Robust Task Grouping with Representative Tasks for Clustered Multi-Task Learning

Yaqiang Yao
School of Computer Science and Technology, University of Science and Technology of China
Hefei, China
yaoyaq@mail.ustc.edu.cn

Jie Cao*
Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics
Nanjing, China
caojie690929@163.com

Huanhuan Chen*
School of Computer Science and Technology, University of Science and Technology of China
Hefei, China
hchen@ustc.edu.cn

ABSTRACT

Multi-task learning aims to learn multiple tasks jointly by sharing information among related tasks such that the generalization performance over different tasks could be improved. Although multi-task learning has been demonstrated to obtain performance gain in comparison with the single task learning, the main challenge that learning what to share with whom is still not fully resolved. In this paper, we propose a robust clustered multi-task learning approach that clusters tasks into several groups by learning the representative tasks. The main assumption behind our approach is that each task can be represented by a linear combination of some representative tasks that can characterize all tasks. The correlation between tasks can be indicated by the corresponding combination coefficient. By imposing a row-sparse constraint on the correlation matrix, our approach could select the representative tasks and encourage information sharing among the related tasks. In addition, the $l_{1,2}$-norm is applied to the representation loss to enhance the robustness of our approach. To solve the resulting bi-convex optimization problem, we design an efficient optimization method based on the alternating direction method of multipliers and accelerated proximal gradient method. Finally, experimental results on synthetic and real-world data sets validate the effectiveness of the proposed approach.

CCS CONCEPTS

• **Information systems → Data mining**; • **Computing methodologies → Multi-task learning**.

KEYWORDS

Clustered Multi-Task Learning; Representative Task Selection

ACM Reference Format:
Yaqiang Yao, Jie Cao, and Huanhuan Chen. 2019. Robust Task Grouping with Representative Tasks for Clustered Multi-Task Learning. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3292500.3330904

*Corresponding authors.

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330904

1 INTRODUCTION

Multi-task learning is a topic of interest in data mining, machine learning, natural language processing, and computer vision communities since many real-world applications in these areas involve learning multiple relevant tasks, such as entity recommendation [11], travel time estimation [18], image captioning [34], human action recognition [20, 38], etc. With the goal of improving generalization performance across different tasks, multi-task learning manages to learn multiple tasks simultaneously by transferring knowledge among them [27, 33]. To be specific, multi-task learning exploits the intrinsic relationships among multiple tasks to share information across related ones [2, 7–9, 13, 21, 36]. For example, in human activity recognition tasks, many activities are related and share basic motion action [38]. Due to the comparative advantages that are demonstrated empirically and theoretically over learning each task independently, multi-task learning has been greatly developed in the past decades.

As a subfield of transfer learning [27], the key challenge in multi-task learning is how to selectively transfer information among the related tasks and prevent information transferring across unrelated tasks meanwhile. The phenomenon that the information transfer among unrelated tasks would degenerate the generalization performance of multi-task learning, which is known as the negative transfer [27]. The traditional methods in the literature of multi-task learning to deal with this challenge can be divided into two main categories. The first kind of methods assumes that all tasks are related to each other, which can be achieved by two strategies: joint feature selection [14, 21, 30] and multi-task feature learning with low-rank structure [1, 2, 8, 16, 29]. On the other side, the second kind of methods assume that all tasks can be clustered into several groups and only the tasks within the same group are correlated and sharing information [13, 16, 22, 36, 38, 39].

Despite its better interpretability over other methods, the joint feature selection method has limited capacity since it fails to share common information that is not in the original feature space. On the other hand, the assumption that all tasks are related is not valid in some applications, which would lead to the negative transfer phenomenon. Another way to share information across all tasks is to constrain the task weights to lie in a low dimensional subspace, which is known as the low-rank methods. Due to the flexibility of sharing common information across tasks and parsimony of the number of efficient parameters, the low-rank methods are employed

to learn the structures of task grouping in recent years [4, 15]. Instead of assuming all tasks are related, the task grouping method or clustered multi-task learning manages to cluster tasks into several groups and share information among tasks in the same group to some extent [13, 16, 36, 39].

We focus on the clustered multi-task learning in this paper. Although the great progress in task grouping has been made during the past few years, there are still several major limitations in the existing clustered multi-task learning methods. First, the number of groups is usually not known in advance, which makes the clustered multi-task learning methods inflexible in practical applications. Second, a lot of task grouping methods cluster tasks into disjoint groups. Such hard-assignment may not be true and would lead to inefficient information sharing among tasks. Third, due to the l_2 distance between tasks behind the assumption that tasks within the same group are close to each other, the negatively correlated tasks will be clustered into different groups, which prevents information sharing between them. Finally, few task grouping methods consider the robustness against the outlier tasks which do not share information with all the other tasks.

1.1 Main Idea and Contributions

Motivated by the subset selection with structured sparsity [25, 40], we propose a new clustered multi-task learning approach by selecting representative tasks for task grouping. A subset of tasks termed representative tasks are selected to represent all tasks with a linear combination. The key insight of the proposed approach is that all tasks can be characterized by representative tasks. To be specific, we first represent each task by a linear combination of other tasks, where the linear combination coefficients indicate the relationship (correlation) between the corresponding tasks to some extent. Next, a row-sparse constraint ($l_{1,2}$-norm) is imposed on the relationship matrix to encourage information sharing among related tasks, which could select the most representative and informative tasks at the same time. Finally, to enhance the robustness of the proposed approach against outlier tasks, the $l_{1,2}$-norm is applied to the representation loss between each task and its linear combination of representative tasks, where l_1-norm and l_2-norm are imposed on tasks and features, respectively. A recent work relevant to ours identified representative tasks by minimizing the weighted l_2 distance between each task and its representative tasks [39]. However, it fails to fully discover the group structure since the negatively correlated tasks are put into different groups.

The idea of representing each task with a linear combination of the other tasks is not new [17, 22]. However, our approach differs from them in the following two points. First, we represent each task with a linear combination of the representative tasks by imposing the $l_{1,2}$-norm on the correlation matrix, which enables each task only to share information with the related tasks to the right extent. Second, we replace the squared l_2-norm on the representation loss with the accumulated l_1-norm, which enhances the robustness of our approach against outlier tasks. To the best our knowledge, Lee et al. [17] first proposed to represent each task model by a linear combination of other task models. This method preferred a sparse combination by imposing l_1-norm on the column of the task correlation matrix, which might fail to provide a complete cluster structure of tasks [22]. Moreover, the combination coefficients are restricted to be positive, which would prevent negatively correlated tasks from sharing information and being clustered into the same group. To cope with these problems, Liu and Pan [22] proposed to relax the positive restriction such that the method can capture both positive and negative correlations among tasks. In addition, the correlation matrix was restricted to be block-diagonal with a trace Lasso norm [10]. However, both the above methods define the diagonal elements of the correlation matrix to be zeros such that each task should be correlated with at least one of the other tasks, which is not always true in situations that exist irrelevant tasks. Moreover, the squared l_2 distance between the task model and its linear combination are sensitive to the outlier tasks.

In summary, our approach has the following advantages:

- Instead of pre-determining the number of task groups, our approach can learn it automatically from data.
- Each task can be clustered into different groups based on the representative tasks.
- The common information can be transferred among negatively correlated tasks.
- Our approach reduces the effect of the outlier tasks due to the accumulated l_1-norm on the representation loss.
- The objective function of our approach is an unconstrained bi-convex optimization problem.

The rest of this paper is organized as follows. We first introduce the proposed approach, including problem formulation, robust representative tasks selection and robust clustered multi-task learning with representative tasks in Section 2. Then we design an efficient optimization method to solve the objective function of our approach with the alternating direction method of multipliers and accelerated proximal gradient in Section 3. Next, extensive experimental studies on both synthetic and real-world data sets are presented in Section 4. Finally, we conclude this paper and describe some directions for future work in Section 5.

2 THE PROPOSED APPROACH

This section introduces the proposed approach, robust clustered multi-task learning with representative tasks (RCMTL). The key idea is to select a subset of tasks that can represent the whole tasks using the linear combination of them in multi-task learning, and then all tasks can be clustered into different groups based on these representative tasks. In the rest of this section, we first describe the problem formulation of clustered multi-task learning, followed by the robust representative tasks selection with structured sparsity. Next, we incorporate the robust task grouping with representative tasks selection into clustered multi-task learning.

2.1 Problem Formulation

Suppose we have a multi-task learning problem with m tasks, where each task $i \in \{1, \cdots, m\}$ is associated with a set of instances,

$$\mathcal{D}_i = \{(\mathbf{x}_1^i, y_1^i), \cdots, (\mathbf{x}_{n_i}^i, y_{n_i}^i)\} \subset \mathbb{R}^d \times \mathbb{R},$$

and a linear function $f_i : f_i(\mathbf{x}_j^i) = \mathbf{w}_i^\top \mathbf{x}_j^i$, where $\mathbf{w}_i$ is the weight of the i-th task, d is the dimensionality of data, and n_i is the number of instances in the i-th task. Denoting $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_m] \in \mathbb{R}^{d \times m}$

as the weight matrix to be estimated, the empirical risk is given by,

$$\mathcal{L}(\mathbf{W}) = \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(\mathbf{w}_i^\top \mathbf{x}_j^i, y_j^i\right).$$

where the loss function $l(\cdot,\cdot)$ is squared loss for regression problem and logistic loss for binary classification problem. To learn the m tasks simultaneously, we follow the well-established approach that searches for a weight matrix $\mathbf{W}$ such that the following regularized empirical risk is minimized,

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) + \Omega(\mathbf{W}),$$

where Ω is the regularization term that encodes the prior knowledge of the group structure of tasks.

2.2 Robust Representative Tasks Selection

Given m tasks, our goal is to select a subset of tasks, dubbed representative tasks, that share the most information with other tasks. Intuitively, the representative tasks can be considered as the most informative tasks that other tasks are highly related to and can be used to represent other tasks efficiently. Formally, assuming the model parameters of the representative tasks are $\mathbf{W}_{\mathcal{R}} = [\mathbf{w}_{n_1}, \cdots, \mathbf{w}_{n_r}] \in \mathbb{R}^{d\times r}$, the model parameter of each task $\mathbf{w}_i$ can be represented by the linear combination of the representative tasks, i.e. $\mathbf{w}_i \approx \mathbf{W}_{\mathcal{R}}\mathbf{c}_i$, where $\mathbf{c}_i = [c_{1i}, \cdots, c_{ri}]^\top$ is the combination coefficients, which indicates the correlation between corresponding tasks and the amount of information transferred from representative tasks to the i-th task. To select the representative tasks, we minimize the reconstruction error between the original model parameters of tasks and reconstructed model parameter by representative tasks as follows,

$$\begin{aligned} \min_{\mathbf{C},\mathbf{W}_{\mathcal{R}}} \quad & \sum_{i=1}^{m} \|\mathbf{w}_i - \mathbf{W}_{\mathcal{R}}\mathbf{c}_i\|_2^2, \\ \text{s.t.} \quad & \mathbf{W}_{\mathcal{R}} \subset \mathbf{W},\ |\mathbf{W}_{\mathcal{R}}| = r, \end{aligned} \tag{1}$$

where $\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_m] \in \mathbb{R}^{r\times m}$ is the coefficient matrix that describes the correlation between tasks.

However, the optimization problem in Eq. (1) is a combinatorial optimization problem that is difficult to solve. To mitigate this problem, we select the representative tasks from the whole tasks pool by replacing $\mathbf{W}_{\mathcal{R}}$ with $\mathbf{W}$ and imposing a row-sparsity constraint on the linear combination matrix. In the framework of multi-task learning, tasks are related to each other in many real-world problems. It is desirable to share useful information among the related tasks. Therefore, we encourage only the relevant tasks to share common information with each other by minimizing the number of non-zero rows in task correlation matrix $\mathbf{C}$. Following the previous work on group lasso [12], we formulate the representative task selection problem as an optimization problem on the task relationship matrix $\mathbf{C} \in \mathbb{R}^{m\times m}$ in the following,

$$\min_{\mathbf{C}} \quad \left\|(\mathbf{W} - \mathbf{W}\mathbf{C})^\top\right\|_F^2 + \lambda\|\mathbf{C}\|_{0,p}, \tag{2}$$

where $\|\mathbf{C}\|_{0,p} = \sum_{i=1}^{m} \mathrm{I}\left(\|\mathbf{C}_{i,:}\|_p\right)$ indicates the non-zero rows of relationship matrix $\mathbf{C}$. Note that $\mathrm{I}(\cdot)$ is the indicator function.

The squared loss of the representation in the above formulation is sensitive to outliers, which correspond to the outlier tasks that do not share information with all the other tasks in multi-task learning. To improve the robustness against outlier tasks, we replace the optimization problem in Eq. (2) with the following problem,

$$\min_{\mathbf{C}} \quad \left\|(\mathbf{W} - \mathbf{W}\mathbf{C})^\top\right\|_{1,2} + \lambda\|\mathbf{C}\|_{0,p}, \tag{3}$$

where $\|\mathbf{M}\|_{1,2} = \sum_i \|\mathbf{M}_{i,:}\|_2$ for a matrix $\mathbf{M}$. In other words, the l_1-norm and l_2-norm are imposed among the tasks and features respectively. Solving this problem can obtain the representative tasks indexed by the non-zero rows of $\mathbf{C}$, where the non-zero elements in the i-th row of $\mathbf{C}$ index tasks that select the i-th task as representative tasks. Consequently, the tasks that select the common representatives would share information.

2.3 Robust Clustered Multi-Task Learning with Representative Tasks

Based on the above preparation, we propose a new approach by incorporating the idea of robust representative tasks selection into clustered multi-task learning. To be specific, these tasks that select a common representative task are regarded as a group, and all tasks can be clustered into different groups based on their representative tasks. Formally, we formulate our approach as follows,

$$\begin{aligned} \min_{\mathbf{W},\mathbf{C}} \ & \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(\mathbf{w}_i^\top \mathbf{x}_j^i, y_j^i\right) + \lambda_1\|\mathbf{W}\|_F^2 \\ & + \lambda_2 \left\|(\mathbf{W} - \mathbf{W}\mathbf{C})^\top\right\|_{1,2} + \lambda_3\|\mathbf{C}\|_{0,p}, \end{aligned} \tag{4}$$

where the second regularization term controls the complexity of each task, the third regularization term is utilized to represent all tasks with their representative tasks, and the last regularization term is to control the number of representative tasks. Note that we absorb the bias parameter b_i into the weight $\mathbf{w}_i$ by defining an additional dummy feature $x_0^i = 1$ for instances in each task.

The optimization problem in Eq. (4) involves counting the number of nonzero rows of $\mathbf{C}$, which is non-convex and NP-hard in general. Following the recent theoretical progress [32] on grouped variables, we relax the l_0-norm to the convex proxy l_1-norm. On the other hand, the typical value of $p \in \{2, \infty\}$, we set $p = 2$ such that each task can be represented by representative tasks of different weights. Therefore, the final optimization problem becomes the following form,

$$\begin{aligned} \min_{\mathbf{W},\mathbf{C}} \ & \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(\mathbf{w}_i^\top \mathbf{x}_j^i, y_j^i\right) + \lambda_1\|\mathbf{W}\|_F^2 \\ & + \lambda_2 \left\|(\mathbf{W} - \mathbf{W}\mathbf{C})^\top\right\|_{1,2} + \lambda_3\|\mathbf{C}\|_{1,2}. \end{aligned} \tag{5}$$

Compared with the previous work on clustered multi-task learning, our approach learn the number of clusters from data automatically. Each task can be clustered into different groups based on the representative tasks. Furthermore, by representing each task with a linear combination of representative tasks, the negatively correlated tasks can be put into the same group and share information.

3 OPTIMIZATION METHOD

The optimization problem in Eq. (5) is not convex in terms of all variables together, but is convex in terms of them respectively. We customize an efficient alternating optimization to obtain the partial

minimum of the objective function. To be specific, the $l_{1,2}$ regularization term is not trivial for optimization due to its non-smoothness and mixed-norm structure, we exploit the well-developed alternating direction method of multipliers (ADMM) [6] and accelerated proximal gradient (APG) [28] to solve it. In the rest of this section, we first give an introduction of accelerated proximal gradient since it is used in both steps of the alternating optimization, followed by two procedures of the alternating optimization.

3.1 Accelerated Proximal Gradient

Due to the optimal convergence rate for the class of first-order methods, accelerated gradient methods [23, 24] has been extensively utilized to solve multi-task learning problems [7, 9, 35, 36] of the following form

$$\min_{\mathbf{W}} \mathcal{F}(\mathbf{W}) + \mathcal{R}(\mathbf{W}), \tag{6}$$

where $\mathcal{F}(\mathbf{W})$ is convex and smooth, and $\mathcal{R}(\mathbf{W})$ is convex but non-smooth. Note that the objective function in Eq. (5) is a composite function of a differential term $\mathcal{F}(\mathbf{W})$ and a non-differential term $\mathcal{R}(\mathbf{W})$. Denote

$$\mathcal{T}_{\mathbf{V},\gamma}(\mathbf{W}) = \mathcal{F}(\mathbf{V}) + \left\langle \frac{\partial \mathcal{F}(\mathbf{V})}{\partial \mathbf{V}}, \mathbf{W} - \mathbf{V} \right\rangle_F + \frac{\gamma}{2}\|\mathbf{W} - \mathbf{V}\|_F^2,$$

where the first two terms are the first order Taylor expansion of $\mathcal{F}(\mathbf{W})$ at $\mathbf{V}$, and the last term is the regularization.

In the traditional gradient descent method, the solution at the k-th iteration is obtained as follows,

$$\mathbf{W}^k = \operatorname*{argmin}_{\mathbf{W}} \mathcal{T}_{\mathbf{W}^{k-1},\gamma_k}(\mathbf{W}) + \mathcal{R}(\mathbf{W}) \quad (k \geq 1),$$

where γ_k is a proper step size. Here, we use the well-known fast iterative shrinkage-thresholding algorithm (FISTA) [5] to solve the optimization problem. To be specific, at the k-th iteration, FISTA generates the solution by computing the proximal operator [3, 21, 31] in the following,

$$\mathbf{W}^k = \operatorname*{argmin}_{\mathbf{W}} \mathcal{T}_{\mathbf{V}^k,\gamma_k}(\mathbf{W}) + \mathcal{R}(\mathbf{W}) \quad (k \geq 1), \tag{7}$$

where $\mathbf{V}^1 = \mathbf{W}^0$, $\mathbf{V}^{k+1} = \mathbf{W}^k + \alpha^k(\mathbf{W}^k - \mathbf{W}^{k-1})$ for $k \geq 1$, and γ_k is scalar obtained by the linear search. In particular, $\gamma_0 = 1$, and the value of γ_k for $k \geq 1$ is set as $\gamma_k = 2^{j_k}\gamma_{k-1}$ by finding the smallest non-negative integer j_k such that

$$\mathcal{F}(\mathbf{W}^k) \leq \mathcal{T}_{\mathbf{V}^k,\gamma_k}(\mathbf{W}^k). \tag{8}$$

Note that $\mathbf{V}^{k+1}$ is a linear combination of $\mathbf{W}^k$ and $\mathbf{W}^{k-1}$, and the coefficient α_k is crucial to the convergence behavior of the algorithm. Following the strategy in [5], we set $\alpha_k = (t_{k-1} - 1)/t_k$, where $t_0 = 1$ and $t_k = \left(1 + \sqrt{4t_{k-1}^2 + 1}\right)/2$ for $k \geq 1$.

3.1.1 Proximal Operator. The proximal operator with respect to $\mathbf{W}$ can be cast into the following optimization problem,

$$\mathbf{W}^k = \operatorname*{argmin}_{\mathbf{W}} \frac{1}{2}\left\|\mathbf{W} - \widehat{\mathbf{W}}^k\right\|_F^2 + \mathcal{R}(\mathbf{W}), \tag{9}$$

where

$$\widehat{\mathbf{W}}^k = \mathbf{V}^k - \nabla_{\mathbf{V}}\mathcal{F}(\mathbf{V}^k)/\gamma_k$$

and $\nabla_{\mathbf{V}}\mathcal{F}(\mathbf{V}^k)$ is the derivative of $\mathcal{F}(\mathbf{V})$ with respect to $\mathbf{V}$ at $\mathbf{V}^k$. Taking the derivative of Eq. (9) with respect to $\mathbf{W}$ and setting the derivative to zero, we can obtain the solution of the above optimization problem. The main algorithm for solving the problem (6) is summarized in Algorithm 1.

Algorithm 1 Accelerated Proximal Gradient Method

Input: Data and parameters for multi-task learning problem, γ_0;
Output: $\mathbf{W}$;

Initialization: $\mathbf{W}^0 = \mathbf{W}^{-1}$, $t_{-1} = 0$, $t_0 = 1$, and $k = 1$
repeat
  $\alpha_k = \frac{t_{k-2}-1}{t_{k-1}}$;
  $\mathbf{V}^k = \mathbf{W}^{k-1} + \alpha_k(\mathbf{W}^{k-1} - \mathbf{W}^{k-2})$;
  Compute $\nabla_{\mathbf{V}}\mathcal{F}(\mathbf{V}^k)$
  for $j = 0$ to $\cdots$ **do**
    Set $\gamma = 2^j\gamma_{k-1}$;
    Compute $\mathbf{W}^k$ by solving the Eq. (9);
    if $\mathcal{F}(\mathbf{W}^k) \leq \mathcal{T}_{\mathbf{V}^k,\gamma}(\mathbf{W}^k)$ **then**
      $\gamma_k = \gamma$;
      break;
    end if
  end for
  $t_k = \left(1 + \sqrt{1 + 4t_{k-1}^2}\right)/2$;
  $k = k + 1$;
until convergence

3.2 Optimizing Task Weights Matrix W

For a fixed task relationship matrix $\mathbf{C}$, the objective function Eq. (5) with respect to task weights matrix $\mathbf{W}$ is as follows,

$$\min_{\mathbf{W}} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(\mathbf{w}_i^\top \mathbf{x}_j^i, y_j^i\right) + \lambda_1\|\mathbf{W}\|_F^2 + \lambda_2\left\|\mathbf{D}^\top\mathbf{W}^\top\right\|_{1,2}, \tag{10}$$

where $\mathbf{D} = \mathbf{I} - \mathbf{C}$. This procedure is learning multiple tasks simultaneously with the given group structure and can be solved by the alternating direction method of multipliers [6]. First, we introduce an auxiliary matrix $\mathbf{E} = \mathbf{W}\mathbf{D} \in \mathbb{R}^{d\times m}$ and obtain the following optimization program,

$$\begin{aligned} \min_{\mathbf{W},\mathbf{E}} & \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(\mathbf{w}_i^\top \mathbf{x}_j^i, y_j^i\right) + \lambda_1\|\mathbf{W}\|_F^2 + \lambda_2\left\|\mathbf{E}^\top\right\|_{1,2} \\ & + \frac{\rho}{2}\|\mathbf{W}\mathbf{D} - \mathbf{E}\|_F^2, \\ \text{s.t. } & \mathbf{W}\mathbf{D} = \mathbf{E}, \end{aligned} \tag{11}$$

where ρ is a penalty parameter. Note that Eq. (10) and Eq. (11) are equivalent since the last term in the objective function of Eq. (11) vanishes for any feasible solution when $\mathbf{W}\mathbf{D} = \mathbf{E}$.

Then we adopt the Lagrange multiplier matrix $\mathbf{\Lambda} \in \mathbb{R}^{d\times m}$ to augment the equality constraint of Eq. (11) to the objective function, which can be written as the Lagrangian function as follows,

$$\begin{aligned} \min_{\mathbf{W},\mathbf{E},\mathbf{\Lambda}} & \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(\mathbf{w}_i^\top \mathbf{x}_j^i, y_j^i\right) + \lambda_1\|\mathbf{W}\|_F^2 + \lambda_2\left\|\mathbf{E}^\top\right\|_{1,2} \\ & + \frac{\rho}{2}\|\mathbf{W}\mathbf{D} - \mathbf{E}\|_F^2 + \mathrm{Tr}\left(\mathbf{\Lambda}^\top(\mathbf{W}\mathbf{D} - \mathbf{E})\right). \end{aligned} \tag{12}$$

Algorithm 2 Robust Clustered Multi-Task Learning

Input: Data set $\{\mathbf{X}^i, \mathbf{y}^i\}_{i=1}^m$; Regularization parameters: λ_1, λ_2, λ_3; Parameters for ADMM: ρ;
Output: Task weights matrix $\mathbf{W}$ and task relationship matrix $\mathbf{C}$;

Initialization: $\mathbf{W}^0$ and $\mathbf{C}^0$;
repeat
  repeat
    Update the auxiliary matrix $\mathbf{E}$;
    Update the task weights matrix $\mathbf{W}$;
    Update the Lagrangian multiplier $\mathbf{\Lambda}$;
  until convergence.
  repeat
    Update the auxiliary matrix $\mathbf{F}$;
    Update the task relationship matrix $\mathbf{C}$;
    Update the Lagrangian multiplier $\mathbf{\Gamma}$;
  until convergence.
until the objective function of Eq. (5) converges.

Based on the above formulation, the auxiliary matrix $\mathbf{E}$, the task weights matrix $\mathbf{W}$, and the Lagrangian multiplier $\mathbf{\Lambda}$ can be updated by using the APG algorithm and ADMM algorithm.

3.3 Optimizing Task Relationship Matrix C

Given the task weights matrix $\mathbf{W}$, the objective function with respect to the task relationship matrix is as follows,

$$\min_{\mathbf{C}} \ \lambda_2 \left\|(\mathbf{W} - \mathbf{WC})^\top\right\|_{1,2} + \lambda_3 \|\mathbf{C}\|_{1,2}, \tag{13}$$

which can be considered as selecting representative tasks for all tasks. Similarly, ADMM is utilized to solve this optimization program. First, we introduce an auxiliary matrix $\mathbf{F} = \mathbf{W} - \mathbf{WC} \in \mathbb{R}^{d\times m}$ and obtain the following optimization problem,

$$\begin{aligned} &\min_{\mathbf{C},\mathbf{F}} \lambda_2 \left\|\mathbf{F}^\top\right\|_{1,2} + \lambda_3 \|\mathbf{C}\|_{1,2} + \frac{\rho}{2}\|\mathbf{F} - \mathbf{W} + \mathbf{WC}\|_F^2, \\ &\text{s.t. } \mathbf{F} = \mathbf{W} - \mathbf{WC}. \end{aligned} \tag{14}$$

where ρ is a penalty parameter. To deal with the equality constraint, the most convenient way is adopting the Lagrange multiplier matrix $\mathbf{\Gamma} \in \mathbf{R}^{d\times m}$ to augment the equality constraint to the objective function, which can be written as the Lagrangian function

$$\begin{aligned} \min_{\mathbf{C},\mathbf{F}} \lambda_2 \left\|\mathbf{F}^\top\right\|_{1,2} + \lambda_3 \|\mathbf{C}\|_{1,2} + \frac{\rho}{2}\|\mathbf{F} - \mathbf{W} + \mathbf{WC}\|_F^2 \\ + \mathrm{Tr}\left(\mathbf{\Gamma}^\top(\mathbf{F} - \mathbf{W} + \mathbf{WC})\right). \end{aligned} \tag{15}$$

Then, we can update the auxiliary matrix $\mathbf{F}$, the task relationship matrix $\mathbf{C}$, and the Lagrangian multiplier $\mathbf{\Gamma}$ in the same way as in Section 3.2. The overall procedures of the proposed approach are summarized in Algorithm 2.

4 EXPERIMENTAL STUDIES

In this section, we study the experimental results of the proposed approach RCMTL on both synthetic and real-world data sets. We first introduce the experimental setting, including comparative methods and performance evaluation criteria, and then analyze the experimental results. Finally, the complexity and convergence analysis of the optimization method is presented.

Table 1: Performance of different methods w.r.t. rMSE on synthetic data sets in the form of 'mean ± std'.

Method	S1	S2	S3
STL	9.1601 ± 0.1072	7.6511 ± 0.1518	10.1069 ± 0.1864
L12	6.4231 ± 0.0838	6.3359 ± 0.2223	9.1835 ± 0.1729
RMTFL	6.3441 ± 0.0830	6.2954 ± 0.2168	8.8231 ± 0.1800
Dirty	4.9838 ± 0.1407	4.7066 ± 0.1757	6.9836 ± 0.2730
CMTL	6.1870 ± 0.3402	5.8370 ± 0.2216	8.8711 ± 0.1739
RCMTL	**3.4015 ± 0.1354**	**3.8406 ± 0.1194**	**6.1476 ± 0.2486**

4.1 Experimental Setting

To demonstrate the competitiveness of the proposed approach, we compare the proposed approach with the following single task learning method and the recently proposed strategies for multi-task learning methods:

- **STL**: The single task learning method in which the tasks are learned separately with l_2-norm regularization.
- **L12**: This method assumes all tasks share a common set of features to capture the task relatedness from multiple related tasks [2].
- **RMTFL**: It assumes that the task model can be decomposed into a shared feature part that captures task relatedness and a group-sparse part that detects outliers [9].
- **Dirty**: To deal with dirty data that do not fall into a single structure, Dirty decomposes the task model into a group sparse component and a sparse component [14].
- **CMTL**: CMTL assumes the tasks exhibit a group structure where the tasks from the same group are closer to each other than those from a different group [36].

The implementation of all these competitive methods is released in [37]. For all the baseline and multi-task learning methods, the hyper-parameters are selected by grid search on the performance of the validation set. In particular, the search ranges of the regularization parameters are $\{2^{-10}, \cdots, 2^5\}$. We employ the root mean squared error (rMSE) to evaluate the performance of the proposed approach for the regression problem. The final measurement for the multi-task problem is the mean of the measurement on all tasks. Note that better regression performance is indicated by the smaller value of rMSE. As for the classification problem, the mean average precision (Mean AP) is employed to measure the performance, and a larger value of Mean AP indicates better classification performance. For each data set, the experiments on different methods are repeated for 10 times by splitting data set randomly, and the mean and standard deviation of the results are reported.

4.2 Synthetic Data Sets

To validate the effectiveness of the proposed approach in terms of task grouping, negative correlation, and robustness against outlier tasks, we first evaluate our approach on three synthetic data sets. To be specific, the task is a linear regression problem. The input data are generated from $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with feature dimensionality $d = 100$ and the output of the i-th task is obtained by $y^i = \mathbf{w}_i^\top \mathbf{x} + \mathcal{N}(0, 1.5)$. Each data set consists of 4 clusters and

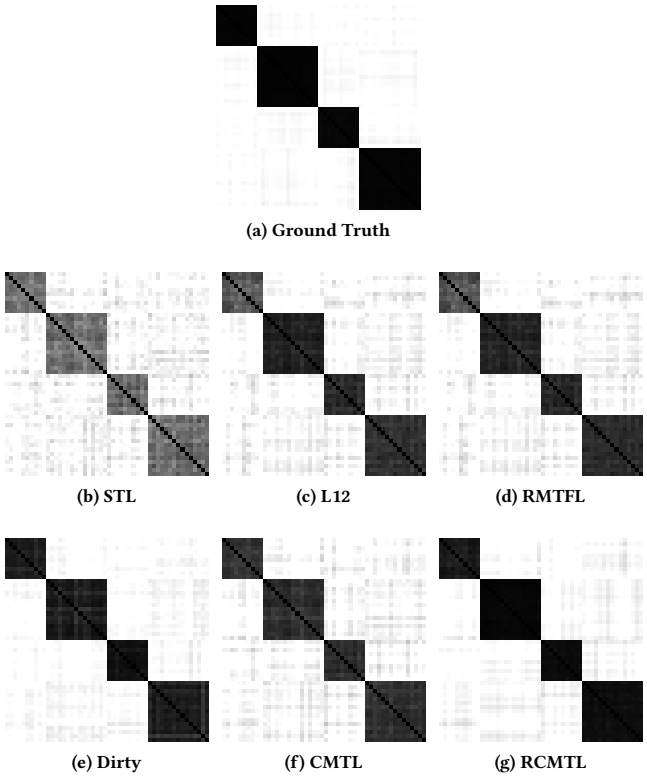


Figure 1: Illustration of the correlation matrix obtained by different methods on S1 data set.

each cluster contains 10, 15, 10, 15 tasks, respectively. All 100 dimensions are divided into 4 disjoint groups and each group contains 20, 30, 20, 30 dimensions and is assigned to only one cluster. For tasks from a particular cluster, the corresponding dimensions are non-zero and all other dimensions are zero, which makes different clusters orthogonal to each other. For each task, we generate 60 samples as training data, 40 samples as the validation set to tune the regularization parameters, and 100 samples for testing.

S1: for the i-th task in the c-th cluster, the value of each dimension is the sum of its cluster center $\overline{\mathbf{w}}_c$ and a task-specific component $\mathbf{w}_i$, where $\overline{\mathbf{w}}_c \sim \mathcal{N}(\mathbf{0}, 3\mathbf{I})$ and $\mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, 0.2\mathbf{I})$.

S2: we first randomly generate 2, 3, 2, 3 correlated tasks from $\mathcal{N}(\mathbf{0}, 3\mathbf{I})$ for 4 clusters, respectively, and then generate the rest tasks in each cluster by a linear combination of their correlated tasks.

S3: this data set is the same as S2 except five tasks are replaced by five outlier tasks generated from $0.5 + \mathcal{N}(0, 3)$, and the dimensions in these outlier tasks are non-zero.

The above three synthetic data sets are used to evaluate the performance of the proposed approach in terms of task grouping, negative correlation, and robustness against outlier tasks, respectively. Table 1 presents the results of different methods on three synthetic data sets. We observe that all multi-task learning methods improve the performance of single task learning by learning tasks simultaneously. To be specific, RCMTL achieves the best performance on all data sets, which validates the effectiveness of our approach. Since the variance of the noise is close to the variance of cluster weights, the obtained data do not fall into a single structure, which blurs the cluster structure of tasks to some extent. As a result, Dirty obtains better performance than CMTL. After corrupting the model parameters with outlier tasks on S3, RMTFL begins to outperform CMTL.

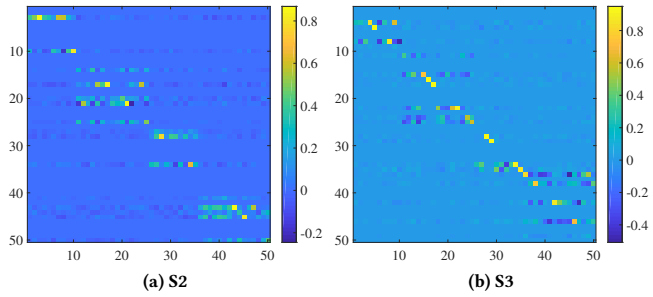


Figure 2: Illustration of the representative tasks obtained by our approach on data sets S2 and S3.

In addition, Fig. 1 illustrates the correlation matrix obtained by different methods on S1. Obviously, all multi-tasking learning methods obtain more accurate cluster structures than STL. In particular, RCMTL learns the underlying cluster structure of tasks most accurately. Although Dirty obtains a result of high quality as well, some noise is introduced to the correlation of tasks within the same cluster, which may be attributed to the group sparse component. Furthermore, we illustrate the representative tasks learned by our approach on S2 and S3 in Fig. 2, which shows that the number of obtained representative tasks is proportional to the number of underlying representative tasks in each cluster on S2. Note that the representative tasks are not deterministic due to the property of linear combination. On S3, each outlier task selects itself as the only representative task.

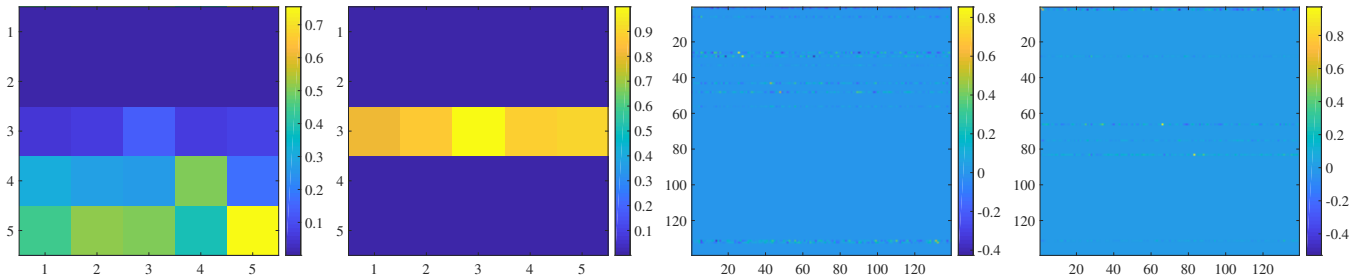
4.3 Real-World Data Sets

Then, we evaluate the performance of our approach on following four real-world data sets, in which the first two data sets Isolet and School are for regression problem and the last two data sets MHC-I Binding and CIFAR-10 are for classification problem.

4.3.1 Regression Problem. **Isolated Letter Speech Recognition**: The Isolet data set is collected from 150 subjects who spoke the name of each letter in the English alphabet twice. Hence there are 52 samples from each subject. These speakers are divided into 5 groups such that each group consists of 30 similar speakers. The problem of recognizing the isolated letter speech can be regarded as a multi-task regression problem, where each group corresponds to a task. Since 3 samples are missing, we have 1560, 1560, 1560, 1558 and 1559 instances in five tasks, respectively. Following the setup in [19], the target values are set as the corresponding letter labels. In the experiments, we first reduce the feature dimension of data to 100 with PCA, and then randomly select 15%, 20% and 30% of the instances from each task as the training data, 20% of the instances as the validation data for the hyper-parameters selection, and the rest as the testing data.

Table 2: Performance of different methods w.r.t. rMSE on Isolet and School data in the form of 'mean ± std'.

Data set	Ratio	STL	L12	RMTFL	Dirty	CMTL	RCMTL
Isolet	15%	5.7979 ± 0.0616	5.3797 ± 0.0549	5.6129 ± 0.0962	5.7896 ± 0.0585	5.3464 ± 0.0400	**5.2721 ± 0.0647**
	20%	5.5206 ± 0.0699	5.2664 ± 0.0419	5.3903 ± 0.0557	5.5176 ± 0.0632	5.2038 ± 0.0424	**5.0930 ± 0.0542**
	25%	5.3583 ± 0.0403	5.1696 ± 0.0598	5.2535 ± 0.0577	5.3586 ± 0.0414	5.1252 ± 0.0397	**5.0520 ± 0.0710**
School	10%	12.1026 ± 0.0815	11.6623 ± 0.0779	11.8670 ± 0.0646	12.7683 ± 0.2005	11.7431 ± 0.1152	**11.6189 ± 0.2440**
	20%	11.1131 ± 0.0678	10.8762 ± 0.0907	10.9902 ± 0.0802	11.3137 ± 0.0811	11.0281 ± 0.0748	**10.8217 ± 0.1612**
	30%	10.7599 ± 0.1209	**10.5897 ± 0.1054**	10.6698 ± 0.0030	10.8354 ± 0.1298	10.7182 ± 0.1099	10.6228 ± 0.1577



(a) Isolet with 15% Data (b) Isolet with 20% Data (c) School with 10% Data (d) School with 20% Data

Figure 3: Illustration of the representative tasks obtained by the proposed approach on Isolet and School data sets.

Examination Score Prediction: The School data set [2] records the examination scores of 15362 students from 139 secondary schools during the three years from 1985 to 1987 in London. Following the processing described in [2], each student is represented by 27 binary features including school-specific and student-specific attributes. The corresponding examination score is an integer. The problem of predicting the examination score of the students can be viewed as a multi-task regression problem, where each school corresponds to a task that consists of a different number of students as the samples. In our experiments, 10%, 20% and 25% of the instances are randomly selected from each task as the training data, 20% of the instances are used as the validation data for the hyper-parameters selection, and the rest instances are served as the testing data.

Table 2 presents the results on the above two data sets. It shows that all multi-task learning methods outperform the method of single task learning except Dirty, which is inferior to STL on School. The possible reason is that the School data is a single structure and decomposing the task model into a group sparse part and a sparse part would destroy the structure. RCMTL achieves the best performance on School data under all setting and the comparable performance with L12 under the third setting due to the assumption that all tasks might share a common set of features is valid in this data, which demonstrate the advantages of our approach. Moreover, the representative tasks obtained by the proposed approach is illustrated in Fig. 3, which shows that 3 and 1 representative task (indicated by the non-zero rows) are selected on Isolet data under the first two settings, respectively. For the School data, 10 and 7 representative tasks are selected. Please zoom in the figures for better visual effect.

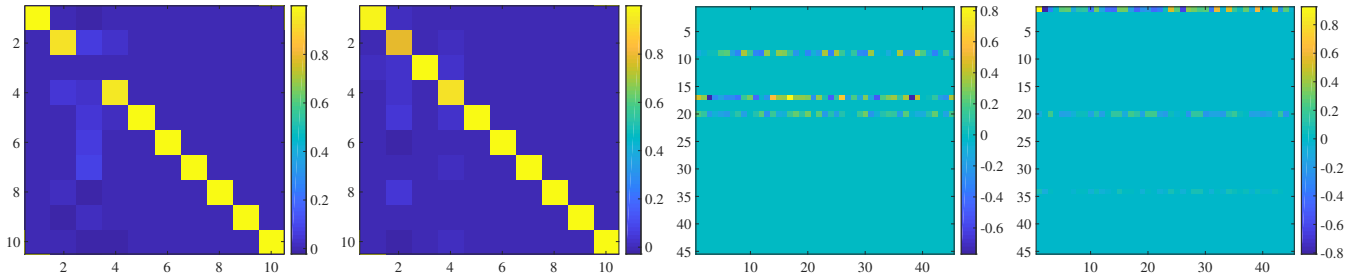
4.3.2 Classification Problem. **MHC-I Binding Data Set**: This data set contains binding affinities of various peptides with different MHC-I molecules [13]. The binary classification problem of predicting whether a peptide binds a molecule can be viewed as a multi-task classification problem, where each molecule corresponds to a task that consists of the different number of peptides as instances. Following the protocol utilized in [13], we perform experiments on the 10 molecules that have less than 200 instances described by 180 features, and we obtain 1200 samples in total. In the experiments, 20% and 40% of the instances are randomly selected from each task as the training data, 20% of the instances are used as the validation data for the hyper-parameters selection, and the remaining data are served as the testing data.

The CIFAR-10 Data Set: The CIFAR-10 data set consists of 60000 color images of 32×32 from 10 classes, each of which has 6000 images. This data set is partitioned into five training batches of size 10000 and one testing batch of the same size. In the testing batch, each class contains exactly 1000 images. Following [22], we generalize this multi-class classification problem to several binary classification tasks with one-versus-one strategy. Hence, we obtain $\binom{10}{2} = 45$ tasks in total, each of which is a binary classification task. We first convert the color images into the grayscale using Luma coding, and then reduce the dimension of data to 76 with PCA. Only the testing batch is used as the data set in our experiments. In each task, we randomly select 30% and 40% of the instances as the training data, 20% of the instances as the validation data for the hyper-parameters selection, and the rest as the testing data.

Table 3 reports the performance on both data sets over 10 trials under two settings. It shows that our approach obtains the best results on both data sets under both settings. In particular, RCMTL

Table 3: Performance of different methods w.r.t. Mean AP on MHC-I and CIFAR-10 data sets in the form of 'mean ± std'.

Data set	Ratio	STL	L12	RMTFL	Dirty	CMTL	RCMTL
MHC-I	20%	0.6754 ± 0.0195	0.6804 ± 0.0270	0.6787 ± 0.0258	0.6790 ± 0.0208	0.6776 ± 0.0254	**0.6872 ± 0.0262**
	40%	0.7032 ± 0.0104	0.7221 ± 0.0098	0.7191 ± 0.0183	0.7186 ± 0.0156	0.7210 ± 0.0202	**0.7336 ± 0.0163**
CIFAR-10	30%	0.7006 ± 0.0018	0.7120 ± 0.0018	0.7040 ± 0.0024	0.7029 ± 0.0018	0.7088 ± 0.0018	**0.7132 ± 0.0018**
	40%	0.7019 ± 0.0021	0.7156 ± 0.0021	0.7062 ± 0.0016	0.7052 ± 0.0022	0.7100 ± 0.0020	**0.7160 ± 0.0022**



(a) MHC-I with 20% Data (b) MHC-I with 40% Data (c) CIFAR-10 with 30% Data (d) CIFAR-10 with 40% Data

Figure 4: Illustration of the representative tasks obtained by the proposed approach on MHC-I and CIFAR-10 data sets.

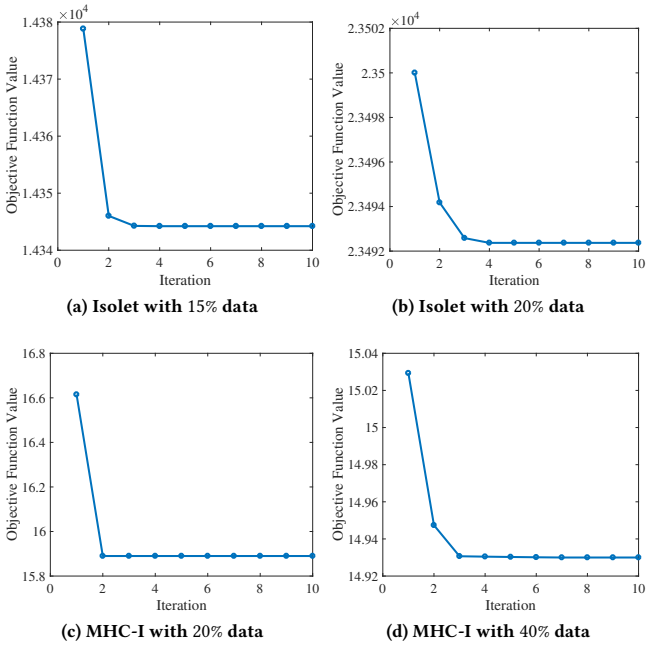
obtains comparable results with L12 and outperform the rest with a relatively large gap. Unsurprisingly, the single task learning method is worse than all methods of multi-task learning. Furthermore, Fig. 4 illustrates the obtained representative tasks and the task relationship matrix. We can observe that even though the number of representative tasks is almost equivalent to the number of total tasks, the performance on MHC-I data is still improved by our approach. For CIFAR-10 data, three different representative tasks (indicated by the non-zero rows) are selected under both settings.

4.4 Complexity and Convergence Analysis

For simplicity, we assume that each task has n training instances. The computational complexity of the proposed approach is composed of the following two parts:

- In the step of optimizing the task weights matrix $\mathbf{W}$, the computational cost is dominated by the computation of the gradient of $\mathbf{W}$. In particular, the total cost of this step is $O(dm(n+m))$.
- In the step of optimizing the task relationship matrix $\mathbf{C}$, the most time-consuming operation is updating $\mathbf{C}$ with accelerated proximal gradient, which can be done with $O(dm^2)$ computational time.

Therefore, the overall computational complexity of our approach is $O(dmn + dm^2)$. On the other hand, the objective function in Eq. (5) is not convex in terms of both variables together, the Algorithm 2 converges to the local minimum only if both the alternating steps converge to their global minimum. However, there is no theory currently to guarantee the convergence of ADMM for the optimization problem in Eq. (5) [6]. As an alternative, we demonstrate the convergence property of our approach empirically in Fig. 5, from which we can observe that the objective function of RCMTL can converge to the optimal value within 10 iterations in most cases.



(a) Isolet with 15% data (b) Isolet with 20% data (c) MHC-I with 20% data (d) MHC-I with 40% data

Figure 5: Illustration of the convergence of our approach on two real-world data sets.

5 CONCLUSION

Based on the assumption that each task can be represented by a linear combination of some representative tasks which can characterize all tasks, we propose a robust task grouping by selecting the representative tasks for clustered multi-task learning in this paper.

To be specific, our approach uncovers the underlying structure of tasks by selecting the representative tasks which share the most information with other tasks. Based on the shared representative tasks, related tasks are grouped into clusters that could be overlapped such that information can be shared among clusters to some extent. Moreover, the accumulated l_1-norm is utilized to measure the representation loss between each task and its reconstruction by the representative tasks, which could reduce the effect of outlier tasks. Experiments on both synthetic and real-world data sets demonstrate the effectiveness of our approach in comparison with the existing methods.

There are two interesting directions to improve the proposed approach in future work. First, in consideration of the flexible property of low-rank structure in multi-task feature learning, we aim to perform our task grouping method based on the low-rank assumption. Second, combining the group structure with joint feature selection would further improve the interpretability and generalization performance of multi-task learning.

ACKNOWLEDGMENTS

This research is supported in part by the National Key Research and Development Program of China under Grant No. 2016YFB1000905, the National Natural Science Foundation of China under Grant No. 91646204, 91846111 and 91746209.

REFERENCES

[1] Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* 6, Nov (2005), 1817–1853.
[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2008. Convex multi-task feature learning. *Machine Learning* 73, 3 (2008), 243–272.
[3] Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. 2011. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning* 5 (2011), 19–53.
[4] Aviad Barzilai and Koby Crammer. 2015. Convex multi-task learning by clustering. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*. 65–73.
[5] Amir Beck and Marc Teboulle. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences* 2, 1 (2009), 183–202.
[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1 (2011), 1–122.
[7] Jianhui Chen, Lei Tang, Jun Liu, and Jieping Ye. 2013. A convex formulation for learning a shared predictive structure from multiple tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 5 (2013), 1025–1038.
[8] Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 109–117.
[9] Pinghua Gong, Jieping Ye, and Changshui Zhang. 2012. Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 895–903.
[10] Édouard Grave, Guillaume Obozinski, and Francis Bach. 2011. Trace Lasso: A Trace Norm Regularization for Correlated Designs. In *Advances in Neural Information Processing Systems*. 2187–2195.
[11] Jizhou Huang, Wei Zhang, Yaming Sun, Haifeng Wang, and Ting Liu. 2018. Improving Entity Recommendation with Search Log and Multi-Task Learning.. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 4107–4114.
[12] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. 2009. Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 433–440.
[13] 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.
[14] 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.
[15] Jun-Yong Jeong and Chi-Hyuck Jun. 2018. Variable Selection and Task Grouping for Multi-Task Learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1589–1598.
[16] Abhishek Kumar and Hal Daumé. 2012. Learning Task Grouping and Overlap in Multi-task Learning. In *Proceedings of the 29th Annual International Conference on Machine Learning*. 1383–1390.
[17] Giwoong Lee, Eunho Yang, and Sung Hwang. 2016. Asymmetric multi-task learning based on task relatedness and loss. In *Proceedings of the 33rd International Conference on Machine Learning*. 230–238.
[18] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. 2018. Multi-task representation learning for travel time estimation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1695–1704.
[19] Ya Li, Xinmei Tian, Tongliang Liu, and Dacheng Tao. 2015. Multi-task Model and Feature Joint Learning. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 3643–3649.
[20] Anan Liu, Yuting Su, Weizhi Nie, and Mohan S Kankanhalli. 2017. Hierarchical Clustering Multi-Task Learning for Joint Human Action Grouping and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 1 (2017), 102–114.
[21] Jun Liu, Shuiwang Ji, and Jieping Ye. 2009. Multi-task feature learning via efficient l 2, 1-norm minimization. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 339–348.
[22] Sulin Liu and Sinno Jialin Pan. 2017. Adaptive group sparse multi-task learning via trace lasso. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2358–2364.
[23] Yurii Nesterov. 2013. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media.
[24] Yurii Nesterov et al. 2007. Gradient methods for minimizing composite objective function.
[25] Feiping Nie, Hua Wang, Heng Huang, and Chris HQ Ding. 2013. Early Active Learning via Robust Representation and Structured Sparsity.. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 1572–1578.
[26] Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization* (second ed.). Springer, New York, NY, USA.
[27] Sinno Jialin Pan, Qiang Yang, et al. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359.
[28] Neal Parikh, Stephen Boyd, et al. 2014. Proximal algorithms. *Foundations and Trends® in Optimization* 1, 3 (2014), 127–239.
[29] Piyush Rai and Hal Daume III. 2010. Infinite predictor subspace models for multitask learning. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. 613–620.
[30] Xin Wang, Jinbo Bi, Shipeng Yu, and Jiangwen Sun. 2014. On multiplicative multitask feature learning. In *Advances in Neural Information Processing Systems*. 2411–2419.
[31] Lei Yuan, Jun Liu, and Jieping Ye. 2011. Efficient methods for overlapping group lasso. In *Advances in Neural Information Processing Systems*. 352–360.
[32] Ming Yuan and Yi Lin. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68, 1 (2006), 49–67.
[33] Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* (2017).
[34] Wei Zhao, Benyou Wang, Jianbo Ye, Min Yang, Zhou Zhao, Ruotian Luo, and Yu Qiao. 2018. A Multi-task Learning Approach for Image Captioning.. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 1205–1211.
[35] Wenliang Zhong and James Kwok. 2012. Convex multitask learning with flexible task clusters. In *Proceedings of the 29th annual international conference on machine learning*. ACM, 49–56.
[36] Jiayu Zhou, Jianhui Chen, and Jieping Ye. 2011. Clustered multi-task learning via alternating structure optimization. In *Advances in Neural Information Processing Systems*. 702–710.
[37] Jiayu Zhou, Jianhui Chen, and Jieping Ye. 2011. MALSAR: Multi-task learning via structural regularization. *Arizona State University* (2011).
[38] Qiang Zhou, Gang Wang, Kui Jia, and Qi Zhao. 2013. Learning to share latent tasks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*. 2264–2271.
[39] Qiang Zhou and Qi Zhao. 2016. Flexible Clustered Multi-Task Learning by Learning Representative Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 2 (2016), 266–278.
[40] Feiyun Zhu, Bin Fan, Xinliang Zhu, Ying Wang, Shiming Xiang, and Chunhong Pan. 2015. 10,000+ times accelerated robust subset selection (ARSS). In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*. 3217–3224.

A REPRODUCIBILITY

For the reproducibility of our experiments, we first present the initialization of parameters $\mathbf{W}$ and $\mathbf{C}$ in the objective function of the proposed approach and then detail the procedures of updating $\mathbf{W}$ and $\mathbf{C}$ by using APG algorithm and ADMM algorithm in the following.

A.1 Initialization

The general performance of multi-task learning algorithm relies on the initial estimates of parameters. Following the initial procedure described in [15], we first learn the task weights matrix with single task learning as follows,

$$\mathbf{W}_{\text{init}} = \underset{\mathbf{W}}{\operatorname{argmin}} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(\mathbf{w}_i^\top \mathbf{x}_j^i, y_j^i\right) + \sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} \|\mathbf{W}\|_F^2.$$

Then, we compute the singular value decomposition of $\mathbf{W}_{\text{init}} \in \mathbb{R}^{d\times m}$ to obtain the left-singular vectors $\mathbf{U} \in \mathbb{R}^{d\times d}$, singular values matrix $\mathbf{\Sigma} \in \mathbb{R}^{d\times m}$ and right-singular vectors $\mathbf{V} \in \mathbb{R}^{m\times m}$. The initial values of $\mathbf{W}$ and $\mathbf{C}$ are given by $\mathbf{W}^0 = \mathbf{U}\mathbf{\Sigma}$ and $\mathbf{C}^0 = \mathbf{V}^\top$, respectively.

A.2 Details of Update Procedures for (12)

The auxiliary matrix $\mathbf{E}$, the task weights matrix $\mathbf{W}$ and the Lagrangian multiplier $\mathbf{\Lambda}$ in Eq. (12) are updated as follows.

A.2.1 Updating $\mathbf{E}$. The auxiliary matrix $\mathbf{E}$ is updated by solving the following problem,

$$\begin{aligned}&\min_{\mathbf{E}} \frac{\rho}{2}\|\mathbf{WD} - \mathbf{E}\|_F^2 + \operatorname{Tr}\left(\mathbf{\Lambda}^\top(\mathbf{WD} - \mathbf{E})\right) + \lambda_2 \left\|\mathbf{E}^\top\right\|_{1,2} \\ =&\min_{\mathbf{E}} \frac{1}{2}\left\|\mathbf{E} - \mathbf{WD} - \frac{\mathbf{\Lambda}}{\rho}\right\|_F^2 + \frac{\lambda_2}{\rho}\left\|\mathbf{E}^\top\right\|_{1,2},\end{aligned}$$

which can be solved by applying the proximal operator on each column of $\mathbf{E}$ separately. To be specific, for the i-th column of $\mathbf{E}$, the closed-form solution is

$$\mathbf{e}_i^{k+1} = \max\left(0, 1 - \frac{\lambda_2}{\rho\|\hat{\mathbf{e}}_i\|_2}\right)\hat{\mathbf{e}}_i, \tag{16}$$

where $\hat{\mathbf{e}}_i = \left[\mathbf{W}^k\mathbf{D} + \mathbf{\Lambda}^k/\rho\right]_i$ is the i-th column of the matrix $\left(\mathbf{W}^k\mathbf{D} + \mathbf{\Lambda}^k/\rho\right)$.

A.2.2 Updating $\mathbf{W}$. Next, the task weights matrix $\mathbf{W}$ can be updated by solving the following problem,

$$\min_{\mathbf{W}} \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} l\left(\mathbf{w}_i^\top \mathbf{x}_j^i, y_j^i\right) + \lambda_1\|\mathbf{W}\|_F^2 + \frac{\rho}{2}\left\|\mathbf{WD} - \mathbf{E} + \frac{\mathbf{\Lambda}}{\rho}\right\|_F^2,$$

which can be solved with the gradient descent method. In particular, we employ the L-BFGS [26] for efficiency. For regression problems with the squared loss, we can compute the gradient of the above function as follows,

$$\nabla_{\mathbf{w}_i} = \frac{1}{n_i}\mathbf{X}^{i\top}(\mathbf{X}^i\mathbf{w}_i - \mathbf{y}^i) + 2\lambda_1\mathbf{w}_i + (\rho\mathbf{WD} - \rho\mathbf{E} + \mathbf{\Lambda})\mathbf{d}_i, \tag{17}$$

where $\mathbf{d}_i$ is the i-th column of matrix $\mathbf{D}$. On the other hand, for binary classification problem with the logistic loss, the gradient is given in the following form,

$$\begin{aligned}\nabla_{\mathbf{w}_i} = &\sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} -\frac{\exp\left(-y_j^i\left(\mathbf{w}_i^\top\mathbf{x}_j^i\right)\right) y_j^i\mathbf{x}_j^i}{1 + \exp\left(-y_j^i\left(\mathbf{w}_i^\top\mathbf{x}_j^i\right)\right)} + 2\lambda_1\mathbf{w}_i \\ &+ (\rho\mathbf{WD} - \rho\mathbf{E} + \mathbf{\Lambda})\mathbf{d}_i.\end{aligned} \tag{18}$$

A.2.3 Updating $\mathbf{\Lambda}$. Finally, the Lagrangian multiplier $\mathbf{\Lambda}$ is updated as follows,

$$\mathbf{\Lambda}^{k+1} = \mathbf{\Lambda}^k + \rho\left(\mathbf{W}^{k+1}\mathbf{D} - \mathbf{E}^{k+1}\right). \tag{19}$$

The primal and dual residuals $\mathbf{r}_1^{k+1}$ and $\mathbf{s}_1^{k+1}$ are given by

$$\mathbf{r}_1^{k+1} = \mathbf{W}^{\mathbf{k+1}}\mathbf{D} - \mathbf{E}^{k+1}, \tag{20}$$

$$\mathbf{s}_1^{k+1} = \rho\left(\mathbf{E}^{k+1} - \mathbf{E}^k\right), \tag{21}$$

which can be used to check the convergence.

A.3 Details of Update Procedures for (15)

The auxiliary matrix $\mathbf{F}$, the task relationship matrix $\mathbf{C}$ and the Lagrangian multiplier $\mathbf{\Gamma}$ in Eq. (15) are updated as follows.

A.3.1 Updating $\mathbf{F}$. The auxiliary matrix $\mathbf{F}$ is updated by solving the following optimization problem,

$$\begin{aligned}&\min_{\mathbf{F}} \frac{\rho}{2}\|\mathbf{F} - \mathbf{W} + \mathbf{WC}\|_F^2 + \operatorname{Tr}\left(\mathbf{\Gamma}^\top(\mathbf{F} - \mathbf{W} + \mathbf{WC})\right) + \lambda_2\left\|\mathbf{F}^\top\right\|_{1,2} \\ =&\min_{\mathbf{F}} \frac{1}{2}\left\|\mathbf{F} - \mathbf{W} + \mathbf{WC} + \frac{\mathbf{\Gamma}}{\rho}\right\|_F^2 + \frac{\lambda_2}{\rho}\left\|\mathbf{F}^\top\right\|_{1,2},\end{aligned}$$

which can be solved by applying the proximal operator on each column of $\mathbf{F}$ separately. To be specific, for the i-th column of $\mathbf{F}$, the closed-form solution is

$$\mathbf{f}_i^{k+1} = \max\left(0, 1 - \frac{\lambda_2}{\rho\|\hat{\mathbf{f}}_i\|_2}\right)\hat{\mathbf{f}}_i, \tag{22}$$

where $\hat{\mathbf{f}}_i = \left[\mathbf{W} - \mathbf{WC}^k - \mathbf{\Gamma}^k/\rho\right]_i$ is the i-th column of the matrix $\left(\mathbf{W} - \mathbf{WC}^k - \mathbf{\Gamma}^k/\rho\right)$.

A.3.2 Updating $\mathbf{C}$. Next, the task relationship matrix $\mathbf{C}$ can be updated by solving the following optimization problem,

$$\min_{\mathbf{C}} \frac{\rho}{2}\left\|\mathbf{WC} - \left(\mathbf{W} - \mathbf{F} - \frac{\mathbf{\Gamma}}{\rho}\right)\right\|_F^2 + \lambda_2\|\mathbf{C}\|_{1,2}, \tag{23}$$

which is consisted of a smooth component and a non-smooth component. We leverage the well-developed proximal algorithm FISTA to solve it efficiently.

A.3.3 Updating $\mathbf{\Gamma}$. Finally, the Lagrangian multiplier $\mathbf{\Gamma}$ is updated as follows,

$$\mathbf{\Gamma}^{k+1} = \mathbf{\Gamma}^k + \rho\left(\mathbf{F}^{k+1} - \left(\mathbf{W} - \mathbf{WC}^{k+1}\right)\right). \tag{24}$$

The primal and dual residuals $\mathbf{r}_2^{k+1}$ and $\mathbf{s}_2^{k+1}$ are given by

$$\mathbf{r}_2^{k+1} = \mathbf{F}^{k+1} - \left(\mathbf{W} - \mathbf{WC}^{k+1}\right), \tag{25}$$

$$\mathbf{s}_2^{k+1} = \rho\left(\mathbf{F}^{k+1} - \mathbf{F}^k\right). \tag{26}$$

which are used to check the convergence of this step.