Multitask Learning using Task Clustering with Applications to Predictive Modeling and GWAS of Plant Varieties

Ming Yu
Booth School of Business, The University of Chicago
Chicago, IL, USA
ming93@uchicago.edu

Karthikeyan Natesan Ramamurthy,
Aurélie C. Lozano, Naoki Abe
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
{knatesa,aclozano,nabe}@us.ibm.com

Eunho Yang
KAIST
Daejeon, South Korea
eunhoy@kaist.ac.kr

Addie M. Thompson, Melba M. Crawford,
Ayman F. Habib, Mitchell R. Tuinstra
Purdue University
W. Lafayette, IN, USA
{thomp464,melbac,ahabib,drmitch}@purdue.edu

ABSTRACT

Inferring predictive maps between multiple input and multiple output variables or tasks has innumerable applications in data science. Multi-task learning attempts to learn the maps to several output tasks simultaneously with information sharing between them. We propose a novel multi-task learning framework for sparse linear regression, where a full task hierachy is automatically inferred from the data, with the assumption that the task parameters follow a hierarchical tree structure. The leaves of the tree are the parameters for individual tasks, and the root is the global model that approximates all the tasks. We apply the proposed approach to develop and evaluate: (a) predictive models of plant traits using large-scale and automated remote sensing data, and (b) GWAS methodologies mapping such derived phenotypes in lieu of hand-measured traits. We demonstrate the superior performance of our approach compared to other methods, as well as the usefulness of discovering hierarchical groupings between tasks. Our results suggest that richer genetic mapping can indeed be obtained from the remote sensing data. In addition, our discovered groupings reveal interesting insights from a plant science perspective.

KEYWORDS

High-throughput phenotyping, Multitask learning, Convex clustering, Sparse linear regression

1 INTRODUCTION

Several problems in data science necessitate discovery of a predictive mapping from input to output data, such that it can be generalized to new inputs. In many applications, there can be more than one output variable and they can be related to each other, and it is also desirable for the mappings to have some structure. Multi-output regression models generalize single-output regression models to learn predictive relationships between multiple input and multiple output variables, also referred to as tasks [1]. Multi-task learning attempts to learn several the inference tasks simultaneously, and the assumption here is that an appropriate sharing of information can benefit all the tasks [2, 18]. There are several ways to define task-relatedness; the parameters can be close to each other [6], they can share a common prior [5], or they can share a common latent feature representation [2]. We will only focus on multi-task learning under the linear sparse modeling framework.

The implicit assumption that sharing all features for all tasks can be excessive and can ignore the underlying specificity of the mappings. There have been several extensions to multi-task learning that address the problem of how to, and with whom to share between the tasks [9–12, 14, 22]. The authors in [10] propose a *dirty* model for feature sharing among tasks, wherein a linear superposition of two sets of parameters - one that is common to all tasks, and one that is task-specific is used. A variant of this was proposed in [22] with elementwise product between the two parameter sets. The approach proposed in [14] learns to share by defining a set of *basis task parameters* and posing the task-specific parameters as a sparse linear combination of these. In [9] and [11], the authors assume that the tasks are clustered into groups and proceed to learn the group structure along with the task parameters using a convex and an integer quadratic program respectively. In contrast to the above approaches, [12] leverages a predefined tree structure among the output tasks (e.g. using hierarchical agglomerative clustering) and imposes group regularizations on the task parameters based on this tree.

In this work, we propose to simultaneously learn the task parameters of a multi-task linear regression model as well as the structured relationship between the tasks in a fully supervised manner. We assume that the task parameters follow a general hierarchical tree

structure, and hence this approach learns the full task sharing hierarchy automatically. We adopt such a structure as it is natural to model task relatedness, similarly as in hierarchical clustering and it is also easily interpretable. Our framework is motivated by several applications such as in genome wide association analysis with multiple output phenotypes [21], and predictive modeling of plant traits with remote sensed data in automated phenotyping [19] to name a few. To the best of our knowledge, ours is the first attempt in the literature to learn a hierarchical task clustering structure in a supervised manner. Our proposed approach constrains the parameters of each task to be sparse and incorporates an additional regularization on the task parameters inspired by convex clustering [7]. Similar to convex clustering, a continuous regularization path of solutions can be obtained and each point in the path corresponds to a threshold for cutting the task parameter tree.

The main contributions of this paper are as follows. We propose a structured multi-task learning estimator that can learn a tree structure among the tasks while simultaneously estimating the task parameters. We present a proximal decomposition approach to efficiently solve the resulting convex problem, and show its numerical convergence. We provide statistical guarantees and study the asymptotic properties of our estimator. Our approach is validated empirically on simulated data. We also explore the applications of the proposed method in two high-throughput phenotyping problems: (a) predictive modeling of plant traits using large-scale and automated remote sensing data, and (b) Genome-Wide Association Studies (GWAS) mapping such derived phenotypes in lieu of hand-measured traits. Initial results are promising and show that our method produces high predictive accuracy while revealing groupings that are insightful.

2 BACKGROUND

Multi-task Learning. Consider the multi-task linear regressions: for each task $s = 1, \ldots, k$

$$y = x^T \theta_s^* + \epsilon,$$

for where $y \in \mathbb{R}$, $x \in \mathbb{R}^p$, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Suppose we are given n_s i.i.d. samples $\{x_i, y_i\}_{i=1}^{n_s}$ for task s. We will overload notation and collate these observations using vector notation as:

$$y_s = X_s \theta_s^* + \epsilon_s,$$

where $y_s \in \mathbb{R}^{n_s}$, $X_s \in \mathbb{R}^{n_s \times p}$, and $\epsilon_s \in \mathbb{R}^{n_s}$. Throughout the paper, we concatenate k parameters in columns to form $\Theta^* \in \mathbb{R}^{p \times k}$ (i.e. s-th column, Θ_s^*, is θ_s^*), and assume that all tasks have the same number of observations n for notational simplicity. Note that in the multi-response model where the design matrix X_s is shared across all tasks, we have $Y = X\Theta^* + E$ in a simpler form with $Y \in \mathbb{R}^{n \times k}$, $X \in \mathbb{R}^{n \times p}$ and $E \in \mathbb{R}^{n \times k}$. Generally, multi-task learning solves the following optimization problem

$$\underset{\Theta}{\text{minimize}} \sum_{s=1}^{k} \|y_s - X_s \Theta_s\|_2^2 + \lambda_1 \|\Theta_s\|_1 + \Omega(\Theta) \qquad (1)$$

where the penalty $\Omega(\Theta)$ on the k task parameters encourages information sharing.

Convex Clustering. Clustering is a fundamental *unsupervised* problem which is broadly used in many scientific applications. Given k data points $x_1, \ldots, x_k \in \mathbb{R}^p$ (our choice of notation here is deliberate for the next section), Hocking et al. [8], Lindsten et al. [15] propose a convex optimization problem for clustering k points:

$$\underset{v}{\text{minimize}} \frac{1}{2} \sum_{s=1}^{k} \|x_s - v_s\|_2^2 + \lambda \sum_{s<t} w_{st} \|v_s - v_t\|_2 \qquad (2)$$

where λ is a positive tuning parameter, w_{st} is a nonnegative weight. Note that for the second term, other penalties rather than ℓ_2 norm have been also considered. As discussed in Chi and Lange [3], each point constitutes an independent cluster when λ is small enough, however, as λ increases, the cluster centers start to combine.

3 MULTITASK LEARNING WITH TASK CLUSTERING

Our goal is to jointly infer (a) the linear regression parameter matrix Θ^* allowing for high-dimensional sampling settings where the number of observations is possibly much larger than the problem dimension p; (b) a general hierarchical tree structure among the tasks.

To accomplish this goal, we propose to solve the following regularized regression problem:

$$\underset{\Theta}{\text{minimize}} \sum_{s=1}^{k} \left\{ \|y_s - X_s \Theta_s\|_2^2 + \lambda_1 \|\Theta_s\|_1 \right\}$$
$$+ \lambda_2 \sum_{s<t} w_{st} \|\Theta_s - \Theta_t\|_2 . \qquad (3)$$

The above formulation has three terms. The first term is the squared loss. While we focus on linear regression, our approach can be readily generalized and employ other loss functions such as the logistic loss, the ℓ_1 loss, etc.

The second term encourages the sparsity of the regression coefficient matrix Θ. Clearly, this can be generalized to a more sophisticated sparsity structure, e.g., group sparsity, based on the needs of the application.

The third term encourages tasks to share the same value of their parameter vectors. It can be viewed as a generalization of the fused lasso penalty [23] wherein we *fuse* the parameter vectors rather than the scalar coefficients. This is identical to the *convex clustering* penalty [3] in (2), but it forms clusters for task parameters rather than data points. Hence, brought into our framework, this penalty induces clustering of task parameters, effectively learning a hierarchical tree of these parameters.

λ_1 and λ_2 are tuning parameters that control the degree of individual task sparsity and task sharing respectively. As in (2), when λ_2 is small enough, each task parameter Θ_s is allowed to focus on minimizing individual loss function $\|y_s - X_s \Theta_s\|_2^2 + \lambda_1 \|\Theta_s\|_1$. However, as λ_2 increases, the relative importance of third term increases and task parameters start to combine. In this sense, our formulation (3) can be viewed as *supervised* clustering problem. Setting these parameters is essential to decide the behavior of (3), and they can be tuned e.g. via cross validation. Finally, w_{st} are optional non-negative weights that can be imposed to reflect prior knowledge on the degree of relatedness between each pair of tasks.

The choice of the weights can dramatically affect the complexity of the minimization problem. Typically it is desirable to have sparse weights w_{st}: only a small portion of them are non-zero. As suggested in [3], we can set $w_{st} = 1_{st}^\kappa \cdot \exp(-\phi\|\boldsymbol{y}_s - \boldsymbol{y}_t\|_2^2)$, where 1_{st}^κ is 1 if t is among s's κ-nearest-neighbors or vice versa and 0 otherwise. The constant ϕ is nonnegative; $\phi = 0$ corresponds to uniform weights for all κ neighbors.

3.1 Optimization via Proximal Decomposition

In this section we describe an optimization algorithm to solve our formulation (3). Since the task loss function for linear regression problem is convex, our formulation overall is also convex in Θ, so it can be solved using modern convex optimization approaches. The story here can be trivially generalized to general loss function $\ell(\Theta)$ beyond squared loss, as long as it is convex.

Here we adopt the proximal decomposition method introduced in [4]. This is an efficient algorithm for minimizing the summation of several convex functions. Our objective function involves $m = 2 + \#\{(s, t) : w_{st} > 0\}$ such functions (Note that w_{st} is usually sparse so we do not have to deal with $O(k^2)$ terms):

$$f_1(\Theta) = \sum_{s=1}^{k} \|y_s - X_s\Theta_s\|_2^2,$$

$$f_2(\Theta) = \lambda_1 \sum_{s=1}^{k} \|\Theta_s\|_1,$$

$$f_{s,t}(\Theta) = \lambda_2 \sum_{s<t} w_{st}\|\Theta_s - \Theta_t\|_2 : \{(s,t) : w_{s,t} > 0\}.$$

At a high level, the algorithm iteratively applies proximal updates with respect to the above functions. We now describe the details of the algorithm as summarized in Algorithm 1. We stack the regression matrix Θ into a column vector $(\Theta_1; ...; \Theta_k) \in \mathbb{R}^{pk}$. In the algorithm l denote the iteration number. Each $\widetilde{\Theta}_{1,l}, \widetilde{\Theta}_{2,l}$ or $\widetilde{\Theta}_{st,l}$ is a pk dimensional column vector, corresponding to our parameter $(\Theta_1; ...; \Theta_k)$. The procedure has two additional parameters γ and μ_l. In practice we can set $\mu_l = 1$ for each iteration l. 'prox' denotes the proximal operator: $\text{prox}_f b = \underset{a}{\text{argmin}} \left(f(a) + \frac{1}{2}\|b-a\|^2 \right)$, where a and b are vectors.

We iteratively apply the proximal updates for f_1, f_2 and $f_{s,t}$ until convergence. The specific update rules are as follows.

- Update for f_1: Let $(a_1; ...; a_k) = \text{prox}_{\sigma f_1}(b_1; ...; b_k)$, For each $s = 1, ... k$, we have

$$a_s = \left(\sigma X_s^T X_s + \frac{1}{2}I_p\right)^{-1} \cdot \left(\sigma X_s^T y_s + \frac{1}{2}b_s\right).$$

This step corresponds to the closed-form formula of a ridge regression problem. For very large p we can employ efficient approaches such as [17] and [16].
- Update for f_2: Let $(a_1; ...; a_k) = \text{prox}_{\sigma f_2}(b_1; ...; b_k)$, For each $s \in \{1, ..., k\}, j \in \{1, ..., p\}$,

$$[a_s]_j = \left[1 - \frac{\lambda_1\sigma}{|[b_s]_j|}\right]_+ \cdot [b_s]_j.$$

Algorithm 1

Initialize:
$\gamma \in (0, \infty), \widetilde{\Theta}_{1,0} \in \mathbb{R}^{pk}, \widetilde{\Theta}_{2,0} \in \mathbb{R}^{pk}$
$\widetilde{\Theta}_{st,0} \in \mathbb{R}^{pk}$, for $(s, t) : w_{st} > 0$
$\widehat{\Theta}_0 = \frac{1}{m}(\widetilde{\Theta}_{1,0} + \widetilde{\Theta}_{2,0}) + \frac{1}{m}\sum_{s,t:w_st>0}\widetilde{\Theta}_{st,0}$

for $l = 0, 1, 2, 3, ...$ **do**
  $p_{1,l} = \text{prox}_{\sigma f_1}\widetilde{\Theta}_{1,l}$ where $\sigma = m\gamma$
  $p_{2,l} = \text{prox}_{\sigma f_2}\widetilde{\Theta}_{2,l}$ where $\sigma = m\gamma$
  for s, t s.t. $w_{st} > 0$ **do**
    $p_{st,l} = \text{prox}_{\sigma f_{st}}\widetilde{\Theta}_{st,l}$ where $\sigma = m\gamma$
  end for
  $p_l = \frac{1}{m}(p_{1,l} + p_{2,l}) + \frac{1}{m}\sum_{s,t:w_{st}>0}p_{st,l}$
  $\mu_l \in (0, 2)$
  $\widetilde{\Theta}_{1,l+1} = \widetilde{\Theta}_{1,l} + \mu_l(2p_l - \widehat{\Theta}_l - p_{1,l})$
  $\widetilde{\Theta}_{2,l+1} = \widetilde{\Theta}_{2,l} + \mu_l(2p_l - \widehat{\Theta}_l - p_{2,l})$
  for s, t s.t. $w_{st} > 0$ **do**
    $\widetilde{\Theta}_{st,l+1} = \widetilde{\Theta}_{st,l} + \mu_l(2p_l - \widehat{\Theta}_l - p_{st,l})$
  end for
  $\widehat{\Theta}_{l+1} = \widehat{\Theta}_l + \mu_l(p_l - \widehat{\Theta}_l)$
end for

Reshape $\widehat{\Theta}$ to get Θ.

- Updates for $f_{s,t}$: Let $(a_1; ...; a_k) = \text{prox}_{\sigma f_{st}}(b_1; ...; b_k)$, For each $(s, t) \in \{1, ..., k\}^2$ with $w_{st} > 0$, let $c_{st} = \frac{\sigma\lambda_2}{\|b_s - b_t\|}$,

$$a_s = (1 - c_{st}) \cdot b_s + c_{st} \cdot b_t,$$
$$a_t = (1 - c_{st}) \cdot b_t + c_{st} \cdot b_s.$$

3.2 Numerical Convergence

The following proposition characterizes the convergence of Algorithm 1. It is a direct consequence of Theorem 3.4. in [4].

PROPOSITION 3.1. *Let G be the set of solutions to problem* (3) *and let $(\widehat{\Theta}_l)_{l\in\mathbb{N}}$ be a sequence generated by Algorithm 1. Then provided that $\sum_{l\in\mathbb{N}}\mu_l(2 - \mu_l) = +\infty$ Then $G \neq \emptyset$ and $(\widehat{\Theta}_l)_{l\in\mathbb{N}}$ converges weakly to a point in G.*

The proof is presented in the appendix.

4 THEORETICAL GUARANTEES

In this section, we provide the asymptotic properties of our estimators (3) that can be understood as the extension of those for vanilla Lasso [13] and fused Lasso [23]. Following the strategy in [13, 23], we assume in the main statement that the feature dimension p is fixed as the number of samples n approaches infinity. Though restrictive, this setting is more effective to illustrate the dynamics of methods.

Before providing the main statement, we first introduce some quantities for clear representation. Let Θ^* be the population minimizer of the risk: $\Theta^* \in \text{argmin}_\Theta \sum_{s=1}^{k} \{\mathbf{E}\|Y - X_s\Theta\|_F^2 + \lambda_1\|\Theta_s\|_1\} + \lambda_2 \sum_{s<t} w_{st}\|\Theta_s - \Theta_t\|_2$. Given the population or target parameter Θ^*, we define two functions corresponding to two types of regularizers. Specifically, let $F(U; \Theta)$ be the limit definition of derivative for individual ℓ_1 regularization terms: $\lim_{h\to+0} \frac{\sum_{s=1}^{k}\{\|\Theta_s+hU_s\|_1-\|\Theta_s\|_1\}}{h}$,

where $U \in \mathbb{R}^{p \times k}$. Then, $F(U; \Theta^*)$ can be rewritten as

$$F(U; \Theta^*) := \sum_{s=1}^{k} \sum_{j=1}^{p} U_{js} \text{sgn}(\Theta_{js}^*) \mathbf{I}(\Theta_{js}^* \neq 0)$$
$$+ |U_{js}| \mathbf{I}(\Theta_{js}^* = 0) \qquad (4)$$

where $\mathbf{I}(\cdot)$ is the indicator function, and Θ_{js}^* (and U_{js} respectively) is the j-th feature weight of the target parameter Θ_s^*. Similarly, let $G_{st}(U_s, U_t; \Theta_s, \Theta_t)$ be the limit definition of derivative for a single fusion term $w_{st} \|\Theta_s - \Theta_t\|_2$, defined as

$$\lim_{h \to +0} \frac{w_{st} \|\Theta_s + hU_s - \Theta_t - hU_t\|_2 - w_{st} \|\Theta_s - \Theta_t\|_2}{h}.$$

Then, it can be easily verified that the sum of G_{st} over all fusion terms is written as

$$G(U; \Theta^*) := \sum_{s<t} w_{st} \frac{(\Theta_s^* - \Theta_t^*)^\top (U_s - U_t)}{\|\Theta_s^* - \Theta_t^*\|_2} \mathbf{I}(\Theta_s^* \neq \Theta_t^*)$$
$$+ w_{st} \|U_s - U_t\|_2 \mathbf{I}(\Theta_s^* = \Theta_t^*). \qquad (5)$$

Armed with these quantities, the following theorem characterizes the limiting distribution on optimal solution of (3), $\widehat{\Theta}$.

THEOREM 4.1. *Consider a sequence of λ_1 and λ_2 such that $\lambda_1/\sqrt{n} \to \lambda_1^0 \geq 0$ and $\lambda_2/\sqrt{n} \to \lambda_2^0 \geq 0$ respectively, as $n \to \infty$. Suppose that $C_s := \lim_{n \to \infty} \left(\frac{1}{n} X_s^\top X_s \right)$ for every task $s = 1, \ldots, k$, is non-singular. Then, the estimator $\widehat{\Theta}$ in (3) satisfies*

$$\sqrt{n} \left(\widehat{\Theta} - \Theta^* \right) \xrightarrow{d} \underset{U}{\arg\min} V(W, U) \qquad (6)$$

where $V(W, U) := \sum_{s=1}^{k} \{-2U_s^\top W_s + U_s^\top C_s U_s\} + \lambda_1^0 F(U; \Theta^*) + \lambda_2^0 G(U; \Theta^*)$ using definitions in (4) and (5), and $W \in \mathbb{R}^{p \times k}$ has an $p \times k$ normal distribution whose s-th column follows $N(0, \sigma^2 C_s)$.

Given the construction of (4) and (5), the proof of Theorem 4.1 follows easily from the line of previous works e.g. [13] or [23].

The theorem confirms that the joint limiting distribution of $\sqrt{n}(\widehat{\Theta} - \Theta^*)$ will have probability concentrated on the line $U_s = U_t$ if $\Theta_s^* = \Theta_t^*$, $w_{st} > 0$ and $\lambda_2^0 > 0$, in addition to a lasso-type effect (as in [13]) on univariate limiting distributions when $\lambda_1^0 > 0$.

Note that in case of a 'multi-output' setting where the design matrix X is shared across all tasks, Theorem 4.1 holds with the simpler form of $V(W, U)$: $-2\text{Tr}(U^\top W) + \text{Tr}(U^\top CU) + \lambda_1^0 F(U; \Theta^*) + \lambda_2^0 G(U; \Theta^*)$.

Note also that the extension of Theorem 4.1 beyond least squares

$$\underset{\Theta}{\text{minimize}} \sum_{s=1}^{k} \left\{ \mathcal{L}(\Theta_s; y_s, X_s) + \lambda_1 \|\Theta_s\|_1 \right\}$$
$$+ \lambda_2 \sum_{s<t} w_{st} \|\Theta_s - \Theta_t\|_2,$$

where $\mathcal{L}$ is a general loss function (e.g. logistic loss) is also trivially attainable following [20].

5 SIMULATED DATA EXPERIMENTS

As a sanity check, we evaluate our approach on synthetic datasets and compare with the following baselines:

- **Single task**: a baseline approach, where the tasks are learned separately via Lasso.

Table 1: RMSE for different comparison methods

Method	RMSE	std	time
Single task	5.665	0.131	0.02
No group multitask learning	5.520	0.115	0.05
Pre-group multitask learning	5.256	0.117	0.10
Kang et al	5.443	0.096	> 10
Tree guided group Lasso	5.448	0.127	0.03
Ours	**4.828**	0.117	0.16

- **No group MTL** [18]: the traditional multitask approach using group lasso penalty, where all tasks are learned jointly and the same features are selected across tasks.
- **Pre-group MTL**: Given the *true* number of groups, first partition the tasks purely according to the correlation among responses and then apply *No group MTL* in each cluster.
- **Kang et al** [11]: Mixed integer program learning a shared feature representations among tasks, while simultaneously determining "with whom" each task should share. We used the code provided by the authors of [11] and used the true number of tasks.
- **Tree-guided group Lasso** [12]: Employs a structured penalty function induced from a predefined tree structure among responses, that encourages multiple correlated responses to share a similar set of covariates. We used the code provided by the authors of [12] where the tree structure is obtained by running a hierarchical agglomerative clustering on the responses.

We consider $n = 20$ samples, $p = 50$ features, and three groups of tasks. Within each group there are 5 tasks whose parameter vectors are sparse and identical to each other. We generate independent train, validation, and test sets. For each method, we select the regularization parameters using the validation sets, and report the root-mean-squared-error (RMSE) of the resulting models on the test sets. We repeat this procedure 5 times. The results are reported in Table 1. From the table we can see that the *Single task* method has the largest RMSE as it does not leverage task relatedness; *No group MTL* has slightly better RMSE; both *Kang et al.* and *Tree guided group Lasso* get some improvement by considering structures among tasks; *Pre-group MLT* achieves a better result, mainly because it is given the true number of groups and for these synthetic datasets it is quite easy to obtain good groups via response similarity, which might not necessarily be the case with real data and when the predictors differ from task to task. Our method achieves the smallest RMSE, outperforming all approaches compared. We also report the running times of each method, where we fix the tolerance parameters to 10^{-5} for comparison. Though the timing for each method could always be improved using more effective implementations, the timing result reflect the algorithmic simplicity of the steps in our approach compare to e.g. the mixed integer program of [11].

An example of tree structure learnt by our approach is shown in Figure 1. It is contrasted with Figure 2, which depicts the structure obtained *a-posteriori* by performing hierarchical clustering on the regression matrix learnt by *Single Task*. The simulation scenario is the same as before except that the non-zero coefficients in each true
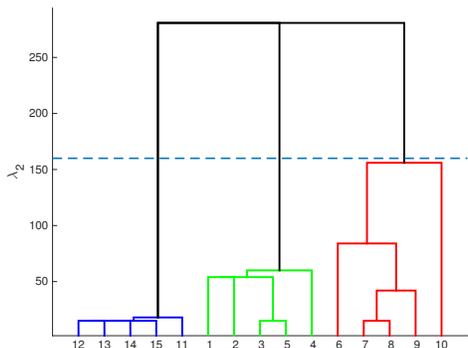
Figure 1: Tree structure learnt by our method

Figure 2: Tree structure learnt by post-clustering of Single Task Lasso

group are not equal but sampled as $0.5 + N(0, 1)/3$, where N denote the normal distribution. As can be seen from Figure 1, no matter what λ_2 is, our approach never makes a mistake in the sense that it never puts tasks from different true groups in the same cluster. For $\lambda_2 > 150$ our approach recognizes the true groups. As λ_2 becomes very large there are no intermediary situation where two tasks are merged first. Instead all tasks are put in the same cluster. We see tasks $\{3, 5\}$ merge first in group $\{1 - 5\}$ and $\{7, 8, 9\}$ merge first in group $\{6 - 10\}$. This corresponds to the fact that tasks $\{3, 5\}$ have largest correlation among group $\{1 - 5\}$ and $\{7, 8, 9\}$ has largest correlation among group $\{6 - 10\}$. We can see in Figure 2 that for *Single Task* post clustering, task 10 does not merge with $\{6 - 9\}$.

Impact of the weights w_{st}.. One might argue that our approach relies on "good" weights w_{st} among tasks. However, it turns out that it is quite robust to the weights. Recall that we select the weight w_{st} by κ-nearest-neighbors. In this synthetic dataset, we have 5 tasks in each group so the most natural way is to set $\kappa = 4$. We also try setting $\kappa = 2, 3, 5, 6$ and see how this affects the result. The test RMSEs for different κ's are given in Table 2. From the table we see that although the best performance is when we select $\kappa = 4$, our method is quite robust to the choice of weights, especially when κ is smaller than the natural one. When κ is large the result gets slightly worse, because now we cannot avoid positive weights across groups. But even if it is slightly worse, our method is still competitive.

Table 2: RMSE for our approach with weight specification by κ- nearest-neighbors.

κ	2	3	4	5	6
RMSE	4.847	4.836	**4.828**	4.896	4.928

6 APPLICATIONS TO PREDICTIVE MODELING AND GWAS OF PLANT VARIETIES

We choose high throughput phenotyping and GWAS with real data obtained from plants to demonstrate the real-world utility of our approach. Accurate phenotyping of different crop varieties is a crucial yet traditionally a time-consuming step in crop breeding requiring manual survey of hundreds of plant varieties for the traits of interest. Recently, there has been a surge in interest and effort to develop automated, high-throughput remote sensing systems [24] for trait development and genome wide association studies. The typical workflow of such systems is illustrated in Figure 3. Machine learning approaches are used to learn mappings between features obtained from remotely sensed images (e.g., RGB, hyperspectral, LiDAR, thermal) of plants as input and the manually collected traits (e.g., plant height, stalk diameter) as outputs. Genetic mapping is performed either with the inferred traits or directly with the features from remote sensing data.

We consider two specific problems from this pipeline: (a) predictive modeling of plant traits using features from remote sensed data (Section 6.1), (b) GWAS using the remote sensing data features (Section 6.2). Both these problems have unique challenges as discussed later. We apply our proposed multitask learning approach to automatically discover hierarchical groupings of output tasks, and learn differentiated models for the different groups while sharing information within groups. Apart from providing an improved predictive accuracy, the discovered groupings reveal interesting and interpretable relations between the tasks. The data used in the two experiments were collected in experimental fields in Summer-Fall 2015 and Summer-Fall 2016 respectively.

6.1 Trait Prediction from Remote Sensed Data

The experimental data consists of 18 varieties of the sorghum crop planted in 6 replicate plot locations. From the RGB and hyperspectral images of each plot, we extract features of length 206. Hence $n = 6$, $p = 206$, and $k = 18$. The presence of multiple varieties with replicates much smaller in number than predictors poses a
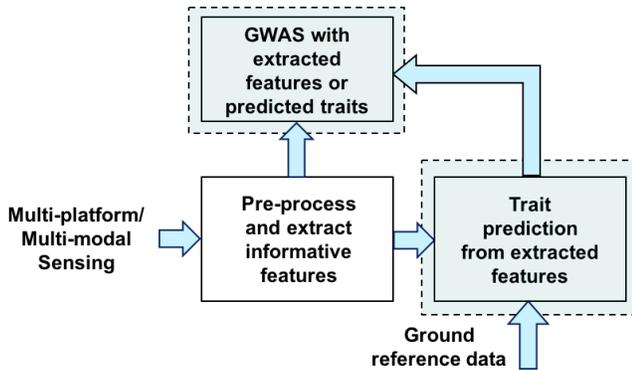
Figure 3: Context of our real data experiments. We apply our multitask learning approach for mapping from input remote sensing features to output plant traits (Section 6.1), and GWAS with remote sensing data features (Section 6.2).

Table 3: RMSE for comparison methods on Plant Height Prediction.

Method	RMSE	std
Single model	44.39	6.55
No group multitask learning	36.94	6.10
Kang et al	37.55	7.60
Ours	**33.31**	5.10

major challenge: Building separate models for each variety is unrealistic, while a single model does not fit all. This is where our hierarchical grouping and modeling approach provides the flexibility to share information that leads to learning at the requisite level of robustness.

We have 3 different responses: plant height, stalk diameter, and stalk volume so we get 3 tree structures. The 3 trees are given in Figure 4, Figure 5 and Figure 6. The cluster provide some interesting insights from a plant science perspective. As highlighted in Figure 4, for predictive models of height, thicker medium dark plants are grouped together. Similarly for thinner tall dark plants, and thick tall plants with many light leaves. Figure 5 and Figure 6 are quite similar, which makes sense, since stalk diameter and stalk volume are highly correlated. In terms of stalk, we can see variety 12 is very different from others. It corresponds to tall thin plants with few small dark leaves.

In terms of predictive accuracy comparison, we focus on predictive plant height. We perform 6-folds CV where in each fold we make sure to include one sample from each variety. As we only have $n = 6$ sample per variety (i.e. per task), it is unrealistic to learn separate models for each variety. Due to the limited sample size we focus on assessing the quality of the groupings of various methods as follows. For each CV split, we first learn a grouping using a comparison method and then treat all the samples within a group as i.i.d and estimate their regression coefficients using Lasso. We compare the groupings of (i) our approach (ii) Kang et al (iii)

learning a single predictive model using Lasso, treating all the varieties as i.i.d. (iv) learning a traditional multitask model using Group Lasso (each variety form a separate group). The results are reported in Table 3, which indicate the superior quality of our groupings in terms of improving predictive accucacy.

6.2 GWAS dataset

In the second dataset, we use SNP data from 850 varieties as input for GWAS. We considered $p = 3025$ SNPs (features). There are $n = 1920$ plots (observations), each containing a single variety. The output data is the histogram of photogrammetrically derived heights obtained from RGB images of the $n = 1920$ plots. We consider 10 bins, and each bin is treated as a task. Therefore, $k = 10$. It has been demonstrated that height histograms describe the structure of the plants in the plot and are hence powerful predictors of various traits [19]. Therefore it is worthwhile performing genomic mapping using them bypassing trait prediction. Since it is reasonable to expect the neighboring bins of the histograms to be correlated, our approach for hierarchical task grouping will result in an improved association discovery.

For this dataset, the Kang et al. method did not scale to handle the amount of features. In general, we noticed that this algorithm is quite unstable, namely the task membership keep changing at each iteration especially as the problem dimensionality becomes large. The tree structure obtained by our method is given in Figure 7. Please note that the y-axis in the figure is $\log(\lambda_2)$. We notice that bins $\{8, 9, 10\}$ merge quickly while bins $\{2,3,4\}$ merge when λ_2 is extremely large. Note that the distance from bin 5 to bin 4 is much larger, compared to the distance from bin 7 to bin 5 (while in the figure it looks similar due to logarithmic scale. Bins $\{1, 7, 8, 9, 10\}$ are rarely populated. They all have small coefficients and merge together quickly; while more populated bins tend to merge later.

We zoom in on two locations on the genome: one on chromosome 7 between positions 5.5e7 and 6.2e7 (bp) and the other on chromosome 9 between positions 5.3e7 and 6e7 (bp). Notice that selected SNPs differ from one histogram bin group to another. Selected SNPs on chromosome 7 co-localize with ground truth (in particular with Dwarf3 gene). On chromosome 9, bin mapping identifies new regions (in addition to Dwarf1), possibly related to canopy closure, leaf distribution, flowering, and other plot-wide characteristics. As future work we plan to assess the statistical significance of the uncovered association and validate them further with domain experts.

7 CONCLUSION

In this paper we proposed a multitask learning approach that jointly learns the task parameters and a tree structure among tasks, in a fully automated, data-driven manner. The induced tree structure is natural for modeling task-relatedness and it is also easily interpretable. We developed an efficient procedure to solve the resulting convex problem and proved its numerical convergence. We also explored the statistical properties of our estimator. Using synthetic data, we demonstrated the superiority of our approach over comparable methods in terms of RMSE as well as inference of underlying task groups. Experiments with real-world high throughput phenotyping data illustrated the practical utility of our approach in an
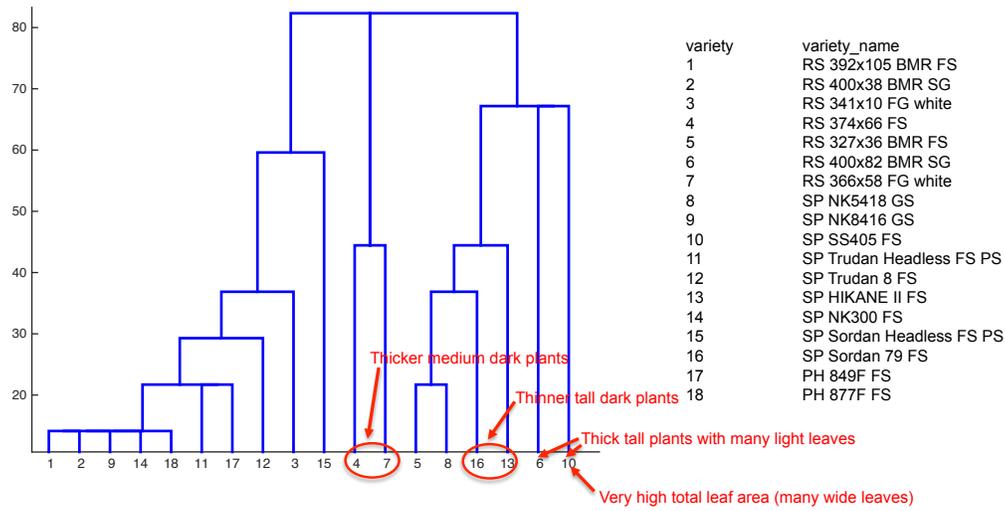
variety	variety_name
1	RS 392x105 BMR FS
2	RS 400x38 BMR SG
3	RS 341x10 FG white
4	RS 374x66 FS
5	RS 327x36 BMR FS
6	RS 400x82 BMR SG
7	RS 366x58 FG white
8	SP NK5418 GS
9	SP NK8416 GS
10	SP SS405 FS
11	SP Trudan Headless FS PS
12	SP Trudan 8 FS
13	SP HIKANE II FS
14	SP NK300 FS
15	SP Sordan Headless FS PS
16	SP Sordan 79 FS
17	PH 849F FS
18	PH 877F FS

Figure 4: Tree structure of tasks (varieties) inferred using our approach for plant height.
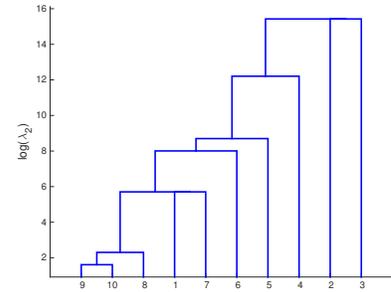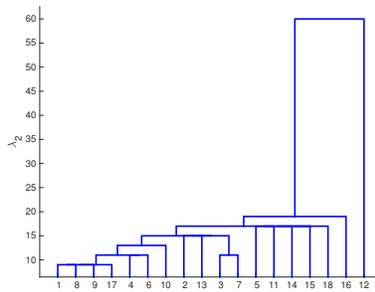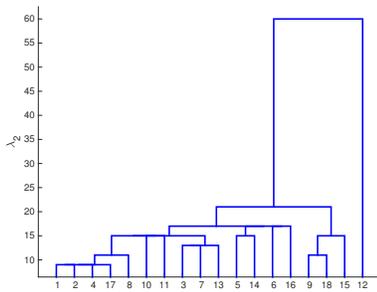
Figure 5: Tree structure of tasks (varieties) inferred using our approach for stalk diameter. **Figure 6: Tree structure of tasks (varieties) inferred using our approach for stalk volume.** **Figure 7: Tree structure of the tasks (height bins) inferred using our approach for the GWAS dataset.**

Bin 2   Bin 3   Bin 4   Bins 1,5-10
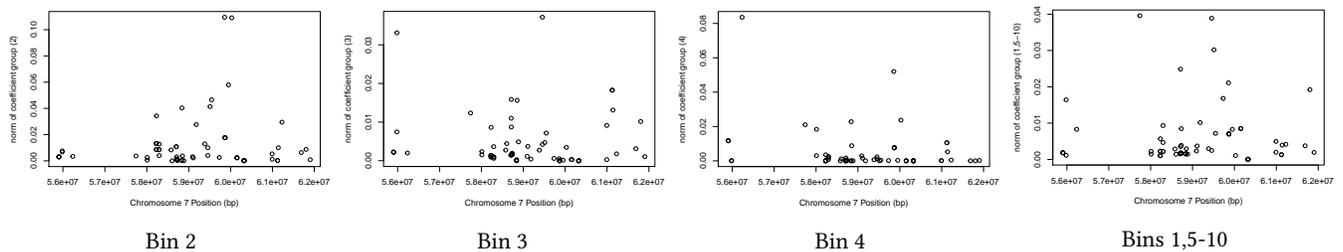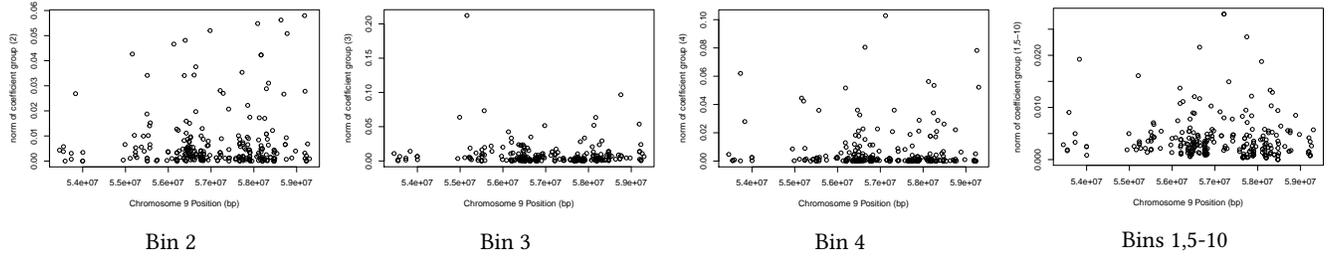
Figure 8: Chromosome 7: Regression coefficients norm for the bin groups uncovered by our approach. x-axis: position on the chromosome (in bp). y-axis: ℓ_2-norm of the regression coefficients within each group.

impactful application. In the future, we will explore other structured sparsity penalties, generalize the loss function to be other than squared ℓ_2, and pursue the applications of our method to high-throughput phenotyping further.

Figure 9: Chromosome 9: Regression coefficients norm for the bin groups uncovered by our approach. x-axis: position on the chromosome (in bp). y-axis: ℓ_2-norm of the regression coefficients within each group.

A PROOF OF PROPOSITION 3.1

We need to check that the conditions in Theorem 3.4 in [4] are satisfied in our case:

$$\text{(i)} \lim_{\|\Theta\| \to +\infty} f_1(\Theta) + f_2(\Theta) + \sum_{s,t} f_{st}(\Theta) = +\infty$$

$$\text{(ii)} (0, \ldots, 0) \in \text{sri}\{(\Theta - \Theta_{(1)}, \Theta - \Theta_{(2)}, \Theta - \Theta_{(1)} \ldots, x - x_m)|$$

$$\Theta \in \mathbb{R}^{pk}, \Theta_{(1)} \in \text{dom} f_1, \Theta_{(2)} \in \text{dom} f_2, \Theta_{(ij)} \in \text{dom} f_{ij}\}$$

Let $\mathcal{H}$ be the domain of Θ which can be set as $\mathbb{R}^{pk}$. Let C be a nonempty convex subset of $\mathcal{H}$, the strong relative interior of C is

$$\text{sri}(C) = \left\{ \Theta \in C | \text{cone}(C - \Theta) = \overline{\text{span}}(C - \Theta) \right\}$$

where $\text{cone}(C) = \bigcup_{\lambda > 0} \{\lambda \Theta | \Theta \in C\}$, and $\overline{\text{span}}(C)$ is the closure of span C.

Now we check the conditions. For (i), $\|\Theta\|$ goes to infinity means $\|\Theta_s\|$ goes to infinity, and then we know f_2 goes to infinity. Therefore (i) holds.

For (ii), we do not have any restriction on Θ, so the right hand side is just $\text{sri}(\mathbb{R}^{pk})$, hence (ii) holds.

Therefore, the proposition follows according to Theorem 3.4 of [4].

ACKNOWLEDGMENTS

REFERENCES

[1] Hanen Borchani, Gherardo Varando, Concha Bielza, and Pedro Larrañaga. 2015. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5, 5 (2015), 216–233.
[2] Rich Caruana. 1998. Multitask learning. In *Learning to learn*. Springer, 95–133.
[3] Eric C. Chi and Kenneth Lange. 2015. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics* 24, 4 (2015), 994–1013.
[4] Patrick L Combettes and Jean-Christophe Pesquet. 2008. A proximal decomposition method for solving convex variational inverse problems. *Inverse problems* 24, 6 (2008), 065014.
[5] Hal Daumé III. 2009. Bayesian multitask learning with latent hierarchies. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 135–142.
[6] Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 109–117.
[7] Toby Dylan Hocking, Armand Joulin, Francis Bach, and Jean-Philippe Vert. 2011. Clusterpath an algorithm for clustering using convex fusion penalties. In *28th international conference on machine learning*. 1.
[8] Toby Dylan Hocking, Armand Joulin, Francis Bach, and Jean-Philippe Vert. 2011. Clusterpath An Algorithm for Clustering using Convex Fusion Penalties. In *International Conference on Machine learning (ICML)*.
[9] Laurent Jacob, Jean-philippe Vert, and Francis R Bach. 2009. Clustered multi-task learning: A convex formulation. In *Advances in neural information processing systems*. 745–752.
[10] Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep K Ravikumar. 2010. A dirty model for multi-task learning. In *Advances in Neural Information Processing Systems*. 964–972.
[11] Zhuoliang Kang, Kristen Grauman, and Fei Sha. 2011. Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 521–528.
[12] Seyoung Kim and Eric P. Xing. 2010. Tree-Guided Group Lasso for Multi-Task Regression with Structured Sparsity. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 543–550.
[13] Keith Knight and Wenjiang Fu. 2000. Asymptotics for lasso-type estimators. *Annals of Statistics* 28 (2000), 1356–1378.
[14] Abhishek Kumar and Hal Daume III. 2012. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417* (2012).
[15] Fredrik Lindsten, Henrik Ohlsson, and Lennart Ljung. 2011. Just Relax and Come Clustering! A Convexication of k-Means Clustering. In *Tech. rep*. Linköpings universitet.
[16] Yichao Lu and Dean P Foster. 2014. Fast ridge regression with randomized principal component analysis and gradient descent. *arXiv preprint arXiv:1405.3952* (2014).
[17] Brian McWilliams, Christina Heinze, Nicolai Meinshausen, Gabriel Krummenacher, and Hastagiri P Vanchinathan. 2014. LOCO: Distributing ridge regression with random projections. *stat* 1050 (2014), 26.
[18] Guillaume Obozinski, Ben Taskar, and Michael Jordan. 2006. Multi-task feature selection. *Statistics Department, UC Berkeley, Tech. Rep* 2 (2006).
[19] K.N. Ramamurthy, Z. Zhang, A. M. Thompson, F. He, M. M. Crawford, A. F. Habib, C. F. Weil, and M. R. Tuinstra. 2016. Predictive Modeling of Sorghum Phenotypes with Airborne Image Features. In *Proc. of KDD Workshop on Data Science for Food, Energy, and Water*.
[20] Guilherme V. Rocha, Xing Wang, and Bin Yu. 2009. Asymptotic distribution and sparsistency for l1-penalized parametric M-estimators with applications to linear SVM and logistic regression. *ArXiv preprint* (2009).
[21] Elizabeth D Schifano, Lin Li, David C Christiani, and Xihong Lin. 2013. Genome-wide association analysis for multiple continuous secondary phenotypes. *The American Journal of Human Genetics* 92, 5 (2013), 744–759.
[22] Grzegorz Swirszcz and Aurelie C Lozano. 2012. Multi-level lasso for sparse multi-task regression. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*. 361–368.
[23] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B* (2005), 91–108.
[24] M. R. Tuinstra et al. 2016. Automated Sorghum Phenotyping and Trait Development Platform. In *Proc. of KDD Workshop on Data Science for Food, Energy, and Water*.