(\forall j \in \mathcal{A}^c, \widetilde{\boldsymbol{\beta}}_j = \mathbf{0}) \to 1$.

Now we show the other half of the model selection consistency: for any $j \in \mathcal{A}$, $\widetilde{\boldsymbol{\beta}}_j \neq \mathbf{0}$. We characterize it in the next proposition.

PROPOSITION 9.2. *Let $\widetilde{\boldsymbol{\beta}}$ be obtained from Eq.* (9.20)*, then we have:*

$$\Pr(\min_{j\in\mathcal{A}} \|\widetilde{\boldsymbol{\beta}}_j\|_2 > 0) \to 1$$

We first introduce

$$\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_2) = \arg\min_{\boldsymbol{\beta}} \Big\{ \|\mathbf{y} - \mathbf{X}_{\mathcal{A}}\boldsymbol{\beta}\|_2^2 + \lambda_2 \sum_{j\in A} \|\boldsymbol{\beta}_j\|_2^2 \Big\}. \tag{9.30}$$

By the same argument as in Eq. (9.25), we have

$$\min_{j\in\mathcal{A}} \|\widetilde{\boldsymbol{\beta}}_j\|_2 > \min_{j\in\mathcal{A}} \|\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_2)_j\|_2 - \|\widetilde{\boldsymbol{\beta}}_{\mathcal{A}} - \widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_2)\|_2$$

and

$$\min_{j\in\mathcal{A}} \|\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_2)_j\|_2 > \min_{j\in\mathcal{A}} \|\boldsymbol{\beta}_j^*\|_2 - \|\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_2) - \boldsymbol{\beta}_{\mathcal{A}}^*\|_2$$

According to Eq. (9.18) and (9.19), we have

$$\begin{aligned} \|\widetilde{\boldsymbol{\beta}}_{\mathcal{A}} - \widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_2)\|_2 &\leq \frac{\lambda_1^* \sqrt{\sum_{j\in\mathcal{A}} \widehat{w}_j^2}}{bn+\lambda_2} \leq \frac{\lambda_1^* \sqrt{p} \max \widehat{w}_j}{bn+\lambda_2} \\ &\leq \frac{\lambda_1^* \sqrt{p} \widehat{\eta}^{-\gamma}}{bn+\lambda_2}, \end{aligned}$$

and

$$\mathbb{E}(\|\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_2) - \boldsymbol{\beta}^*\|_2^2) \leq 2\frac{\lambda_2^2 \|\boldsymbol{\beta}^*\|_2^2 + pK\sigma^2}{(bn+\lambda_2)^2}$$

Then we can easily show that both $\|\widetilde{\boldsymbol{\beta}}_{\mathcal{A}} - \widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_2)\|_2$ and $\mathbb{E}(\|\widetilde{\boldsymbol{\beta}}_{\mathcal{A}}(\lambda_2) - \boldsymbol{\beta}^*\|_2^2)$ converge to zero in probability, i.e., $o_p(1)$. Since $\eta = \min_{j\in\mathcal{A}} \|\boldsymbol{\beta}_j^*\|_2 > 0$, we have $\Pr(\min_{j\in\mathcal{A}} \|\widetilde{\boldsymbol{\beta}}_j\|_2 > 0) \to 1$.