Learning Multi-Level Task Groups in Multi-Task Learning

Lei Han[1] **and Yu Zhang**[1,2*]
[1]Department of Computer Science, Hong Kong Baptist University, Hong Kong
[2]The Institute of Research and Continuing Education, Hong Kong Baptist University (Shenzhen)

Abstract

In multi-task learning (MTL), multiple related tasks are learned jointly by sharing information across them. Many MTL algorithms have been proposed to learn the underlying task groups. However, those methods are limited to learn the task groups at only a single level, which may be not sufficient to model the complex structure among tasks in many real-world applications. In this paper, we propose a Multi-Level Task Grouping (MeTaG) method to learn the multi-level grouping structure instead of only one level among tasks. Specifically, by assuming the number of levels to be H, we decompose the parameter matrix into a sum of H component matrices, each of which is regularized with a ℓ_2 norm on the pairwise difference among parameters of all the tasks to construct level-specific task groups. For optimization, we employ the smoothing proximal gradient method to efficiently solve the objective function of the MeTaG model. Moreover, we provide theoretical analysis to show that under certain conditions the MeTaG model can recover the true parameter matrix and the true task groups in each level with high probability. We experiment our approach on both synthetic and real-world datasets, showing competitive performance over state-of-the-art MTL methods.

Introduction

Multi-task learning (MTL) (Caruana 1997) seeks to improve the generalization performance of multiple learning tasks by sharing common information among them. MTL has gained its popularity among a wide range of applications including image annotation (Fan, Gao, and Luo 2008), speech recognition (Parameswaran and Weinberger 2010), disease progression predication (Zhou et al. 2011) and so on.

Many MTL algorithms have been proposed to learn task structure and model parameters from data simultaneously. For example, some works aim to identify the existent of the outlier tasks (Chen, Zhou, and Ye 2011; Gong, Ye, and Zhang 2012), some assume that the model parameters of all the tasks lies in a low dimensional subspace (Ando and Zhang 2005; Chen, Liu, and Ye 2010; Chen, Zhou, and Ye 2011), some works learn the task relations (Zhang, Yeung, and Xu 2010; Zhang and Schneider 2010; Zhang and Yeung 2010; Zhang 2013; Zhang and Yeung 2014), and some assume the task structure is hierarchical (Daumé III 2009; Jalali et al. 2010; Görnitz et al. 2011; Lozano and Swirszcz 2012; Zweig and Weinshall 2013). Among them, some interesting MTL algorithms assume that the tasks form several clusters and aim to learn the underlying task groups. For example, Jacob et al. (2008) design a regularizer based on the k-means clustering algorithm to directly learn the task clusters with the number of cluster predefined, Kang et al. (2011) identify the task groups by learning the cluster assignments based on relaxed integer programming, Kumar and Daume III (2012) generalize the non-overlapping cluster assignments for tasks to overlapping ones, and Zhong and Kwok (2012) focus on learning feature-level task groups where different features in one task can have different task groups.

However, all the existing task grouping techniques are limited to learn the task groups at only a single level based on the model parameters but in real-world applications, the structure between tasks can be so complex that the single-level task grouping is not enough to model it. For example, to find the cross-talks from gene expressions, the correlated genes are often interacted with multi-level clusters as studied in (Kim and Xing 2010; Han et al. 2014). However, we are not aware of any work which can learn the multi-level task clusters from data automatically.

In this paper, we want to fill this gap by learning multi-level task groups as well as the parameters learning to model the complex task relations. Specifically, we propose a Multi-Level Task Grouping (MeTaG) method which decomposes the parameter matrix (i.e., a matrix containing the model parameters of all the tasks) into a sum of H component matrices with each component matrix corresponding to one level. In order to learn the task groups in each level, we impose a ℓ_2 norm on the pairwise difference among the column vectors (corresponding to tasks) of each component matrix to construct level-specific task groups without the need to predefine the number of the groups. The proposed objective function is convex but non-smooth, and the smoothing proximal gradient method (Chen et al. 2011) is employed to seek the global optimum efficiently. Moreover, we provide theoretical analysis for the proposed MeTaG method by proving a error bound between the estimation by our MeTaG method

*Both authors contribute equally.

and the ground truth. We further show that with an assumption on the noise for the true grouping pattern, our MeTaG method can recover the the true task groups in each level with high probability. For empirical studies, we compare our MeTaG method with some state-of-the-art MTL methods on both synthetic and real datasets, and the experimental results demonstrate that the proposed MeTaG method is competitive compared with existing MTL methods.

Notations: Lower-case letters are used for scalars, boldface, lower-case letters refer to vectors, and bold-face and capital letters are for matrices. A vector $\mathbf{x}$ with length m is denoted by $\mathbf{x} \in \mathbb{R}^m$ and similarly a matrix $\mathbf{X}$ with size $d \times m$ is represented as $\mathbf{X} \in \mathbb{R}^{d\times m}$. For a matrix $\mathbf{X}$, its jth row, ith column, and (j,i)th element are represented as $\mathbf{x}^j$, $\mathbf{x}_i$, and x_{ji} respectively. For any vector $\mathbf{x}$, $\|\mathbf{x}\|_q$ represents its ℓ_q norm. For any matrix $\mathbf{X}$, $\|\mathbf{X}\|_{p,q}$ and $\|\mathbf{X}\|_F$ represent its $\ell_{p,q}$ norm and Frobenius norm separately. $\langle \mathbf{X}, \mathbf{Y}\rangle$ denotes the inner product of any matrices (or vectors) $\mathbf{X}$ and $\mathbf{Y}$. $\mathbb{N}_m$ represents the set of integers $\{1, \cdots, m\}$. $\mathcal{N}(\mu, \sigma^2)$ represents a normal distribution with mean μ and variance σ^2.

The MeTaG Model

Suppose we have m learning tasks and the feature dimensionality is d. The training data for the ith task is denoted by $(\mathbf{X}_i, \mathbf{y}_i)$, where $\mathbf{X}_i \in \mathbb{R}^{n_i\times d}$ is the data matrix with n_i training samples stored in the rows, and $\mathbf{y}_i \in \mathbb{R}^{n_i}$ is a vector of class labels for the n_i training samples in $\mathbf{X}_i$. If the values in $\mathbf{y}_i$ are continuous, the ith task is a regression problem and otherwise a classification problem. Each column in $\mathbf{X}_i$ ($i \in \mathbb{N}_m$) corresponding to one feature is assumed to be normalized with zero mean and unit variance:

$$\sum_{k=1}^{n_i} x_{kj}^{(i)} = 0,\ \sum_{k=1}^{n_i} \left(x_{kj}^{(i)}\right)^2 = 1,\ \forall j \in \mathbb{N}_d, i \in \mathbb{N}_m, \qquad (1)$$

where $x_{kj}^{(i)}$ is the (k,j)th element in matrix $\mathbf{X}_i$. The linear function for the ith task is defined as $\mu_i(\mathbf{x}) = \mathbf{w}_i^T\mathbf{x},\ i \in \mathbb{N}_m$, where an offset is assumed to be absorbed into $\mathbf{w}_i$. Define $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_m] \in \mathbb{R}^{d\times m}$ as the parameter matrix.

Since we aim to learn multi-level task groups, by assuming that there are H levels where H is a user-defined parameter, we decompose the parameter matrix $\mathbf{W}$ into the sum of H component matrices each of which is to learn the task groups in a level. Specifically, the parameter matrix $\mathbf{W}$ is decomposed as

$$\mathbf{W} = \sum_{h=1}^{H} \mathbf{W}_h. \qquad (2)$$

In Eq. (2), $\mathbf{W}_h = [\mathbf{w}_{h,1}, \cdots, \mathbf{w}_{h,m}] \in \mathbb{R}^{d\times m}$ is the component matrix corresponding to the hth level and $\mathbf{w}_{h,i}$ is the parameter for the ith task in the hth level. Then we formulate the objective function of the MeTaG method as

$$\min_{\mathbf{W}} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n_i}\|\mathbf{y}_i - \mathbf{X}_i\sum_{h=1}^{H}\mathbf{w}_{h,i}\|_2^2 + \sum_{h=1}^{H}\lambda_h\sum_{i<j}^{m}\|\mathbf{w}_{h,i} - \mathbf{w}_{h,j}\|_2, \qquad (3)$$

where λ_h's are positive regularization parameters. The first term in problem (3) measures the averaged square loss on the training data. By denoting the second term in problem (3) by $\Omega(\mathbf{W})$, we observe that $\Omega(\mathbf{W})$ imposes a ℓ_2 norm on the pairwise difference among the column vectors in $\mathbf{W}_h$, which encourages each pair of columns $\mathbf{w}_{h,i}$ and $\mathbf{w}_{h,j}$ in $\mathbf{W}_h$ to be identical. If this happens, then the ith and jth tasks belong to a task group in the hth level. λ_h controls the strength of task grouping at the hth level, and a larger λ_h is likely to lead to smaller number of task groups in the hth level. When $\lambda_h \to \infty$, there will be only one task group with all identical columns in $\mathbf{W}_h$. By assuming a descending order for the numbers of task groups from the Hth level to the first one, we set $\lambda_h = \lambda_{h-1}/\phi$ for $h \geq 2$ with constant $\phi > 1$.

It is worth mentioning that $\Omega(\mathbf{W})$ differs from the fused lasso regularizer (FLR) (Tibshirani et al. 2005) and its variant, generalized fused lasso regularizer (GFLR) (Friedman et al. 2007). The FLR and GFLR enable to group data features in terms of scalars, while $\Omega(\mathbf{W})$ is for task grouping in terms of column vectors. Therefore, $\Omega(\mathbf{W})$ can be viewed as a generalization of the FLR and GFLR. Note that solving an optimization problem regularized by $\Omega(\mathbf{W})$ is more challenged than that with the FLR and GFLR. With the same reason, $\Omega(\mathbf{W})$ differs from the feature-level task grouping regularizer in (Zhong and Kwok 2012).

Problem (3) is not easy to solve due to the non-smoothness of $\Omega(\mathbf{W})$. In the next section, we show how to solve problem (3) efficiently.

Optimization Procedure

Both the square loss and the regularizer $\Omega(\mathbf{W})$ are convex with respect to $\mathbf{W}$, making problem (3) convex. Since learning all levels simultaneously involves a large number of parameters, we propose to decompose problem (3) into several subproblems corresponding to the levels. We then develop a bottom-up iterative scheme, an instance of the coordinate descent method, where the subproblem corresponding to the hth level in the $(k+1)$th iteration seeks for $\mathbf{W}_h^{k+1}$ by solving the following problem as

$$\min_{\mathbf{W}_h} \sum_{i=1}^{m}\frac{1}{mn_i}\|\widetilde{\mathbf{y}}_i - \mathbf{X}_i\mathbf{w}_{h,i}\|_2^2 + \lambda_h\sum_{i<j}^{m}\|\mathbf{w}_{h,i} - \mathbf{w}_{h,j}\|_2, \quad (4)$$

with $\widetilde{\mathbf{y}}_i = \mathbf{y}_i - \mathbf{X}_i\left(\sum_{h'<h}\mathbf{w}_{h',i}^{k+1} + \sum_{h''>h}\mathbf{w}_{h'',i}^{k}\right)$ defined based on the parameters of the other levels from their last updates. The bottom-up iterative scheme is shown in Algorithm 1. Since the second term in problem (4) is non-smooth, we employ the smoothing proximal gradient (SPG) method (Chen et al. 2011) to solve problem (4). The problem solved by the SPG method takes the form

$$\min_{\mathbf{Z}} f(\mathbf{W}) + r(\mathbf{Z}), \qquad (5)$$

where $f(\cdot)$ is convex and Lipschitz continuous, and $r(\cdot)$ is convex but non-smooth.

In order to employ the SPG method, we use $f(\cdot)$ and $r(\cdot)$ to represent the first and the second terms in problem (4) respectively. Then we can rewrite $r(\cdot)$ as

$$r(\mathbf{W}_h) = \lambda_h\sum_{i<j}^{m}\|\mathbf{w}_{h,i} - \mathbf{w}_{h,j}\|_2 = \|\mathbf{C}\mathbf{W}_h^T\|_{1,2}, \qquad (6)$$

where $\mathbf{C} \in \mathbb{R}^{\frac{m(m-1)}{2}\times m}$ is a sparse matrix with each row having only two non-zero entries λ_h and $-\lambda_h$ in two corresponding positions. Therefore, the storage requirement of

Algorithm 1 The Bottom-Up Iterative Scheme for Problem (3).

Input: $\mathbf{X}$, $\mathbf{Y}$;
Output: $\mathbf{W}$;

Initialize $k = 0$, $\mathbf{W}_1^0 = \cdots = \mathbf{W}_H^0 = \mathbf{0}$;
repeat
  for $h = 1, \cdots, H$ **do**
    Solve problem (4);
  end for
  $k := k + 1$;
until Some convergence criterion is satisfied;

$\mathbf{C}$ is very small. Based on the definition of the dual norm, $r(\mathbf{W}_h)$ can be reformulated as

$$r(\mathbf{W}_h) = \max_{\mathbf{A} \in \mathcal{Q}} \langle \mathbf{C}\mathbf{W}_h^T, \mathbf{A} \rangle, \tag{7}$$

where $\mathbf{A} = \left(\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_{m(m-1)/2}\right)^T$ is the auxiliary matrix variable, $\boldsymbol{\alpha}_i$ is a vector of auxiliary variables corresponding to the ith row of $\mathbf{C}\mathbf{W}_h^T$, and $\mathcal{Q} = \{\mathbf{A} | \|\boldsymbol{\alpha}_i\|_2 \leq 1, \forall i \in \mathbb{N}_{m(m-1)/2}\}$ is the domain of $\mathbf{A}$. Then the smooth approximation of Eq. (7) is given by

$$g_\mu(\mathbf{W}_h) = \max_{\mathbf{A} \in \mathcal{Q}} \langle \mathbf{C}\mathbf{W}_h^T, \mathbf{A} \rangle - \mu d(\mathbf{A}), \tag{8}$$

where $d(\mathbf{A}) = \frac{1}{2}\|\mathbf{A}\|_F^2$. According to (Chen et al. 2011), problem (8) is convex and smooth with gradient $\nabla g_\mu(\mathbf{W}) = (\mathbf{A}^*)^T\mathbf{C}$, where $\mathbf{A}^*$ is the optimal solution to problem (8). The computation of $\mathbf{A}^*$ is depicted in the following proposition.

Proposition 1 *By denoting by* $\mathbf{A}^* = \left(\boldsymbol{\alpha}_1^*, \cdots, \boldsymbol{\alpha}_{m(m-1)/2}^*\right)^T$ *the optimal solution to problem (8), for any* $i \in \mathbb{N}_{(m-1)m/2}$, *we have*

$$\boldsymbol{\alpha}_i^* = S\left(\left[\mathbf{C}\mathbf{W}_h^T\right]^i / \mu\right), \tag{9}$$

where $[\mathbf{M}]^i$ *denotes the* i*th row of a matrix* $\mathbf{M}$ *and* $S(\cdot)$ *is the projection operator to project vector* $\mathbf{u}$ *on the* ℓ_2 *ball as*

$$S(\mathbf{u}) = \begin{cases} \frac{\mathbf{u}}{\|\mathbf{u}\|_2}, & \|\mathbf{u}\|_2 > 1, \\ \mathbf{u}, & \|\mathbf{u}\|_2 \leq 1. \end{cases}$$

Instead of directly solving problem (4), we solve its smooth approximation as

$$\min_{\mathbf{W}_h} \widetilde{f}(\mathbf{W}_h) = f(\mathbf{W}_h) + g_\mu(\mathbf{W}_h). \tag{11}$$

The gradient of $\widetilde{f}(\mathbf{W}_h)$ w.r.t. $\mathbf{W}_h$ can be computed as

$$\nabla_{\mathbf{W}_h} \widetilde{f}(\mathbf{W}_h) = \nabla_{\mathbf{W}_h} f(\mathbf{W}_h) + (\mathbf{A}^*)^T\mathbf{C}. \tag{12}$$

By using the square loss in problem (4), the ith column of $\nabla_{\mathbf{W}_h} f(\mathbf{W}_h)$ can be easily obtained as $\frac{2}{mn_i}\mathbf{X}_i^T(\mathbf{X}_i\mathbf{w}_{h,i} - \mathbf{y}_i)$. Moreover, it is easy to prove that $\widetilde{f}(\mathbf{W}_h)$ is L-Lipschitz continuous where L can be determined by numerical approaches (Chen et al. 2011). The whole SPG algorithm to solve problem (11) is depicted in Algorithm 2, where problem (10) has a closed-form solution as $\mathbf{W}_h^{(t+1)} = \widehat{\mathbf{W}}_h^{(t)} - \frac{1}{L}\nabla\widetilde{f}(\widehat{\mathbf{W}}_h^{(t)})$. Let $D = \max_{\mathbf{A} \in \mathcal{Q}} d(\mathbf{A})$ and $\mathbf{W}_h^*$ be the optimal solution of Eq. (4). If the desired accuracy is ε, i.e., $|\widetilde{f}(\mathbf{W}_h^{(t)}) - \widetilde{f}(\mathbf{W}_h^*)| \leq \varepsilon$, according to (Chen et al. 2011), Algorithm 2 needs $O(\sqrt{2D}/\varepsilon)$ iterations to converge. Moreover, in our experiments, we find that Algorithm 1 needs very few iterations to converge, making it very efficient.

Algorithm 2 SPG algorithm for solving problem (11).

Input: $\mathbf{X}$, $\widetilde{\mathbf{Y}}$, $\widehat{\mathbf{W}}^{(0)}$, μ, h, λ_h;
Output: $\mathbf{W}_h$;

Initialize $t = 0$ and $\tau_0 = 1$;
repeat
  Compute $\nabla\widetilde{f}(\widehat{\mathbf{W}}_h^{(t)})$ as in Eq. (12);
  Solve the proximal step:

$$\mathbf{W}_h^{(t+1)} = \arg\min_{\mathbf{W}_h} \widetilde{f}(\widehat{\mathbf{W}}_h^{(t)}) + \langle \mathbf{W}_h - \widehat{\mathbf{W}}_h^{(t)}, \nabla\widetilde{f}(\widehat{\mathbf{W}}_h^{(t)}) \rangle + \frac{L}{2}\|\mathbf{W}_h - \widehat{\mathbf{W}}_h^{(t)}\|_F^2. \tag{10}$$

  $\tau_{t+1} = \frac{2}{t+3}$;
  $\widehat{\mathbf{W}}_h^{(t+1)} = \mathbf{W}_h^{(t+1)} + \frac{1-\tau_t}{\tau_t}\tau_{t+1}(\mathbf{W}_h^{(t+1)} - \widehat{\mathbf{W}}_h^{(t)})$;
  $t := t + 1$;
until Some convergence criterion is satisfied;

Theoretical Analysis

In this section, we provide theoretical analysis for the proposed MeTaG model. For notational simplicity, we assume that the numbers of training samples for all the tasks are the same and denote it by n. The general case that different tasks have different numbers of training samples can be similarly analyzed. We assume that the true relation between the data sample and its class label is a linear function plus a Gaussian noise, which is defined as

$$y_{ji} = \left(\mathbf{x}_j^{(i)}\right)^T \mathbf{w}_i^* + \epsilon_{ji}, i \in \mathbb{N}_m, j \in \mathbb{N}_n, \tag{13}$$

where y_{ji} is the jth element in $\mathbf{y}_i$, $\mathbf{W}^* = [\mathbf{w}_1^*, \ldots, \mathbf{w}_m^*]$ is the true parameter matrix, ϵ_{ji} is a Gaussian noise, and $\mathbf{W}^*$ can be decomposed into the sum of H true component matrix $\mathbf{W}_1^*, \ldots, \mathbf{W}_H^*$ as $\mathbf{W}^* = \sum_{h=1}^H \mathbf{W}_h^*$. Each noise ϵ_{ji} follows a normal distribution, i.e., $\epsilon_{ji} \sim \mathcal{N}(0, \sigma^2)$, and all the noises are assumed to be independent of each other. We define $\mathbf{f}_i^* = \mathbf{X}_i\mathbf{w}_i^*$ and $\mathbf{y}_i = \mathbf{f}_i^* + \boldsymbol{\epsilon}_i$ for $i \in \mathbb{N}_m$, where $\boldsymbol{\epsilon}_i = [\epsilon_{1i}, \ldots, \epsilon_{ni}]^T$. Let $\mathbf{X} \in \mathbb{R}^{dm \times mn}$ be a block diagonal matrix with $\mathbf{X}_i^T \in \mathbb{R}^{d \times n} (i \in \mathbb{N}_m)$ as the ith block. We define a vectorization operator $\mathrm{vec}(\cdot)$ over an arbitrary matrix $\mathbf{P} \in \mathbb{R}^{d \times m}$ as $\mathrm{vec}(\mathbf{P}) = [\mathbf{p}_1^T, \cdots, \mathbf{p}_m^T]^T$ where $\mathbf{p}_i$ is the ith column of $\mathbf{P}$. Let $\mathbf{F}^* = [\mathbf{f}_1^*, \ldots, \mathbf{f}_m^*] \in \mathbb{R}^{n \times m}$.

For any matrix $\mathbf{Q} \in \mathbb{R}^{d \times m}$, we define $E(\mathbf{Q}) = \{(i,j) | \mathbf{q}_i \neq \mathbf{q}_j, i \in \mathbb{N}_m, j \in \mathbb{N}_m\}$ and its complement $E_c(\mathbf{Q}) = \{(i,j) | \mathbf{q}_i = \mathbf{q}_j, i \in \mathbb{N}_m, j \in \mathbb{N}_m, i \neq j\}$ as its true column grouping pattern, where $\mathbf{q}_i$ is the ith column of $\mathbf{Q}$. For any matrix $\mathbf{Q} \in \mathbb{R}^{d \times m}$, since each pair (i,j) corresponds to one row in $\mathbf{C}\mathbf{Q}^T \in \mathbb{R}^{\frac{m(m-1)}{2} \times d}$, which is $\mathbf{q}_i^T - \mathbf{q}_j^T$, with $\mathbf{C}$ defined in Eq. (6), the projections of the rows in $\mathbf{C}\mathbf{Q}^T$ on the set $E(\mathbf{Q})$, denoted by $\left(\mathbf{C}\mathbf{Q}^T\right)^{E(\mathbf{Q})}$, consist of the rows with non-zero ℓ_2 norms in $\mathbf{C}\mathbf{Q}^T$, and similarly the projections of the rows in $\mathbf{C}\mathbf{Q}^T$ on set $E_c(\mathbf{Q})$, denoted by $\left(\mathbf{C}\mathbf{Q}^T\right)^{E_c(\mathbf{Q})}$, are the zero rows in $\mathbf{C}\mathbf{W}_h^T$. We define $D(\mathbf{Q})$ as the index set of distinct non-zero column vectors in $\mathbf{Q}$, i.e. for any $i, j \in D(\mathbf{Q})$, $\mathbf{q}_i \neq \mathbf{q}_j$. Denote $\mathbf{Q}^{D(\mathbf{Q})}$ as the projection of the columns of $\mathbf{Q}$ on set $D(\mathbf{Q})$. Let $D_c(\mathbf{Q})$ be the complement of $D(\mathbf{Q})$. Now, in order to analyze our method, we need the following assumption.

Assumption 1 *Let $\hat{\mathbf{W}} = \sum_{h=1}^{H} \hat{\mathbf{W}}_h$ be an optimal solution of Eq. (3). For any matrix $\mathbf{W} = \sum_{h=1}^{H} \mathbf{W}_h \in \mathbb{R}^{d\times m}$ and $h \in \mathbb{N}_H$, we define matrix $\mathbf{\Delta}_h$ as $\mathbf{\Delta}_h = \mathbf{W}_h - \hat{\mathbf{W}}_h$ and matrix $\mathbf{\Gamma}_h$ as $\mathbf{\Gamma}_h = \mathbf{C}\mathbf{W}_h^T - \mathbf{C}\hat{\mathbf{W}}_h^T$. Let $\mathbf{\Delta} = \sum_{h=1}^{H} \mathbf{\Delta}_h$. We assume that there exist positive scalar β_h, and scalars $\theta_h \geq 1$ and $\gamma_h \geq 1$ such that*

$$\beta_h = \min_{\mathbf{\Delta}_h \neq \mathbf{0}} \frac{\|\mathbf{X}^T \mathrm{vec}(\mathbf{\Delta})\|_2}{\sqrt{mn}\|\mathbf{\Delta}_h^{D(\mathbf{W}_h)}\|_F},$$

$$\|\mathbf{\Delta}_h\|_F = \theta_h \|\mathbf{\Delta}_h^{D(\mathbf{W}_h)}\|_F, \ \ \|\mathbf{\Gamma}_h\|_{1,2} = \gamma_h \|\mathbf{\Gamma}_h^{E(\mathbf{W}_h)}\|_{1,2}.$$

Assumption 1 refers to the restricted eigenvalue assumption as introduced in (Lounici et al. 2009). Similar assumptions are commonly used in the MTL literature, e.g., (Chen, Zhou, and Ye 2011; Gong, Ye, and Zhang 2012). Note that $\theta_h = 1$ leads to $\gamma_h = 1$ and vice versa. Moreover, $\theta_h = 1$ if and only if $\mathbf{\Delta}_h^{D_c(\mathbf{W}_h)} = \mathbf{0}$ or $D_c(\mathbf{W}_h) = \emptyset$ which implies that all tasks differ from each other. Now, we present important theoretical results for MeTaG model in the following theorem.

Theorem 1 *Let $\hat{\mathbf{W}} = \sum_{h=1}^{H} \hat{\mathbf{W}}_h$ be an optimal solution of problem (3). If the regularization parameters λ_h for any $h \in \mathbb{N}_H$ satisfies* [1]

$$\lambda_h \geq \frac{2\sigma}{m(m-1)n}\sqrt{m + \frac{\delta}{d}}, \tag{14}$$

then under Assumption 1, the following results hold with probability of at least $1 - \exp(-\frac{1}{2}(\delta - dm\log(1 + \frac{\delta}{dm})))$:

$$\|\mathbf{X}^T \mathrm{vec}(\hat{\mathbf{W}}) - \mathrm{vec}(\mathbf{F}^*)\|_2^2 \leq m(m-1)^2 nd\mathcal{H}^2, \tag{15}$$

$$\|\hat{\mathbf{W}}_h - \mathbf{W}_h^*\|_F \leq \frac{\theta_h(m-1)\sqrt{d}\mathcal{H}}{\beta_h}, \tag{16}$$

$$\|\mathbf{C}\hat{\mathbf{W}}_h^T - \mathbf{C}(\mathbf{W}_h^*)^T\|_{1,2} \leq \frac{\gamma_h(m-1)^2 d\mathcal{H}}{\beta_h}, \tag{17}$$

where $\mathcal{H} = \sum_{h'=1}^{H} \frac{\lambda_{h'}(\theta_{h'}+1)}{\beta_{h'}}$. In addition, if the following condition holds for $h \in \mathbb{N}_H$:

$$\min_{(i,j)\in E(\mathbf{W}_h^*)} \left\| \left[\mathbf{C}(\mathbf{W}_h^*)^T\right]^{(i,j)} \right\|_2 > \frac{2d\gamma_h(m-1)^2\mathcal{H}}{\beta_h}, \tag{18}$$

where $\left[\mathbf{C}(\mathbf{W}_h^*)^T\right]^{(i,j)}$ denotes one row in $\mathbf{C}(\mathbf{W}_h^*)^T$ corresponding to the pair (i,j), then with the probability of at least $1 - \exp(-\frac{1}{2}(\delta - dm\log(1 + \frac{\delta}{dm})))$, the following set

$$\hat{E}_h = \left\{ (i,j) \left| \left\| \left(\mathbf{C}\hat{\mathbf{W}}_h\right)^{(i,j)} \right\|_2 > \frac{d\gamma_h(m-1)^2\mathcal{H}}{\beta_h} \right. \right\} \tag{19}$$

can recover the true pattern of task groups $E(\mathbf{W}_h^*)$ in the h-th level, i.e. $\hat{E}_h = E(\mathbf{W}_h^*)$ and $(\hat{E}_h)_c = E_c(\mathbf{W}_h^*)$.

Remark 1 *Theorem 1 provides important theoretical guarantee for the MeTaG model. Specifically, those bounds measure how well our model can approximate the ground truth of the component matrix $\mathbf{W}_h^*$ in each level as well as the true parameter matrix $\mathbf{W}^* = \sum_{h=1}^{H} \mathbf{W}_h^*$. Moreover, with an assumption on the noise of the underlying grouping pattern in the true component matrices in Eq. (18), the MeTaG model can estimate the true grouping pattern for each level with high probability based on Eq. (19).*

[1] Since we assume a descending order for the λ_h's from λ_1 to λ_H, we only have to make λ_H satisfy Eq. (14).

Remark 2 *In the robust multi-task learning (rMTL) model (Gong, Ye, and Zhang 2012), the parameter matrix $\mathbf{W}$ is decomposed into two components $\mathbf{W}_1$ and $\mathbf{W}_2$, and the estimation error bound is also derived for their model. By setting $H = 2$ in our MeTaG model, the error bound in Eq. (15) is considerably better than that of their model especially when the feature dimension and number of tasks are large.*

Related Work

Our work is related to some MTL approaches (Jalali et al. 2010; Zweig and Weinshall 2013), since they assume the task relations can be represented by multiple hierarchies, which in some aspect is a bit similar to the multi-level structure in the proposed MeTaG method. However, those works do not learn task groups, which is one of the main concerns in this paper.

Experiments

In this section, we conduct empirical experiments on both synthetic and real-world problems to study the proposed MeTaG method. Baselines used for comparison include a wide range of competitive MTL models: the multi-task feature learning (MTFL) model (Liu, Ji, and Ye 2009), the dirty model (DM) (Jalali et al. 2010), the Cascade model (Zweig and Weinshall 2013), the clustered multi-task learning (CMTL) model (Jacob, Bach, and Vert 2008), the model that learns with whom to share (Whom) (Kang, Grauman, and Sha 2011), and the grouping and overlap MTL (GO-MTL) model (Kumar and Daume III 2012).

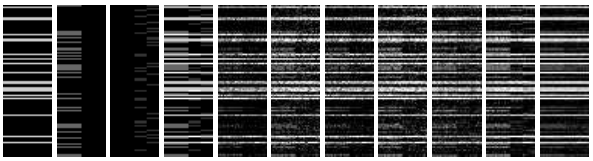
Synthetic Data

We first evaluate our method on synthetic data. We simulate a multi-task regression problem with $m = 32$ and $d = 100$. For the ith task, each column of $\mathbf{X}_i \in \mathbb{R}^{n\times d}$ is generated from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, where $\mathbf{I}_n$ is an $n \times n$ identity matrix. We assume there are three levels, i.e. $H = 3$. For the first level, we assume all the tasks are in the same group and randomly choose 20 rows of $\mathbf{W}_1^*$ corresponding to features to be non-zero with value 0.8. In the second level, we assume the first 16 tasks are in the same group and randomly select 30 rows of $\mathbf{W}_2^*$ such that the sub-matrix defined by the 30 rows and the first 16 columns is non-zero with all the elements equal to 0.4. For the third level, we assume the 17th-24th tasks are in one group and the 25th-32th tasks in another group. By randomly selecting 20 rows for the 17th-24th and 25th-32th columns in $\mathbf{W}_3^*$ separately, the elements in the selected sub-matrices are set to 0.2. Then the true parameter matrix $\mathbf{W}^*$ is generated by $\mathbf{W}^* = \mathbf{W}_1^* + \mathbf{W}_2^* + \mathbf{W}_3^*$. The decreasing order of the weights over levels can simulate the gradually decreased task sharing over levels. The response $\mathbf{y}_i$ for the ith task is then generated as $\mathbf{y}_i = \mathbf{X}_i\mathbf{w}_i^* + \boldsymbol{\epsilon}_i$, where $\boldsymbol{\epsilon}_i$ is a noise vector generated from $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_n)$ with $\sigma = 2$. The matrices $\mathbf{W}_1^*$, $\mathbf{W}_2^*$, $\mathbf{W}_3^*$, and $\mathbf{W}^*$ are depicted in Figures 1(a)-1(d) where columns correspond to tasks, rows represent features, and black entries denote zero elements.

We use the mean square error (MSE) to measure the performance of the estimation, which is defined as $\mathrm{MSE}(\mathbf{W}) =$

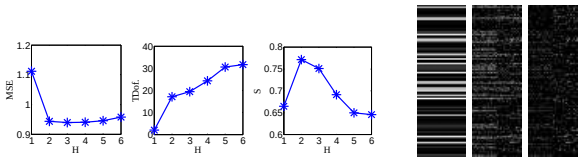
Table 1: The performance of various methods over 10 simulations on the synthetic data in terms of mean±standard deviation.

Training size		MTFL	DM	Cascade	CMTL	Whom	GO-MTL	MeTaG
50	MSE	5.972±0.295	4.816±0.280	6.538±0.437	3.164±0.137	2.639±0.191	2.082±0.074	**1.488±0.040**
	TDof	32.00±0.00	32.00±0.00	32.00±0.00	32.00±0.00	32.00±0.00	19.00±1.15	**17.00±4.03**
	S	0.645±0.000	0.645±0.000	0.645±0.000	0.645±0.000	0.645±0.00	0.674±0.010	**0.685±0.024**
100	MSE	3.470±0.133	3.212±0.139	3.216±0.217	2.538±0.143	2.282±0.155	1.482±0.171	**1.118±0.042**
	TDof	32.00±0.00	32.00±0.00	32.00±0.00	32.00±0.00	32.00±0.00	26.50±1.27	**11.20±4.24**
	S	0.645±0.000	0.645±0.000	0.645±0.000	0.645±0.000	0.645±0.00	0.662±0.006	**0.796±0.060**
150	MSE	2.625±0.074	2.388±0.090	2.137±0.118	1.810±0.074	2.247±0.200	1.046±0.083	**0.939±0.030**
	TDof	30.80±1.23	32.00±0.00	32.00±0.00	32.00±0.00	32.00±0.00	25.20±1.55	**19.40±3.78**
	S	0.648±0.003	0.645±0.000	0.645±0.000	0.645±0.000	0.645±0.00	0.666±0.008	**0.751±0.053**



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k)

Figure 1: Parameter matrices in synthetic data when the training size is 150, where (a)-(d) are the true component matrices and parameter matrix, and (e)-(j) are the estimators $\widehat{\mathbf{W}}$ from different methods: (a) $\mathbf{W}_1^*$; (b) $\mathbf{W}_2^*$; (c) $\mathbf{W}_3^*$; (d) $\mathbf{W}^*$; (e) MTFL; (f) DM; (g) Cascade; (h) CMTL; (i) Whom; (j) GO-MTL; (k) MeTaG.



(a) (b) (c) (d) (e) (f)

Figure 2: (a)-(c) The performance of the MeTaG method by varying H in the synthetic data when the training size is 150; (d)-(f) recovered components by the MeTaG method when $H = 3$.

$\frac{1}{mn}\sum_{i=1}^{m}(\mathbf{w}_i - \mathbf{w}_i^*)^T\mathbf{X}^T\mathbf{X}(\mathbf{w}_i - \mathbf{w}_i^*)$. We introduce two metrics, degree of freedom for tasks (TDof) and metric S, to measure the performance of task grouping. TDof is originated from the degree of freedom used in the feature grouping literatures such as (Bondell and Reich 2008), and is defined as the number of 'unequal' columns in $\mathbf{W}$, where two columns $\mathbf{w}_i$ and $\mathbf{w}_j$ are defined to be 'equal', denoted by $\mathbf{w}_i \simeq \mathbf{w}_j$, if $\|\mathbf{w}_i - \mathbf{w}_j\|_2 \leq \varepsilon_0$ for some threshold constant ε_0. We set $\varepsilon_0 = 0.6$ in the experiments, which shows a better discrimination. Although the task group structures in different levels are different, there are totally *three* task groups in the parameter matrix, which is shown in Figure 1(d). Therefore the closer the estimation of TDof is to 3, the better performance the corresponding method achieves. The second metric S is defined as $S = \frac{\sum_{i\neq j,\mathbf{w}_i^*\simeq\mathbf{w}_j^*}\mathbb{I}(\mathbf{w}_i\simeq\mathbf{w}_j)+\sum_{i\neq j,\mathbf{w}_i^*\not\simeq\mathbf{w}_j^*}\mathbb{I}(\mathbf{w}_i\not\simeq\mathbf{w}_i)}{\sum_{i\neq j,\mathbf{w}_i^*\simeq\mathbf{w}_j^*}1+\sum_{i\neq j,\mathbf{w}_i^*\not\simeq\mathbf{w}_j^*}1}$, where $\mathbb{I}(\cdot)$ is the indicator function. S is also motivated from the measurement for feature grouping as introduced in (Yang et al. 2012). The numerator in S consists of two parts, where the first and second terms represent the recovery of 'equal' columns and 'unequal' columns separately. The denominator is the sum of the exact number of 'equal' columns and the number of 'unequal' columns in the true parameter matrix. Thus, S can provide a measurement for the performance of task grouping. We vary the number of training samples from 50 to 150 to test the sensitivity of each method. In each experimental setting, we always generate 100 samples for testing and another 100 samples for parameter validation to select the hyperparameters in all the methods in comparison, including the number of groups in CMTL and Whom, the number of latent tasks in GO-MTL, the number of cascades in Cascade, and the number of levels in MeTaG, whose candidate values are from a set $\{1, \cdots, 10\}$. We set $\phi = 1.2$ in all the experiments, and choose λ_1 of the MeTaG model from the set $[10^{-6}, 10^{-5}, \cdots, 10^3]$.

Table 1 shows the performance of all the methods over 10 simulations. As shown in Table 1, the task grouping algorithms, i.e. the CMTL, Whom, GO-MTL, and MeTaG methods, have lower estimation errors compared with others. Our MeTaG method has the best performance in terms of all three performance measures. Figures 1(e)-1(k) show the estimated $\widehat{\mathbf{W}}$'s of all the methods in comparison when the training size is 150. By comparing Figures 1(e)-1(k) with the ground truth shown in Figure 1(d), the estimation learned from our MeTaG method has better recovery result. We also examine the performance of the MeTaG method when the value of H varies. Figures 2(a)-2(c) shows the change of MSE, TDof, and S of the MeTaG method by varying H. We observe that at the beginning when H increases, the performance improves in terms of MSE and S. When H reaches 2, the MeTaG method reaches the best S. When H reaches 3, which is just the true number of levels, the MeTaG method has the best MSE. Then the performance becomes worse for larger H's. Moreover, Figure 2(b) shows that a lower H leads to a lower TDof. One reason is the upper levels learned from the MeTaG method tend to have a larger number of small task groups due to the setting for the regularization parameters $\lambda_h = \lambda_{h-1}/\phi < \lambda_{h-1}$ and so when H increases, all the levels will contain more task groups, which leads to a higher TDof. Moreover, we plot the three component matrices learned by the MeTaG method when $H = 3$ in Figures 2(d)-2(f) and by comparing with the ground truth in Figures 1(a)-1(c), we can see that the recovery is good.

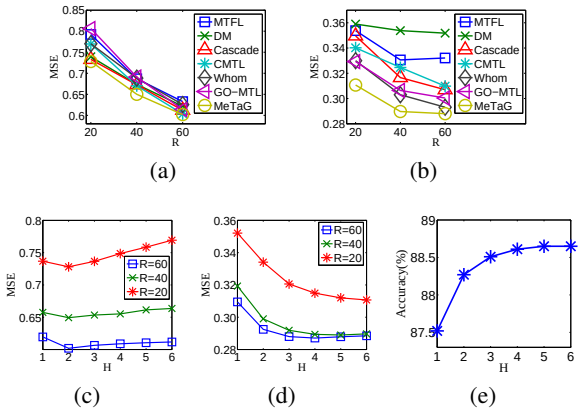


Figure 3: (a)-(b) Averaged MSE vs. R in the microarray data and traffic data respectively; (c)-(e) Performance of MeTaG vs. H in the microarray data, traffic data and handwritten data respectively.

Microarray Data

We report results on microarray data (Wille et al. 2004). The data is a gene expression dataset with microarray data related to isoprenoid biosynthesis in plant organism. The tasks are regression problems which aim to find the cross-talks from the expression levels of 21 genes in the mevalonate pathway (data features) to the expression levels of 18 genes in the plastidial pathway (labels). There are 118 samples and all the data are log-transformed and standardized to have zero mean and unit variance. We perform 10 random splits, each of which uses R%, (80-R)%, and 20% samples for training, testing and validation separately with R as the training ratio. We vary R to test the performance of the compared methods. Figure 3(a) shows the averaged MSE over 10 random splits under different training ratios. As shown in Figure 3(a), our MeTaG method stably achieves the best performance under all the settings. Moreover, Figure 3(c) shows the performance of the MeTaG method when H changes. We can observe that the MeTaG method achieves the best performance when $H = 2$ for all training ratios, which may imply that the number of the true task levels in the data is 2.

Traffic Data

In this experiment, a traffic data is used to compare different methods. The samples in this dataset are vehicle counts collected from 272 sensors placed in a highway traffic network, where one half of the sensors (i.e., 136 sensors) are placed in the exits of the highway and the others are in the entries of the highway. The tasks here are regression problems to find the casual relationships between the vehicle flows from the entries to the exits, where each exit corresponds to one task and the information collected in entries is considered as the data matrix shared by all the tasks. Previous analysis (Han et al. 2012) has shown that such casual relationships in highway traffic networks are likely to be grouped. There are 384 samples for each sensor. Each feature in the data is normalized to have zero mean and unit variance. The averaged MSE's over 10 random splits under different training ratios are reported in Figure 3(b). Again, our MeTaG method performs the best under all the training ratios. Figure 3(d) shows the performance of MeTaG when H changes. We can see that its performance improves when H increases. When H reaches 5 or 6, the improvement in the performance becomes small for all the training ratios and this can be viewed as an indicator for the true number of the levels.

Table 2: The averaged classification accuracy (%) of various methods on the handwritten letter data.

	c/e	g/y	m/n	a/g	a/o	f/t	h/n
MTFL	88.74	72.54	90.16	93.99	92.21	80.95	93.39
DM	88.66	74.41	89.61	93.79	92.39	80.57	92.78
Cascade	88.69	**74.96**	90.04	93.98	92.39	82.84	**94.40**
CMTL	88.31	74.22	86.71	93.70	92.39	79.91	94.18
Whom	**89.34**	73.83	90.84	**94.09**	**92.71**	83.17	**94.62**
GO-MTL	88.69	72.54	89.19	93.47	92.04	80.60	92.97
MeTaG	**89.39**	**74.70**	**91.12**	**94.17**	**92.84**	**83.69**	**94.63**

Handwritten Letter Data

In the handwritten letter dataset, there are seven tasks each of which is a binary classification problem to discriminate between two letters: c/e, g/y, m/n, a/g, a/o, f/t and h/n. We use the square loss for all the methods. Each data sample consists of 128 features representing the pixel values of the handwritten letter. For each task, there are about 1000 positive samples and 1000 negative samples. We randomly choose 10%, 20%, and 70% of the data for training, validation and testing. The averaged classification accuracy over 10 random splits are shown in Table 2. The highlighted numbers stand for the best results under the significance t-test with 95% confidence. From Table 2, we see that the MeTaG method shows competitive performance in all the tasks. Figure 3(e) shows the averaged accuracy of all the seven tasks of MeTaG when H changes. Similar to that in the traffic data, the performance of the MeTaG method improves when H increases, and becomes stable when H reaches 5 or 6.

Conclusion and Future Work

In this paper, we proposed a novel MeTaG model to learn multi-level task groups in multi-task learning. Efficient algorithms and performance bounds are derived for the MeTaG model. Experimental results conducted on both synthetic and real-world datasets demonstrate the effectiveness of the proposed method. At current stage, the number of the levels needs to be predefined. In future work, we are interested in learning the number of levels from data automatically.

Acknowledgment

This work is supported by NSFC 61305071 and HKBU FRG2/13-14/039. We thank Prof. Kunqing Xie and Prof. Guojie Song for providing the traffic data and valuable discussions.

References

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

Bondell, H. D., and Reich, B. J. 2008. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics* 64(1):115–123.

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

Chen, X.; Lin, Q.; Kim, S.; Carbonell, J. G.; and Xing, E. P. 2011. Smoothing proximal gradient method for general structured sparse regression. In *The Conference on Uncertainty in Artificial Intelligence*.

Chen, J.; Liu, J.; and Ye, J. 2010. Learning incoherent sparse and low-rank patterns from multiple tasks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Chen, J.; Zhou, J.; and Ye, J. 2011. Integrating low-rank and group-sparse structures for robust multi-task learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Daumé III, H. 2009. Bayesian multitask learning with latent hierarchies. In *The Conference on Uncertainty in Artificial Intelligence*.

Fan, J.; Gao, Y.; and Luo, H. 2008. Integrating concept ontology and multitask learning to achieve more effective classifier training for multilevel image annotation. *IEEE Transactions on Image Processing* 17(3):407–426.

Friedman, J.; Hastie, T.; Höfling, H.; and Tibshirani, R. 2007. Pathwise coordinate optimization. *The Annals of Applied Statistics* 1(2):302–332.

Gong, P.; Ye, J.; and Zhang, C. 2012. Robust multi-task feature learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Görnitz, N.; Widmer, C. K.; Zeller, G.; Kahles, A.; Rätsch, G.; and Sonnenburg, S. 2011. Hierarchical multitask structured output learning for large-scale sequence segmentation. In *Advances in Neural Information Processing Systems*.

Han, L.; Song, G.; Cong, G.; and Xie, K. 2012. Overlapping decomposition for causal graphical modeling. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Han, L.; Zhang, Y.; Song, G.; and Xie, K. 2014. Encoding tree sparsity in multi-task learning: A probabilistic framework. In *AAAI Conference on Artificial Intelligence*.

Jacob, L.; Bach, F.; and Vert, J.-P. 2008. Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems*.

Jalali, A.; Ravikumar, P.; Sanghavi, S.; and Ruan, C. 2010. A dirty model for multi-task learning. In *Advances in Neural Information Processing Systems*.

Kang, Z.; Grauman, K.; and Sha, F. 2011. Learning with whom to share in multi-task feature learning. In *International Conference on Machine Learning*.

Kim, S., and Xing, E. P. 2010. Tree-guided group lasso for multi-task regression with structured sparsity. In *International Conference on Machine Learning*.

Kumar, A., and Daume III, H. 2012. Learning task grouping and overlap in multi-task learning. In *International Conference on Machine Learning*.

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

Lounici, K.; Pontil, M.; Tsybakov, A. B.; and van de Geer, S. 2009. Taking advantage of sparsity in multi-task learning. In *Conference on Learning Theory*.

Lozano, A. C., and Swirszcz, G. 2012. Multi-level lasso for sparse multi-task regression. In *International Conference on Machine Learning*.

Parameswaran, S., and Weinberger, K. Q. 2010. Large margin multi-task metric learning. In *Advances in Neural Information Processing Systems*.

Tibshirani, R.; Saunders, M.; Rosset, S.; Zhu, J.; and Knight, K. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(1):91–108.

Wille, A.; Zimmermann, P.; Vranová, E.; Fürholz, A.; Laule, O.; Bleuler, S.; Hennig, L.; Prelić, A.; von Rohr, P.; Thiele, L.; Zitzler, E.; Gruissem, W.; and Bühlmann, P. 2004. Sparse graphical Gaussian modeling of the isoprenoid gene network in arabidopsis thaliana. *Genome Biol* 5(11):R92.

Yang, S.; Yuan, L.; Lai, Y.-C.; Shen, X.; Wonka, P.; and Ye, J. 2012. Feature grouping and selection over an undirected graph. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Zhang, Y., and Schneider, J. G. 2010. Learning multiple tasks with a sparse matrix-normal penalty. In *Advances in Neural Information Processing Systems*.

Zhang, Y., and Yeung, D.-Y. 2010. A convex formulation for learning task relationships in multi-task learning. In *The Conference on Uncertainty in Artificial Intelligence*.

Zhang, Y., and Yeung, D.-Y. 2014. A regularization approach to learning task relationships in multitask learning. *ACM Transactions on Knowledge Discovery from Data* 8(3):article 12.

Zhang, Y.; Yeung, D.-Y.; and Xu, Q. 2010. Probabilistic multi-task feature selection. In *Advances in Neural Information Processing Systems*.

Zhang, Y. 2013. Heterogeneous-neighborhood-based multi-task local learning algorithms. In *Advances in Neural Information Processing Systems 26*.

Zhong, W., and Kwok, J. 2012. Convex multitask learning with flexible task clusters. In *International Conference on Machine Learning*.

Zhou, J.; Yuan, L.; Liu, J.; and Ye, J. 2011. A multi-task learning formulation for predicting disease progression. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Zweig, A., and Weinshall, D. 2013. Hierarchical regularization cascade for joint learning. In *International Conference on Machine Learning*.