Flexible Clustered Multi-Task Learning by Learning Representative Tasks

Qiang Zhou and Qi Zhao, *Member, IEEE*

Abstract—Multi-task learning (MTL) methods have shown promising performance by learning multiple relevant tasks simultaneously, which exploits to share useful information across relevant tasks. Among various MTL methods, clustered multi-task learning (CMTL) assumes that all tasks can be clustered into groups and attempts to learn the underlying cluster structure from the training data. In this paper, we present a new approach for CMTL, called flexible clustered multi-task (FCMTL), in which the cluster structure is learned by identifying representative tasks. The new approach allows an arbitrary task to be described by multiple representative tasks, effectively soft-assigning a task to multiple clusters with different weights. Unlike existing counterpart, the proposed approach is more flexible in that (a) it does not require clusters to be disjoint, (b) tasks within one particular cluster do not have to share information to the same extent, and (c) the number of clusters is automatically inferred from data. Computationally, the proposed approach is formulated as a row-sparsity pursuit problem. We validate the proposed FCMTL on both synthetic and real-world data sets, and empirical results demonstrate that it outperforms many existing MTL methods.

Index Terms—Clustered multi-task learning, representative task, group sparsity

1 Introduction

Many real-world applications involve the learning of multiple relevant tasks. For example, in fine grained visual recognition, the task is to recognize many but closely relevant object categories [48]. Instead of learning them separately, previous works [1], [3], [21], [26] have shown that the generalization performance can be improved by learning them jointly. This idea is called multi-task learning (MTL) [4], [10], [12], [23], [25], [34], [38], [41], [43], [51], [52], [58] and it attempts to share useful information across multiple relevant tasks by exploiting their intrinsic relationships. Multi-task learning has been applied to many areas including computational biology [28], [33], [35], [62], computer vision [47], [56], natural language processing [1], [40] and music recommendation [17].

A large number of existing MTL methods assume that all tasks are relevant and share information to the same extent. For example, Regularized MTL [21] enforces that the model parameters of all tasks are similar to each other, and a set of common features are imposed to share in multi-task feature learning methods [3], [12], [36]. However, this assumption is often invalid in many practical problems, and the performance of MTL can be significantly degraded due to the negative transfer among unrelated tasks.

Various methods have been proposed to address the negative transfer problem. Some works propose to use prior knowledge on task relationship structure to guide information sharing among multiple tasks, for example, with pairwise task relationship network [20], [31], tree-guided MTL [33], and graph-guided MTL [14]. While the above works make use of prior knowledge on relevant tasks, Romera-Paredes et al. [42] further exploit prior information on irrelevant tasks to improve the performance of target tasks that are to be learned. The assumption of all methods along this line, however, is that the task relationships are available as a priori, which is not always true.

Instead of assuming all tasks to be relevant, clustered multi-task learning (CMTL) assumes that all tasks can be clustered into disjoint groups [26]. Compared to Regularized MTL [21] that enforces all tasks to be similar to each other, the assumption of CMTL is that the model parameters for tasks in the same group should be close to each other.

Despite the success of CMTL, there are two major limitations in existing methods: first, the number of clusters needs to be specified, while it is rarely available in real-world tasks. Second, CMTL assumes that all tasks can be clustered into a set of disjoint groups and tasks in the same cluster share information to the same extent. This assumption, however, may not be true and such hard-assignment can lead to either negative transfer (some tasks that are not strongly relevant are forced to cluster into the same group) or ineffective sharing across all tasks (some relevant tasks are clustered into different groups).

Motivated by representatives/exemplars used in dictionary learning and data clustering [18], [19], this work proposes a new approach for clustered multi-task learning. In this approach, a subset of tasks (called representative tasks) are identified and used to describe tasks. An arbitrary task is allowed to be described by multiple representative tasks for an accurate representation. Since each representative task establishes one cluster, an arbitrary task in this framework can be assigned to more than one cluster with different weights, allowing tasks in the same cluster to share information to different extents. Furthermore, the number of clusters is automatically inferred from training data

• *The authors are with the Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117583. E-mail: zhouqiang@u.nus.edu, eleqiz@nus.edu.sg.*

Manuscript received 24 Jan. 2014; revised 9 June 2015; accepted 25 June 2015. Date of publication 5 July 2015; date of current version 13 Jan. 2016.
Recommended for acceptance by F. Fleuret.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2452911

instead of manually set. We call the proposed approach flexible clustered multi-task learning (FCMTL).

1.1 Main Idea

The key insight is that representative tasks can effectively describe all tasks in multi-task learning. Intuitively, if one task is selected by another task as the representative task, it means that these two tasks are relevant and information can be shared between them. Furthermore, those tasks which select a common representative task can be considered as clustered into the same group and sharing information with the same representative task. Therefore, clustering tasks in multi-task learning can be casted as identifying a set of representative tasks where each representative task is considered as one cluster.

In practice, one task may have multiple representative tasks since one representative task generally has limited power in characterizing all important features of an arbitrary task. Therefore, unlike previous CMTL works, we allow one task to be clustered into multiple clusters, and with different weights. The weights determine how much information one task shares with each of its representative tasks.

The proposed approach involves identifying representative tasks and using them to cluster all tasks. Intuitively, the objective is that a small number of representative tasks can encode well all tasks thus we formulate it by minimizing the number of representative tasks with some constraints. The problem, however, is intractable due to the NP-hard property of ℓ_0-norm. Alternatively, we consider optimizing its convex surrogate and formulating it as a row-sparsity pursuit problem. We adopt the block coordinate descent optimization algorithm to solve the optimization problem in our approach.

1.2 Contributions

In this work, we propose a flexible clustered multi-task learning approach. The advantages of the new method can be summarized as: (a) it allows each task to be assigned to multiple clusters thus it does not require the clusters to be disjoint, (b) tasks within the same group does not have to share information to the same extent, and (c) the number of clusters can be automatically learned instead of set as a priori. We demonstrated the effectiveness of the proposed FCMTL on common data sets for MTL research, as well as for fine grained visual recognition.

The remainder of this paper is organized as follows: we review related multi-task learning works in Section 2. We then introduce the proposed Flexible Clustered Multi-Task Learning and its kernel extension in Section 3. Extensive experimental results are presented in Section 4, followed by conclusions and future works in Section 5.

2 Related Work on Multi-Task Learning

This section discusses several previous multi-task learning works that are relevant to the proposed approach and shows the differences between the proposed approach and them.

Regularized MTL [21] assumes that all tasks are similar so they can all be clustered into one cluster. To this end, it is a special case of the proposed FCMTL with all tasks selecting only one representative task.

In order to deal with outlier tasks in multi-task learning, Robust MTL [13] uses a low-rank structure to capture the relevant tasks and models the outlier tasks by a group sparsity structure. There are at least two important differences between the referred work and the proposed approach: (a) the referred work aims at identifying irrelevant tasks from multiple tasks, while our goal is to cluster all tasks into groups. (b) Although both works include a group sparsity regularization, the motivation is totally different. [13] uses it to model outlier tasks, while the proposed work uses it to regularize the number of representative tasks in clustered multi-task learning.

CMTL [26] assumes that all tasks are clustered into some disjoint groups and learns the cluster structure from data. However, such hard-assignment can lead to either negative transfer or ineffective sharing across all tasks. In addition, CMTL limits itself in modeling the exact cluster structure due to the spectral relaxation used in [26], [61]. Furthermore, compared to CMTL, available prior knowledge can be easily incorporated into the proposed approach by introducing additional constraints on the assignment matrix that describes the assignment of all tasks to representative tasks. Clustering tasks into disjoint groups has also been exploited in [30] to improve multi-task feature learning [3]. The task relatedness in [30] is modeled as learning shared features among the tasks, while the proposed FCMTL assumes that the model parameters of relevant tasks are similar. Unlike CMTL that clusters tasks at the task level, Zhong and Kwok [60] have investigated how to cluster tasks at the feature level. Recently, the equivalence relationship between alternating structure optimization [1] and CMTL has also been studied [61].

Several works [6], [7], [39], [44], [53] have studied multi-task learning in the context of Gaussian process, which assumes that the models of different tasks are generated from a common distribution. In [7], the authors explicitly model the task relationships via a task covariance matrix in their formulations. In their work, the final covariance matrix is a Kronecker product of the task covariance matrix and the sample variance matrix. As the method needs to calculate the inverse for the covariance matrix, its computational cost grows cubically with both the sample size and the task number, which does not scale well to large-scale problems. Zhang and Yeung [55] propose a framework to automatically learn task relationships via a regularization formulation, which uses a matrix-variate distribution to model the model parameters of multiple tasks. In their formulation, a positive semi-definite constraint is imposed on the task relationship matrix, which is not sufficiently strong in some cases, e.g. all tasks follow a cluster structure. Compared to [55], the proposed approach encourages row-sparsity on the assignment matrix which is more effective.

3 Proposed Approach

In this section, we introduce the proposed flexible clustered multi-task learning (FCMTL) approach. The key insight of FCMTL is that a subset of tasks in multi-task learning can be used to represent other tasks due to the similarity among multiple tasks. We call this subset as representative tasks and use them as bridges between any two relevant tasks. In general, we aim to identify these representative tasks and use them for clustered multi-task learning.

In the rest of this section, we first describe the concept of representative tasks and ways to identify them, followed by the introduction of the FCMTL approach that incorporates the representative tasks for multi-task learning. We then discuss how to solve the optimization problem by the block coordinate descent procedure. We also mention how to extend the proposed approach to nonlinear kernel functions.

Problem setup. Suppose we are given m learning tasks, the n_i training samples associated with the ith task are $\{(x_1^i, y_1^i), \dots, (x_{n_i}^i, y_{n_i}^i)\}$ where $x_j^i \in \mathbb{R}^d$ is the input (d is the feature dimension) and the corresponding output is $y_j^i \in \mathbb{R}$ for regression problems and $y_j^i \in \{-1, 1\}$ for binary classification problems. For the ith task, the goal is to learn a linear function $f_i(x_j^i) = w_i^T x_j^i + b_i$ where $w_i \in \mathbb{R}^d$ is the model parameter for the ith task. $\mathbf{W} = [w_1, \dots, w_m] \in \mathbb{R}^{d\times m}$ and $\mathbf{b} = [b_1, \dots, b_m]^T$ denote the model parameters for all tasks.

3.1 Representative Tasks

Representative tasks are a subset of the given m tasks. Intuitively, a representative task is one that other tasks are relevant to, and can be used to describe or represent other tasks. Formally, if the rth task is selected by the gth task as a representative task, it is expected that the model parameters for the gth task (w_g) is similar to those of the rth task (w_r). To describe one task in an accurate way, one representative task can be insufficient to capture all important characteristics of the task. Furthermore, the similarity between an arbitrary task and each one of its representative tasks may be different as different representative tasks describe different aspects of the task.

Let $\mathbf{Z} \in \mathbb{R}^{m\times m}$ denote the assignment of representative tasks for all tasks. Specifically, we consider $\mathbf{Z}_{ik}$ ($\mathbf{Z}_{ik} \in [0, 1]$) as the probability that the kth task selects the ith task as its representative task. If $\mathbf{Z}_{ik} = 0$, the ith task will not be the representative task of the kth task, and if $\mathbf{Z}_{ik} = 1$, it denotes that the ith task will be the only one representative task of the kth task. Otherwise, the ith task will be one of the representative tasks of the kth task when $0 < \mathbf{Z}_{ik} < 1$. To ensure that the total probability of all tasks selected by one task as its representative tasks sums up to one, we impose a constraint on $\mathbf{Z}$: $\sum_{i=1}^m \mathbf{Z}_{ik} = 1$.

3.1.1 Identifying Representative Tasks

Since each task is expected to be similar to its representative task, we determine the representative tasks for each task according to the distance or dissimilarity of the model parameters between it and any other tasks. Intuitively, the goal is to minimize the weighted distance between each task and its representative tasks. In this work, we define the distance between two tasks as the square of Euclidean distance between their model parameters, thus the weighted distance of the kth task to all its representative tasks is formulated as

$$\sum_{i=1}^m \mathbf{Z}_{ik}\|w_i - w_k\|_2^2. \tag{1}$$

It is easy to verify that each task will select itself as the only representative task if we straightforwardly minimize (1) with the mentioned constraint on $\mathbf{Z}$ ($\sum_{i=1}^m \mathbf{Z}_{ik} = 1$). In this setting, one task cannot benefit from its representative tasks since no relationship has been established between any two tasks. This will lead to the conventional single-task learning (STL).

In many real-world problems, tasks are relevant. It is thus highly desirable to establish relationships for relevant tasks in a multi-task learning framework, which enables these relevant tasks to share useful information with each other. To encourage information sharing, the number of representative tasks is expected to be small. Consequently, relevant tasks will select common representative tasks and establish relationships through their representative tasks.

Formally, we formulate the representative task selection problem as row-sparsity pursuit on the assignment matrix $\mathbf{Z}$. Take the ith row in $\mathbf{Z}$ for example, if at least one element in this row is non-zero, it means that the ith task is a representative task to those tasks indexed by non-zero elements in this row. Otherwise, no task has selected the ith task as a representative task if all elements in the ith row are zero. Hence, the row-sparsity pursuit aims to minimize the number of non-zero rows in $\mathbf{Z}$. Following previous works on group sparsity [27], [54], we use the ℓ_q-norm of one vector to determine whether all elements are zero or not. Let $\mathbf{Z}(i,:)$ denote the ith row in $\mathbf{Z}$, then $\|\mathbf{Z}(i,:)\|_q$ as the ℓ_q-norm of $\mathbf{Z}(i,:)$ will be non-zero except $\mathbf{Z}(i,:) = \mathbf{0} \in \mathbb{R}^m$. The number of representative tasks can then be calculated as the number of rows in $\mathbf{Z}$ whose ℓ_q is non-zero. Let $\mathcal{I}(x)$ denote the indicator function whose function value is zero if $x = 0$ and is one otherwise, the non-zero rows in $\mathbf{Z}$ can be obtained by the $\ell_{0,q}$-norm of $\mathbf{Z}$

$$\|\mathbf{Z}\|_{0,q} = \sum_{i=1}^m \mathcal{I}\Big(\|\mathbf{Z}(i,:)\|_q\Big).$$

Overall, the problem of learning representative tasks can be formulated as

$$\begin{aligned} \min_{\mathbf{Z}} \ & \lambda \sum_{i=1}^m \sum_{k=1}^m \mathbf{Z}_{ik}\|w_i - w_k\|_2^2 + \mu\|\mathbf{Z}\|_{0,q} \\ s.t.\ & \mathbf{0} \preceq \mathrm{vec}(\mathbf{Z}) \preceq \mathbf{1}_{mm}, \mathbf{Z}^T\mathbf{1}_m = \mathbf{1}_m, \end{aligned} \tag{2}$$

where $\preceq$ denotes componentwise inequality for vector, $\mathrm{vec}(\cdot)$ denotes vectorization operator, and $\mathbf{1}_m$ is a m-dimensional vector where all components are one.

3.2 Flexible Clustered Multi-Task Learning

Next, we introduce a new multi-task learning approach by incorporating the idea of representative tasks into multi-task learning. Among various multi-task learning methods, our focus is a new clustered multi-task learning approach. Specifically, we consider tasks that select a common representative task as a group, then all tasks can be clustered into groups based on their representative tasks. According to the definition of the representative task, tasks assigned to the same group have similar model parameters. Formally, we formulate the proposed approach as follows

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\mathbf{Z}} \ & \mathcal{L}(\mathbf{W}) + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \frac{\lambda}{2}\sum_{i=1}^m \sum_{k=1}^m \mathbf{Z}_{ik}\|w_i - w_k\|_2^2 \\ & + \frac{\mu}{2}\|\mathbf{Z}\|_{0,q} \\ s.t.\ & \mathbf{0} \preceq \mathrm{vec}(\mathbf{Z}) \preceq \mathbf{1}_{mm},\ \mathbf{Z}^T\mathbf{1}_m = \mathbf{1}_m, \end{aligned} \tag{3}$$

where $\mathcal{L}(\mathbf{W})$ is the empirical loss, which is squared loss for regression problem

$$\mathcal{L}(\mathbf{W}) = \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \left(\left(w_i^T x_j^i + b_i \right) - y_j^i \right)^2,$$

and logistic loss for binary classification problem

$$\mathcal{L}(\mathbf{W}) = \sum_{i=1}^{m} \frac{1}{n_i} \sum_{j=1}^{n_i} \log \left(1 + \exp\left(-y_j^i \left(w_i^T x_j^i + b_i \right) \right) \right).$$

In (3), the squared Frobenius norm $\|\mathbf{W}\|_F^2 = \mathrm{Tr}(\mathbf{W}\mathbf{W}^T)$ is used to control the complexity of each linear model. The third term is used to enforce the similarity between each task and their representative tasks, and the last term is to regularize the number of representative tasks or clusters. The first constraint expresses that the probability of each task being assigned to a particular cluster is from 0 to 1, and the second constraint ensures that the probability of each task assigned to all clusters sums up to 1.

Previous CMTL methods assume that the number of clusters is known as a priori, while it is usually unavailable in practice. In comparison, the proposed approach does not require the number beforehand and automatically infers it from training data. Furthermore, compared to previous CMTL works that assume each task to be assigned to only one cluster, an arbitrary task in the proposed approach is allowed to be assigned to multiple clusters with different weights. This enables each task to share information with its relevant tasks to the right extent.

The optimization problem in (3) involves the ℓ_0-norm and it is intractable due to the NP-hard property of the ℓ_0-norm. Following the previous work [54], we relax the ℓ_0-norm by its convex proxy ℓ_1-norm, so the last term becomes the $\ell_{1,q}$-norm of $\mathbf{Z}$: $\|\mathbf{Z}\|_{1,q} = \sum_{i=1}^{m} \|\mathbf{Z}(i,:)\|_q$. According to [27], [54], the value of q is typically chosen from $\{2, \infty\}$. If $q = 2$, the values of the elements in a row can be different within the range 0 to 1, while $q = \infty$ encourages the entire row to be the same value. Obviously, $q = 2$ is more suitable in the proposed approach, which allows tasks to select representative tasks with different probabilities. Consequently, the final formulation of the proposed FCMTL is

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\mathbf{Z}} \ & \mathcal{L}(\mathbf{W}) + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \frac{\lambda}{2}\sum_{i=1}^{m}\sum_{k=1}^{m} \mathbf{Z}_{ik}\|w_i - w_k\|_2^2 \\ & + \frac{\mu}{2}\|\mathbf{Z}\|_{1,2} \\ s.t. \ & \mathbf{0} \preceq \mathrm{vec}(\mathbf{Z}) \preceq \mathbf{1}_{mm}, \ \mathbf{Z}^T\mathbf{1}_m = \mathbf{1}_m. \end{aligned} \tag{4}$$

3.3 Solving the Optimization Problem

In order to solve the problem in (4), we adopt block coordinate descent method by iteratively updating $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{Z}$. Specifically, when updating $\mathbf{W}$ and $\mathbf{b}$ with fixed $\mathbf{Z}$, the optimization problem can be written as

$$\min_{\mathbf{W},\mathbf{b}} \quad \mathcal{L}(\mathbf{W}) + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \frac{\lambda}{2}\sum_{i=1}^{m}\sum_{k=1}^{m} \mathbf{Z}_{ik}\|w_i - w_k\|_2^2. \tag{5}$$

Proposition 1. *The optimization problem (5) is convex with respect to* $\mathbf{W}$ *and* $\mathbf{b}$.

Proof. The proof is shown in Appendix A. □

Problem (5) can be solved by performing gradient descent on $\mathbf{W}$ and $\mathbf{b}$. Here, we apply the accelerated proximal gradient (APG) method [5], [37] to optimize the problem. APG has been extensively used to solve machine learning problems [11], [12], [24], [59], [60], [61] due to the optimal convergence rate among all first-order methods.

Next, with fixed $\mathbf{W}$ and $\mathbf{b}$, the subproblem that minimizes (4) over $\mathbf{Z}$ can be written as

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \frac{\lambda}{2}\mathrm{Tr}(\mathbf{D}^T\mathbf{Z}) + \frac{\mu}{2}\|\mathbf{Z}\|_{1,2} \\ s.t. \quad & \mathbf{0} \preceq \mathrm{vec}(\mathbf{Z}) \preceq \mathbf{1}_{mm}, \ \mathbf{Z}^T\mathbf{1}_m = \mathbf{1}_m, \end{aligned} \tag{6}$$

where $\mathbf{D} \in \mathcal{R}^{m\times m}$ with $\mathbf{D}_{ik} = \|w_i - w_k\|_2^2$.

Solving the optimization problem in (6) can be considered as identifying representative tasks for all tasks. The following theorem establishes the conditions for (a) each task selects itself as its only representative task, and (b) only one representative task is selected for all tasks. Otherwise, multiple representative tasks will be learned for all tasks.

Theorem 1. *In the optimization problem with fixed* $\mathbf{W}$ *and* $\mathbf{b}$ *(6), let* $\beta = \mu/\lambda$ *and* $\mathbf{D}_i$ *denotes the ith row of* $\mathbf{D}$,

$$k = \arg\min_i \mathbf{D}_i\mathbf{1}_m, \tag{7}$$

$$\beta_{\min} = \min_j\left(\min_{i\neq j}\mathbf{D}_{ij} - \mathbf{D}_{jj}\right), \tag{8}$$

$$\beta_{\max} = \max_{i\neq k}\frac{\sqrt{m}}{2}\frac{\|\mathbf{D}_i - \mathbf{D}_k\|_2^2}{(\mathbf{D}_i - \mathbf{D}_k)\mathbf{1}_m}, \tag{9}$$

when $\beta \le \beta_{\min}$*, the optimal* $\mathbf{Z}$ *of the optimization problem (6) is an identity matrix, which means each task selects itself as its only representative task and the method reduces to single-task learning. When* $\beta \ge \beta_{\max}$*, all tasks select the kth task as their only common representative task and the optimal solution is* $\mathbf{Z} = \mathbf{e}_k\mathbf{1}_m^T$*, where* $\mathbf{e}_k \in \mathbb{R}^m$ *denotes the vector whose elements are all zero except its kth element which equals to 1.*

Proof. The proof is provided in Appendix B. □

The problem in (6) involves certain constraints and we adopt the alternating direction method of multipliers (ADMM) [8] to solve it. In order to use ADMM, we first convert (6) to the following equivalent problem

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \lambda\,\mathrm{Tr}(\mathbf{D}^T\mathbf{Z}) + g(\mathbf{P}) + \mu\|\mathbf{Q}\|_{1,2} \\ s.t. \quad & \mathbf{0} \preceq \mathrm{vec}(\mathbf{Z}) \preceq \mathbf{1}_{mm}, \ \mathbf{Z}^T\mathbf{1}_m = \mathbf{1}_m \\ & \mathbf{Z} = \mathbf{P}, \ \mathbf{P} = \mathbf{Q}, \end{aligned} \tag{10}$$

where $g(\mathbf{P})$ is the indicator function of convex set $\{\mathcal{C} = \mathbf{P} | 0 \preceq \mathrm{vec}(\mathbf{P}) \preceq \mathbf{1}_{mm}\}$. Then, the augmented Lagrangian for (10) can be written as

$$\begin{aligned} & L_\rho(\mathbf{Z},\mathbf{P},\mathbf{Q},\mathbf{C}_1,\mathbf{C}_2,\mathbf{C}_3) \\ & = \lambda\,\mathrm{Tr}(\mathbf{D}^T\mathbf{Z}) + g(\mathbf{P}) + \mu\|\mathbf{Q}\|_{1,2} \\ & + \langle\mathbf{C}_1, \mathbf{Z}-\mathbf{P}\rangle + \langle\mathbf{C}_2, \mathbf{P}-\mathbf{Q}\rangle + \langle\mathbf{C}_3, \mathbf{Z}^T\mathbf{1}_m - \mathbf{1}_m\rangle \\ & + \frac{\rho}{2}\left(\|\mathbf{Z}-\mathbf{P}\|_F^2 + \|\mathbf{P}-\mathbf{Q}\|_F^2 + \|\mathbf{Z}^T\mathbf{1}_m - \mathbf{1}_m\|_2^2\right), \end{aligned} \tag{11}$$

where $\mathbf{C}_1 \in \mathbb{R}^{m\times m}, \mathbf{C}_2 \in \mathbb{R}^{m\times m}, \mathbf{C}_3 \in \mathbb{R}^m$ are Lagrange multipliers and ρ is a positive penalty parameter. Details of the ADMM procedure for (11) are described in Appendix C.

The entire optimization procedure will be terminated when the changes of $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{Z}$ between two consecutive iterations are all small. Although the algorithm does not guarantee a global optimum, we found it perform well in our experiments. We summarize the optimization procedure for FCMTL in Algorithm 1.

In addition, we also show that the proposed FCMTL (4) can be easily extended to nonlinear kernel functions and the details are shown in Appendix D.

Algorithm 1. Solving the Optimization Problem in (4)

1: **Input:** Training data $\{\left(x_j^i, y_j^i|_{j=1}^{n_i}\right), i = 1, \ldots, m\}$.
2: Initialize $\mathbf{W}$ and $\mathbf{b}$ by single-task learning (with $\mathbf{Z} = \mathbf{I}$ in (5)).
3: **while** not converged **do**
4: Update $\mathbf{Z}$ by using the ADMM algorithm to solve (11)
5: Update $\mathbf{W}$ and $\mathbf{b}$ by using the APG method to optimize (5)
6: **end while**
7: **Output:** $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{Z}$

4 EXPERIMENTS

To evaluate the performance of the proposed approach, we perform extensive experiments on both synthetic and real-world data sets. We compare the proposed approach with the following baseline and multi-task learning methods:

STL: single-task learning method as a baseline, in which all tasks are learned separately.

Regularized MTL [21]: the method assumes that all tasks are relevant and enforces their model parameters to be close to a single center.

Dirty MTL [29]: model parameters of all tasks are considered as two parts: the first part is shared by all tasks and the second part represents specific features of each task.

Robust MTL [13]: instead of assuming all tasks to be relevant, this work aims at identifying outlier tasks in multi-task learning.

Group MTFL [30]: the method improves multi-task feature learning [2] by clustering tasks into disjoint groups and learning shared features in each group. Notice that, following their paper, we repeat the gradient descent with 10 random initializations and choose the best local optimum among them.

FlexTClus [60]: this work also decomposes the model parameters to two parts: one part models the shared features by all tasks and another part models specific features of each task. The shared part is clustered using ℓ_1.

MTRL [55]: the work learns the relationships between tasks and uses the learned task relationships to improve the multi-task learning methods.

CMTL [26]: all tasks are clustered into disjoint groups and tasks in the same group are enforced to have similar model parameters.

4.1 Synthetic Data Sets

We evaluate comparative methods on three different synthetic data sets as sanity check to show that the proposed approach can learn the underlying cluster structure of tasks in various scenarios. Specifically, the task is a linear regression problem and the dimension of the input feature $d = 100$. The input data are generated from $x \sim \mathcal{N}(0, \mathbf{I})$ and the output of the ith task is obtained by $y_i \sim w_i^T x + \mathcal{N}(0, 150)$. For each task, we generate 30 samples as training data and 100 samples for testing. In order to tune the regularization parameters of all methods, we generate a validation set with 100 samples separately for each data set. Note that the synthetic data sets are generated using a similar procedure as reported in [26].

TABLE 1
Mean and Standard Deviation of NMSE of All Methods on the Three Synthetic Data Sets

	Data Set 1	Data Set 2	Data Set 3
STL	0.703 ± 0.011	0.719 ± 0.015	0.698 ± 0.014
Regularized MTL	0.605 ± 0.040	0.627 ± 0.016	0.638 ± 0.020
Dirty MTL	0.612 ± 0.022	0.670 ± 0.015	0.653 ± 0.013
Robust MTL	0.078 ± 0.010	0.253 ± 0.014	0.319 ± 0.017
Group MTFL	0.363 ± 0.018	0.504 ± 0.026	0.587 ± 0.032
FlexTClus	0.498 ± 0.019	0.552 ± 0.025	0.560 ± 0.187
MTRL	0.147 ± 0.020	0.293 ± 0.016	0.360 ± 0.024
CMTL	0.073 ± 0.010	0.214 ± 0.012	0.303 ± 0.016
FCMTL	0.040 ± 0.017	0.129 ± 0.017	0.212 ± 0.024

4.1.1 Data Set 1

This data set consists of four clusters and each cluster contains 10 tasks. All 100 dimensions are randomly divided into four disjoint groups and each group is assigned to only one cluster. The model parameters for tasks from a particular cluster are non-zero only for corresponding dimensions, and are zero for all other dimensions, so that different clusters are orthogonal to each other. For the ith task in the cth cluster, the value of each dimension is the sum of its cluster center $\overline{w}_c$ and a task specific component w_i, where $\overline{w}_c \sim \mathcal{N}(0, 900)$ and $w_i \sim \mathcal{N}(0, 16)$.

4.1.2 Data Set 2

This data set is the same as data set 1 except we generate the four cluster centers from the first 96 dimensions and the remaining four dimensions for all tasks are generated from $\mathcal{N}(0, 16)$.

4.1.3 Data Set 3

This data set is the same as data set 2 except we generate another five outlier tasks from $50 + \mathcal{N}(0, 900)$. All dimensions are non-zero in these outlier tasks.

We use the normalized mean square error (NMSE) as the evaluation measure, which is obtained by using the variance of the ground truth to normalize the mean square error. Table 1 reports the mean and standard derivation over 10 trials on the three synthetic data sets.

It is shown that the proposed FCMTL performs the best on all three data sets. Furthermore, all multi-task learning methods outperform single-task learning. However, the improvements of Regularized MTL and Dirty MTL are insignificant due to the invalid assumption that all tasks are related. Robust MTL performs well on the first data set as

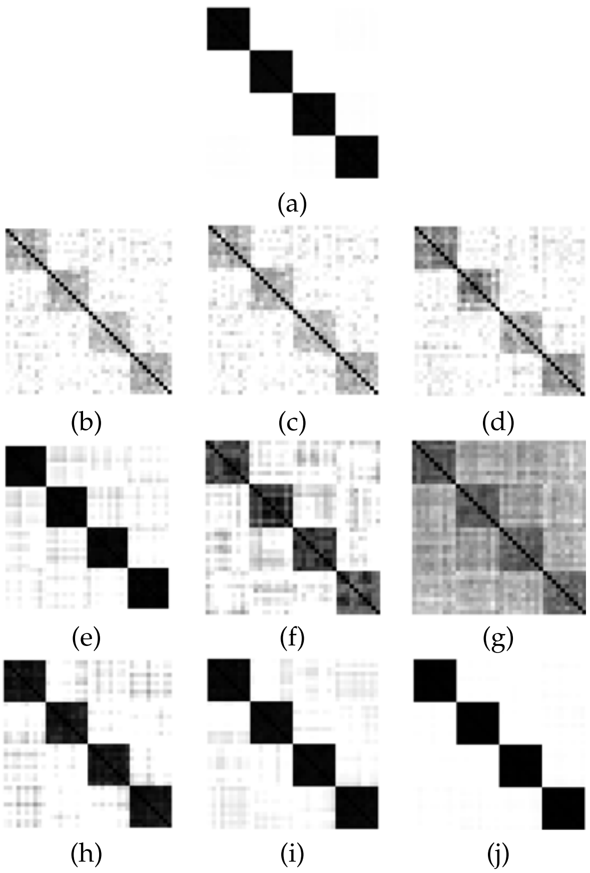


Fig. 1. The correlation matrices of different methods: (a) Ground Truth, (b) STL, (c) Regularized MTL, (d) Dirty MTL, (e) Robust MTL, (f) Group MTFL, (g) FlexTClus, (h) MTRL, (i) CMTL, and (j) FCMTL. Darker color indicates higher correlation.

all tasks indeed follow a low-rank structure, yet FCMTL achieves significantly better performances than it on the second and third data sets. One possible reason is that the last four dimensions for all tasks make the low-rank assumption in these two data sets invalid. Feature sharing has been restricted in each disjoint group, yet the performances of Group MTFL are still clearly worse than CMTL or FCMTL. This is largely because the regularization used in Group MTFL is the square of trace norm instead of the trace norm as used in Robust MTL, where the latter is more powerful on pursing a low-rank solution due to the regularization of ℓ_1 norm of the singular values. Since tasks are from various clusters, there does not exist a shared part for all tasks (data set 1) or the shared part is not extensive enough (data sets 2 and 3), the improvement of FlexTClus is not significant. Although MTRL attempts to learn pairwise task relationships, the positive semi-definite constraint on the task relationship matrix may not be strong enough. When four clusters are exactly orthogonal, the performance of CMTL is comparable to FCMTL, otherwise, FCMTL clearly outperforms CMTL, possibly due to the spectral relaxation used in CMTL.

Fig. 1 shows the correlation matrices of the learned model parameters on synthetic data set 1. From the figure, we observe that the proposed FCMTL learns the exact underlying cluster structure. In comparison, although CMTL also obtains a good quantitative result, it introduces some noise to the correlation matrix which is possibly

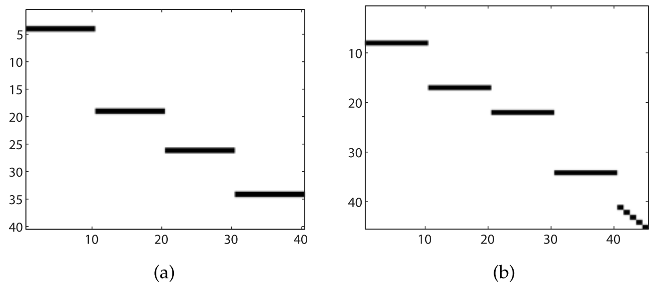


Fig. 2. The representative tasks and the corresponding assignment matrix $\mathbf{Z}$ obtained by the proposed method on the synthetic data set 2 and 3. Darker color indicates larger value.

attributed to the spectral relaxation used in CMTL. Similar observations can be made in Robust MTL (Fig. 1e), where certain incorrect correlations have been introduced between irrelevant tasks due to the noise in the structure of all tasks which is not exactly low-rank. As shown in Fig. 1f, the task relationships in each group learned by Group MTFL are not close to the ground truth, which is still largely due to the use of the square of trace norm in regularization. Due to the assumption that one part is shared by all tasks in FlexTClus, there are considerable noises between two tasks from two different clusters. Of course, the noise can be decreased by using a smaller regularization parameter for the shared part. We find, however, that the current regularization parameter gives better performance, which is possibly because smaller regularization parameter also less enforces the sharing between relevant tasks. Fig. 1h shows that MTRL learns well the relationships of tasks from the same cluster, yet unwanted correlations exist between tasks from different clusters whose model parameters are orthogonal and uncorrelated. This is probably due to that MTRL only imposes the positive semi-definite constraint on the task relationship matrix, which is ineffective for unrelated tasks. Other MTL methods and the STL baseline fail to obtain good results even for the relevant tasks, and introduce considerable noise for unrelated tasks due to the invalid assumptions on task relationships.

Fig. 2 shows the representative tasks and the assignment matrix $\mathbf{Z}$ obtained by the proposed FCMTL. It can be seen from the figure that the proposed FCMTL can effectively capture the underlying cluster structure even though not all tasks are orthogonal and in cases of outlier tasks. In the synthetic data set 2, all tasks within a particular cluster are assigned to the same representative task from their cluster. In data set 3, each outlier task is selected as a representative task only by itself.

4.2 Examination Score Prediction

In this section, we evaluate the algorithms on the School data set [2] which has been widely used in multi-task learning research. The data set contains the examination scores of 15,362 students from 139 secondary schools and each school has been considered as one task. The problem is to predict the scores for students according to their input attributes. The same preprocessing as [2] is used in our experiments. We run the experiments under five different settings: 10, 20, 30, 40 and 50 percent of the data are used as training data. Similar to [13], [60], we use 20 percent of the

TABLE 2
Mean and Standard Deviation of NMSE of All Methods on the School Data Set

	10%	20%	30%	40%	50%
STL	1.083 ± 0.017	0.953 ± 0.012	0.894 ± 0.010	0.855 ± 0.013	0.840 ± 0.010
Regularized MTL	0.815 ± 0.012	0.770 ± 0.011	0.773 ± 0.005	0.770 ± 0.011	0.767 ± 0.012
Dirty MTL	1.016 ± 0.025	0.885 ± 0.017	0.843 ± 0.013	0.814 ± 0.012	0.807 ± 0.011
Robust MTL	0.993 ± 0.024	0.863 ± 0.014	0.819 ± 0.010	0.792 ± 0.014	0.787 ± 0.013
Group MTFL	0.953 ± 0.023	0.830 ± 0.015	0.795 ± 0.010	0.773 ± 0.013	0.755 ± 0.010
FlexTClus	0.816 ± 0.010	0.783 ± 0.010	0.776 ± 0.008	0.766 ± 0.010	0.752 ± 0.012
MTRL	0.991 ± 0.027	0.852 ± 0.014	0.806 ± 0.010	0.784 ± 0.015	0.774 ± 0.013
CMTL	0.831 ± 0.011	0.806 ± 0.008	0.795 ± 0.004	0.772 ± 0.012	0.770 ± 0.008
FCMTL	0.813 ± 0.013	0.770 ± 0.010	0.763 ± 0.006	0.758 ± 0.012	0.759 ± 0.012

data as a validation set to tune the regularization parameters for all methods. The rest of the data are used for testing. For each setting, we repeat the experiments 10 times by randomly splitting the data.

Table 2 reports the mean and standard derivation over 10 trials for five different split schemes. As can be seen, all multi-task learning methods achieve better results than the single-task learning method. FCMTL outperforms the other methods or achieves comparable results under all settings, which clearly demonstrates its effectiveness. Furthermore, we note that the performance of Regularized MTL is also better than other MTL methods. This is because in this scenario, all tasks are indeed closely relevant which has been mentioned in previous works [4], [20]. Similarly, FlexTClus can perform well by using a large regularization parameter for the shared part. Fig. 3 shows the representative tasks and the assignment matrix $\mathbf{Z}$ obtained by the proposed FCMTL. In both settings, two tasks are selected as representative tasks and the probabilities that they are assigned to each task are slightly different (the reader please zoom-in the figure for the best viewing effect).

4.3 MHC-I Binding Data Set

Next, we experiment on the MHC-I binding data set which has been used in [26]. The data set contains binding affinities of various peptides with different MHC-I molecules. In this experiment, each task is a binary classification problem. Following the protocol used in [26], we conduct experiments on the same 10 tasks where each has less than 200 examples. In total, we have 1,200 examples for all the 10 tasks and the feature dimension is 180. We run the experiments by using 20 and 40 percent of the data for training. We use 20 percent of the data as a validation set and the

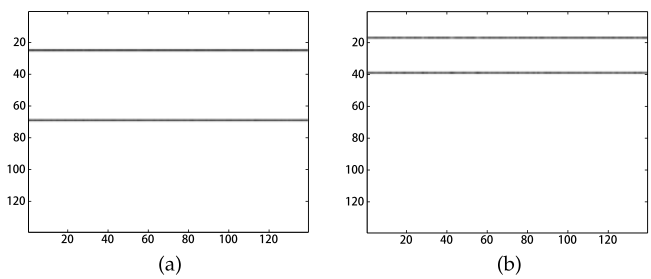


Fig. 3. The representative tasks and the corresponding assignment matrix $\mathbf{Z}$ obtained by the proposed method on the School data set by using 10 and 30 percent of the data as training data. Darker color indicates larger value. Please zoom-in the image for the best visual results.

regularization parameters for all methods are tuned on it. The remaining data are used for testing. To evaluate the performance, we report the mean average precision (mean AP) on all 10 tasks. For each setting, we repeat the experiments five times by randomly splitting data.

Table 3 shows the mean and standard derivation over five trials on both settings. We observe that the performance of the Regularized MTL and Dirty MTL are worse than the single-task learning, since these two methods assume all tasks to be relevant which is not valid in this data set. Although Robust MTL explicitly models the outlier tasks in its formulation, it also fails to achieve good performance, possibly due to low-rank structure of all tasks on this data set is not quite obvious. The performance of Group MTFL is better than Robust MTL as tasks have been clustered into groups, which leads to a more reasonable assumption than for Robust MTL. In comparison, FCMTL, CMTL, FlexTClus, and MTRL perform better than STL, since all these methods attempt to learn the underlying task structure or relationships from training data. The proposed FCMTL outperforms all other methods on both settings.

Fig. 4 shows the representative tasks and the assignment matrix $\mathbf{Z}$ obtained by the proposed FCMTL. As shown in Fig. 4, some tasks in this data set are not related to others and they are selected as representative tasks only by themselves. Other tasks select multiple representative tasks with different probabilities, making the sharing across tasks more flexible. As shown in Fig. 4, the proposed FCMLT is still able to achieve performance gains even though the number of representative tasks is equivalent to the number of total tasks. Intuitively, the strength of sharing of the

TABLE 3
Mean Average Precision ($\%$) for the 10 Molecules with Less Than 200 Training Samples Each in the MHC-I Data Set

	20%	40%
STL	74.4 ± 2.0	79.9 ± 2.8
Regularized MTL	73.8 ± 2.2	79.7 ± 3.3
Dirty MTL	73.0 ± 2.1	79.7 ± 3.9
Robust MTL	72.2 ± 1.2	79.3 ± 2.8
Group MTFL	74.7 ± 1.5	79.5 ± 2.9
FlexTClus	75.3 ± 1.6	80.5 ± 2.4
MTRL	74.5 ± 1.8	81.3 ± 3.2
CMTL	75.1 ± 1.2	81.1 ± 2.4
FCMTL	76.6 ± 1.7	81.9 ± 2.2

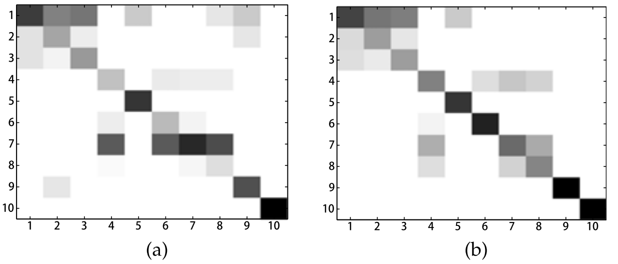


Fig. 4. The representative tasks and the corresponding assignment matrix $\mathbf{Z}$ obtained by the proposed method on the MHC-I data set by using 20 and 40 percent of the data as training data. Darker color indicates larger value.

proposed FCMTL is more significant if less tasks are selected as representative tasks. We would like to, however, clarify that according to the third term $(\mathbf{Z}_{ik}\|\mathbf{w}_i - \mathbf{w}_k\|_2^2)$ in (4), sharing always holds between the ith and kth tasks as long as $\mathbf{Z}_{ik}$ is nonzero, because this enforces $\mathbf{w}_i$ and $\mathbf{w}_k$ to be similar to each other. As shown in Fig. 4, sharing is mainly observed within two clusters: tasks 1, 2 ,3 can be considered as to form one cluster and task 4, 6, 7, 8 to form another cluster. In addition, in both 20 and 40 percent cases, task 5 also selects task 1 as its representative tasks with a weight of about 0.21. The flexible clustering makes it possible for task 5 to share with task 1, while at the same time not sharing with tasks 2 and 3. Note that in CMTL, tasks can only be shared within each disjoint cluster, and with a same fixed weight for each pair of tasks in the cluster. Differently, however, in the proposed FCMTL, tasks can be flexibly shared beyond disjoint clusters; further even in one cluster, the sharing strength for each pair of task is determined by a learned weight ($\mathbf{Z}_{ik}$) instead of a same fixed one for all pairs.

4.4 Fine Grained Visual Recognition

Finally, we report evaluation results for fine grained visual recognition. Different from the traditional visual recognition, fine grained visual recognition aims at solving subordinate category classification (all categories are divided into families where each family consists of fine grained categories). In our experiments, we consider the fine grained bird classification on the Caltech-UCSD (CUB) Birds data set [48], [50], which contains 200 categories and each category includes about 30 and 60 images in versions 2010 and 2011, respectively.

Specifically, we run experiments on six families (*Flycatcher, Gull, Term, Vireo, Woodpecker, and Wren*) which contain 42 bird categories in total. We extract the HOG feature [15] and use LLC [49] to represent the low-level descriptors with 1,024 visual words. PCA has been applied for dimension reduction with 40 percent energy preserved. Table 4 summarizes the experimental setting on this data set. About 15 training images for each category on CUB2010 and around 30 training images for each category on CUB2011 are used. As the training data per category are scare, we validate the parameters on the test data and report the best results of each method.

TABLE 4
Summarization of 42 Categories (Six Families) of the Caltech-UCSD Birds Data Set Used in Our Experiments

	# Dim	# Training	# Testing
CUB2010	94	630	656
CUB2011	145	1,257	1,223

TABLE 5
Mean AP (%) of All Methods on the Caltech-UCSD Birds Data Set by Running Experiments on 42 Categories (Six Families)

	CUB2010	CUB2011
STL	14.57	22.64
Regularized MTL	12.46	22.31
Dirty MTL	14.06	21.46
Robust MTL	13.62	21.77
Group MTFL	15.14	22.93
FlexTClus	14.84	23.22
MTRL	15.39	23.54
CMTL	15.27	23.87
FCMTL	16.44	24.07

Table 5 reports the results on this data set. We again use the mean average precision (mean AP) of all categories to evaluate the performance. Our approach achieves the best performance on both CUB2010 and CUB2011, especially when only few training images are available for each category. We also note that the results of Regularized MTL, Dirty MTL and Robust MTL are worse than STL. This can be attributed to the invalid assumption which fails to capture the correct task relationship.

4.5 Convergence Analysis

Since the overall model in (4) is non-convex with both subproblems (5) and (6) convex, Algorithm 1 converges to local optimum if both subproblems converge to their global optimum. For the first subproblem, with smooth convex objective function in (5), APG converges to the global optimum. For the second subproblem, the convergence property of ADMM for convex objective function with more than two block variables cannot be theoretically guaranteed as is generally accepted currently [22], [8]. Therefore, the theoretical proof of the convergence of ADMM for (6) is not straightforward since there are three block variables in the ADMM procedure for the second subproblem (6). Consequently, the convergence of FCMTL cannot be shown by theoretically analysis. Empirically, we observe from experiments on both synthetic and real data sets that the Algorithm 1 performs well in terms of convergence. In Fig. 5, we experimentally demonstrate the convergence of FCMTL and show typical examples on synthetic, School and MHC-I data sets. As shown in Fig. 5, the objective values of FCMTL (4) usually converge in less than five iterations. We have similar observations on other data sets.

4.6 Computational Complexity

In the block coordinate descent procedure, the FCMTL problem (4) is solved by iteratively solving (5) and (11). We focus on discussing the computational complexity of the main components involved in each iteration of these two subproblems. As shown in Fig. 5, the block coordinate descent procedure usually converges after less than five iterations.

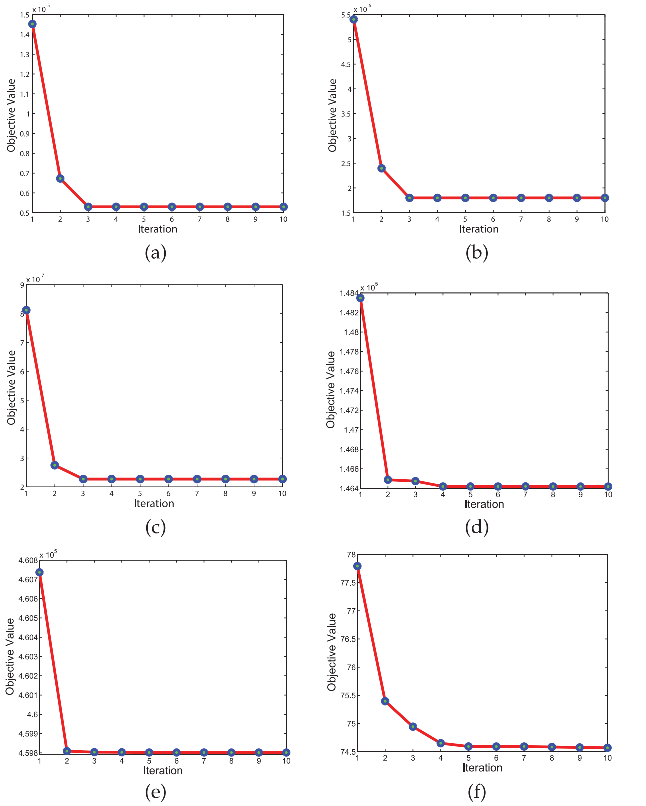


Fig. 5. Illustration of the convergence of FCMTL. (a) Synthetic data set 1. (b) Synthetic data set 2. (c) Synthetic data set 3. (d) School data set with 10 percent data . (e) School data set with 30 percent data. (f) MHC-I data set with 20 percent data.

For ease of analysis, we assume that the number of training samples for each task is n. For the subproblem (5), the main computational cost comes from computing the gradient of **W**. Specifically, the computational cost for the three terms in (5) are $\mathcal{O}(mdn)$, $\mathcal{O}(md)$ and $\mathcal{O}(m^2d)$, respectively. Therefore, the overall computational complexity of the subproblem (5) is $\mathcal{O}(mdn + m^2d)$, which grows quadratically with the task number, and linearly with both the feature dimensionality and the number of training data. For the subproblem (11), the computational complexity is $\mathcal{O}(m^3)$, which grows cubically with the task number.

In addition, we also compare the proposed method with previous methods on computational running time. All the computational running times are assessed on a PC with 3.40 GHz Intel(R) Core(TM) i7-3770 CPU and 32 GB memory. The results on the three synthetic data sets are shown in Table 6. Notice that, for Group MTFL, following their paper, the gradient descent has been conducted 10 times for obtaining better performance. In the comparison shown in Table 6, we only count the mean computational time of the 10 gradient descent processes, which means the true running time of Group MTFL is much longer than the time we reported. All the codes are written in MATLAB except some parts of Dirty MTL and FlexTClus. The time consuming part in Dirty MTL and FlexTClus has been speeded up by using C-MEX programming; otherwise, the computational time will be much longer for them. For Dirty MTL, we find that the speed of C-MEX code is about five times faster than its MATLAB counterpart. In general, the computational time of the proposed FCMTL has the same scale in running time as Dirty MTL, MTRL and FlexTClus. We observe that STL, Regularized MTL and Robust MTL perform considerably faster as the models are relatively straightforward, while Group MTFL and CMTL are clearly slower than STL and other MTL methods.

TABLE 6
Computational Running Time (in Seconds)

	Synthetic Data		
	Set 1	Set 2	Set 3
STL	0.30	0.28	0.34
Regularized MTL	0.41	0.44	0.47
Dirty MTL	2.17	2.23	2.37
Robust MTL	0.63	0.79	0.93
Group MTFL	32.71	31.45	41.59
FlexTClus	2.82	3.09	2.90
MTRL	2.07	2.21	2.87
CMTL	23.24	61.56	59.02
FCMTL	6.08	6.94	8.00

5 Conclusions and Future Work

This paper proposes a new approach called Flexible Clustered Multi-Task Learning for multi-task learning. The proposed FCMTL learns the underlying cluster structure among tasks by identifying representative tasks, and all tasks are clustered into groups according to the shared representative tasks. The new approach does not require all clusters to be disjoint or all tasks within the same cluster to share information to the same extent, thus more flexible in characterizing an arbitrary task and capturing the underlying clustering structure in terms of information sharing. Promising results on both synthetic and real-world data sets demonstrate the effectiveness of the proposed method. For future work, we plan to investigate on a convex formulation for FCMTL.

Appendix A
Proof of Proposition 1

Proof. It is easy to verify that the first two terms in the objective function of problem (5) are convex respect to **W** and **b**. For the third term, (a) the jth dimension in $\|w_i - w_k\|_2^2$ is $(w_{ij} - w_{kj})^2$, whose convexity can be proved by verifying its Hessian to be positive semi-definite. (b) Since $\mathbf{Z}_{ik}$ is nonnegative, thus this third term is also convex as a nonnegative weighted sum of convex functions is convex [9]. Same as (b), (5) is convex as it is the sum of three convex terms. □

Appendix B
Proof of Theorem 1

Proof. To start with, we convert the problem in (6) to the following equivalent problem

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \mathrm{Tr}\big(\mathbf{D}^T\mathbf{Z}\big) + \beta\|\mathbf{Z}\|_{1,2} \\ s.t. \quad & \mathbf{0} \preceq \mathrm{vec}(\mathbf{Z}),\ \mathbf{Z}^T\mathbf{1}_m = \mathbf{1}_m, \end{aligned} \tag{12}$$

where $\beta = \lambda/\mu$.

It is easy to verify that (12) satisfies the Slaters's condition, thus strong duality holds. The Lagrangian of (12) is

$$\begin{aligned}\mathcal{L}(\mathbf{Z},\mathbf{A},\mathbf{B}) &= \sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{D}_{ij}\mathbf{Z}_{ij} + \beta\sum_{i=1}^{m}\|\mathbf{Z}_i\|_2 \\ &- \sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{A}_{ij}\mathbf{Z}_{ij} + \sum_{j=1}^{m}\mathbf{B}_j\left(\sum_{i=1}^{m}\mathbf{Z}_{ij}-1\right),\end{aligned} \tag{13}$$

where $\mathbf{A}\in\mathbb{R}^{m\times m}$ and $\mathbf{B}\in\mathbb{R}^{m}$ are the Lagrange multipliers associated with the inequality and equality constraints, respectively.

Then, the *Karush-Kuhn-Tucker* (KKT) condition of (13) is

$$\mathbf{A}_{ij} \geq 0, \tag{14}$$

$$\mathbf{A}_{ij}\mathbf{Z}_{ij} = 0, \tag{15}$$

$$\frac{\partial\mathcal{L}(\mathbf{Z},\mathbf{A},\mathbf{B})}{\partial\mathbf{Z}_i} = \mathbf{D}_i + \beta\partial\|\mathbf{Z}_i\|_2 - \mathbf{A}_i + \mathbf{B}^T \ni \mathbf{0}^T, \tag{16}$$

where $\partial\|\mathbf{Z}_i\|_2$ is the subgradient of $\|\mathbf{Z}_i\|_2$ with respect to $\mathbf{Z}_i$ and it is defined as [46]

$$\partial\|\mathbf{Z}_i\|_2 = \begin{cases}\frac{\mathbf{Z}_i}{\|\mathbf{Z}_i\|_2} & \text{if } \mathbf{Z}_i \neq \mathbf{0}^T \\ \left\{\mathbf{\Omega}_i \,|\, \|\mathbf{\Omega}_i\|_2 \leq 1\right\} & \text{if } \mathbf{Z}_i = \mathbf{0}^T,\end{cases} \tag{17}$$

and $\mathbf{\Omega}\in\mathbb{R}^{m\times m}$.

We first prove the requirement of β for each task select itself as the only representative task, i.e., $(\mathbf{Z}=\mathbf{I})$.

Since $\mathbf{Z}_i \neq \mathbf{0}$, the gradient of $\|\mathbf{Z}_i\|_2$ with respect to $\mathbf{Z}_i$ exists. For each $\mathbf{Z}_{ij}$, the condition in (16) can be written as

$$\frac{\partial\mathcal{L}(\mathbf{Z},\mathbf{A},\mathbf{B})}{\partial\mathbf{Z}_{ij}} = \mathbf{D}_{ij} + \frac{\mathbf{Z}_{ij}}{\|\mathbf{Z}_i\|_2}\beta - \mathbf{A}_{ij} + \mathbf{B}_j = 0. \tag{18}$$

Since $\mathbf{Z}_{ij}=0$ for $j\neq i$, thus we have

$$\mathbf{B}_j = \mathbf{A}_{ij} - \mathbf{D}_{ij}. \tag{19}$$

Apply (18) on the $\mathbf{Z}_{jj}$, we get

$$\beta = \mathbf{A}_{jj} - \mathbf{D}_{jj} - \mathbf{B}_j. \tag{20}$$

According to (15), we have $\mathbf{A}_{jj}=0$ due to $\mathbf{Z}_{jj}=1$. Therefore,

$$\beta = -\mathbf{D}_{jj} - \mathbf{B}_j. \tag{21}$$

Substitute (19) for $\mathbf{B}_j$ in (21), we get

$$\beta = -\mathbf{D}_{jj} - \mathbf{A}_{ij} + \mathbf{D}_{ij}. \tag{22}$$

By (14), we have

$$\beta \leq \mathbf{D}_{ij} - \mathbf{D}_{jj}. \tag{23}$$

Consider (23) for all $i\neq j$ together, we get

$$\beta \leq \min_{i\neq j}\mathbf{D}_{ij} - \mathbf{D}_{jj}. \tag{24}$$

Since all columns of $\mathbf{Z}$ should satisfy (24), thus we obtain

$$\beta_{\min} = \min_j\left(\min_{i\neq j}\mathbf{D}_{ij} - \mathbf{D}_{jj}\right). \tag{25}$$

Next, we prove the requirement of β for only one representative is selected for all tasks, i.e., $(\mathbf{Z}=e_k\mathbf{1}_m^T)$:

It is easy to verify if all tasks select only one common representative task, then the representative task is the kth task that satisfies

$$k = \arg\min_i \mathbf{D}_i\mathbf{1}_m. \tag{26}$$

As the constraint of $\mathbf{Z}^T\mathbf{1}_m=\mathbf{1}_m$, so each $\mathbf{Z}_{kj}$ can be represented as $\mathbf{Z}_{kj}=1-\sum_{i\neq k}\mathbf{Z}_{ij}$. Based on this, the objective function in (12) can be written as

$$\begin{aligned}&\sum_{i\neq k}\left\{\mathbf{D}_i\mathbf{Z}_i^T + \beta\|\mathbf{Z}_i\|_2\right\} + \sum_{j=1}^{m}\mathbf{D}_{kj}\Big(1-\sum_{i\neq k}\mathbf{Z}_{ij}\Big) \\ &+ \beta\sqrt{\sum_{j=1}^{m}\Big(1-\sum_{i\neq k}\mathbf{Z}_{ij}\Big)^2}.\end{aligned} \tag{27}$$

The optimality condition for the kth task be the only representative task is

$$\mathbf{D}_i + \beta\partial\|\mathbf{Z}_i\|_2 - \mathbf{D}_k - \beta\frac{\mathbf{1}_m^T}{\sqrt{m}} \ni \mathbf{0}^T, \forall i\neq k, \tag{28}$$

which implies

$$\left(\frac{\mathbf{D}_k-\mathbf{D}_i}{\beta} - \frac{\mathbf{1}_m^T}{\sqrt{m}}\right) \in \partial\|\mathbf{Z}_i\|_2. \tag{29}$$

According to the definition of subgradient in (17), we have

$$\left\|\frac{\mathbf{D}_k-\mathbf{D}_i}{\beta} - \frac{\mathbf{1}_m^T}{\sqrt{m}}\right\|_2 \leq 1, \tag{30}$$

which implies

$$\beta \geq \frac{\sqrt{m}}{2}\frac{\|\mathbf{D}_i-\mathbf{D}_k\|_2^2}{(\mathbf{D}_i-\mathbf{D}_k)\mathbf{1}_m}. \tag{31}$$

Since for all $i\neq k$ should satisfy (31), we get

$$\beta \geq \beta_{\max} = \max_{i\neq k}\frac{\sqrt{m}}{2}\frac{\|\mathbf{D}_i-\mathbf{D}_k\|_2^2}{(\mathbf{D}_i-\mathbf{D}_k)\mathbf{1}_m}. \tag{32}$$

This ends of the proof of the theorem. □

APPENDIX C
DETAILS OF THE ADMM PROCEDURE FOR (11)

The ADMM procedure for (11) consists of iteratively applying the following update equations

$$\begin{aligned}&(a)\mathbf{Z}^{k+1} \leftarrow \arg\min_{\mathbf{Z}} L_\rho\big(\mathbf{Z},\mathbf{P}^k,\mathbf{Q}^k,\mathbf{C}_1^k,\mathbf{C}_2^k,\mathbf{C}_3^k\big)\\ &(b)\mathbf{P}^{k+1} \leftarrow \arg\min_{\mathbf{P}} L_\rho\big(\mathbf{Z}^{k+1},\mathbf{P},\mathbf{Q}^k,\mathbf{C}_1^k,\mathbf{C}_2^k,\mathbf{C}_3^k\big)\\ &(c)\mathbf{Q}^{k+1} \leftarrow \arg\min_{\mathbf{Q}} L_\rho\big(\mathbf{Z}^{k+1},\mathbf{P}^{k+1},\mathbf{Q},\mathbf{C}_1^k,\mathbf{C}_2^k,\mathbf{C}_3^k\big)\\ &(d)\mathbf{C}_1^{k+1} \leftarrow \mathbf{C}_1^k + \rho\big(\mathbf{Z}^{k+1}-\mathbf{P}^{k+1}\big)\\ &\mathbf{C}_2^{k+1} \leftarrow \mathbf{C}_2^k + \rho\big(\mathbf{P}^{k+1}-\mathbf{Q}^{k+1}\big)\\ &\mathbf{C}_3^{k+1} \leftarrow \mathbf{C}_3^k + \rho\Big({\mathbf{Z}^{k+1}}^T\mathbf{1}_m-\mathbf{1}_m\Big).\end{aligned}$$

Next, we describe each of these steps in turn.

Minimizing Over Z

The subproblem in Step (a) is

$$\min_{\mathbf{Z}} \lambda \operatorname{tr}(\mathbf{D}^T\mathbf{Z}) + g(\mathbf{P}^k) + \langle \mathbf{C}_1^k, \mathbf{Z} - \mathbf{P}^k \rangle + \langle \mathbf{C}_3^k, \mathbf{Z}^T\mathbf{1}_m - \mathbf{1}_m \rangle + \frac{\rho}{2}\left(\|\mathbf{Z} - \mathbf{P}^k\|_F^2 + \|\mathbf{Z}^T\mathbf{1}_m - \mathbf{1}_m\|_2^2 \right). \tag{33}$$

This problem has the closed-form solution

$$\mathbf{Z}^{k+1} = \left(\mathbf{I}_{m\times m} + \mathbf{1}_m\mathbf{1}_m^T\right)^{-1}\left(\mathbf{P}^k + \mathbf{1}_m\mathbf{1}_m^T - \frac{\lambda}{\rho}\mathbf{D} - \frac{1}{\rho}\mathbf{C}_1^k - \frac{1}{\rho}\mathbf{1}_m\mathbf{C}_3^{kT}\right). \tag{34}$$

Minimizing Over P

In Step (b), we need to solve the following problem

$$\min_{\mathbf{P}} g(\mathbf{P}) + \langle \mathbf{C}_1^k, \mathbf{Z}^{k+1} - \mathbf{P} \rangle + \langle \mathbf{C}_2^k, \mathbf{P} - \mathbf{Q}^k \rangle + \frac{\rho}{2}\left(\|\mathbf{Z}^{k+1} - \mathbf{P}\|_F^2 + \|\mathbf{P} - \mathbf{Q}^k\|_F^2 \right). \tag{35}$$

This problem also has the closed-form solution

$$\mathbf{P}^{k+1} = \Pi_{\mathcal{C}}\left(\frac{1}{2}\left(\mathbf{Z}^{k+1} + \mathbf{Q}^k\right) + \frac{1}{2\rho}\left(\mathbf{C}_1^k - \mathbf{C}_2^k\right) \right), \tag{36}$$

where $\Pi_{\mathcal{C}}$ denotes the Euclidean projection onto the set $\mathcal{C}$.

Minimizing Over Q

The problem need to be solved in Step (c) is

$$\min_{\mathbf{Q}} \mu\|\mathbf{Q}\|_{1,2} + \langle \mathbf{C}_2^k, \mathbf{P}^{k+1} - \mathbf{Q} \rangle + \frac{\rho}{2}\|\mathbf{P}^{k+1} - \mathbf{Q}\|_F, \tag{37}$$

which is equivalent to the following problem

$$\min_{\mathbf{Q}} \frac{1}{2}\left\| \mathbf{Q} - \left(\mathbf{P}^{k+1} + \frac{1}{\rho}\mathbf{C}_2^k \right) \right\|_F^2 + \frac{\mu}{\rho}\|\mathbf{Q}\|_{1,2}, \tag{38}$$

and the closed-form solution for it can be obtained by applying the proximity operator on each row of $\mathbf{Q}$ separately. For the ith row, the solution is

$$\mathbf{R} = \mathbf{P}^{k+1} + \frac{1}{\rho}\mathbf{C}_2^k \qquad \mathbf{Q}^{k+1}(i,:) = \left[\frac{\|\mathbf{R}(i,:)\|_2 - \frac{\mu}{\rho}}{\|\mathbf{R}(i,:)\|_2} \right]_+ \mathbf{R}(i,:). \tag{39}$$

Appendix D
Kernel Extension

Several previous works also studied the non-linear extension of MTL methods [3], [16], [20], [30], [45], [55], [57]. Here, we show that the proposed FCMTL (4) can be easily extended to nonlinear kernel functions. In the following, we demonstrate it with the nonlinear regression problem as an example, yet it can be generalized to other forms.

Formally, for the ith task, the goal is to learn a regression function $f_i(x_j^i) = w_i^T\phi(x_j^i) + b_i$ where $\phi(x_j^i)$ denotes the nonlinear feature map by a reproducing kernel. Then, the optimization problem for learning $\mathbf{W}$ and $\mathbf{b}$ with fixed $\mathbf{Z}$ is

$$\min_{\mathbf{W},\mathbf{b}} \sum_{i=1}^m \frac{1}{n_i}\sum_{j=1}^{n_i}\left(y_j^i - w_i^T\phi(x_j^i) - b_i \right)^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \frac{\lambda}{2}\sum_{i=1}^m\sum_{k=1}^m \mathbf{Z}_{ik}\|w_i - w_k\|_2^2, \tag{40}$$

which can be written as the following equivalent problem

$$\min_{\mathbf{W},\mathbf{b}} \sum_{i=1}^m \frac{1}{n_i}\sum_{j=1}^{n_i}\left(\xi_j^i\right)^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \frac{\lambda}{2}\sum_{i=1}^m\sum_{k=1}^m \mathbf{Z}_{ik}\|w_i - w_k\|_2^2 \quad s.t.\ y_j^i - \left(w_i^T\phi(x_j^i) + b_i\right) = \xi_j^i, \forall i, j. \tag{41}$$

The Lagrangian of problem (41) can be written as

$$L = \sum_{i=1}^m \frac{1}{n_i}\sum_{j=1}^{n_i}\left(\xi_j^i\right)^2 + \frac{\gamma}{2}\|\mathbf{W}\|_F^2 + \frac{\lambda}{2}\sum_{i=1}^m\sum_{k=1}^m \mathbf{Z}_{ik}\|w_i - w_k\|_2^2 + \sum_{i=1}^m\sum_{j=1}^{n_i}\alpha_j^i\left(y_j^i - \left(w_i^T\phi(x_j^i) + b_i\right) - \xi_j^i \right), \tag{42}$$

where α_j^i is the Lagrange multiplier associated with the jth training sample of the ith task. Setting the derivative of L with respect to w_i equal to zero, we obtain

$$\frac{\partial L}{\partial w_i} = \gamma w_i + \lambda\sum_{k\neq i}\mathbf{Z}_{ik}(w_i - w_k) - \lambda\sum_{k\neq i}\mathbf{Z}_{ki}(w_k - w_i) - \sum_{j=1}^{n_i}\alpha_j^i\phi(x_j^i) = 0.$$

Combining the above equation for all w_i, we have

$$\mathbf{WS} = \sum_{i=1}^m\sum_{j=1}^{n_i}\alpha_j^i\phi(x_j^i)\mathbf{e}_i^T, \tag{43}$$

where $\mathbf{e}_i \in \mathbb{R}^m$ is the ith column vector of $\mathbf{I}_{m\times m}$. $\mathbf{S} \in \mathbb{R}^{m\times m}$ and its element is defined by

$$\mathbf{S}_{ii} = \gamma + \lambda\sum_{k\neq i}(\mathbf{Z}_{ik} + \mathbf{Z}_{ki}) \text{ and } \mathbf{S}_{ki} = -\lambda(\mathbf{Z}_{ik} + \mathbf{Z}_{ki}).$$

It is easy to verify that the matrix $\mathbf{S}$ is positive definite for any $\gamma > 0$. Since $\mathbf{S}$ is positive definite, $\mathbf{W}$ in (43) is

$$\mathbf{W} = \sum_{i=1}^m\sum_{j=1}^{n_i}\alpha_j^i\phi(x_j^i)\mathbf{e}_i^T\mathbf{S}^{-1}.$$

Similarly, setting the derivatives of L with respect to b_i and ξ_j^i, we obtain

$$\frac{\partial L}{\partial b_i} = -\sum_{j=1}^{n_i}\alpha_j^i = 0$$

$$\frac{\partial L}{\partial \xi_j^i} = \frac{2}{n_i}\xi_j^i - \alpha_j^i = 0 \Rightarrow \xi_j^i = \frac{n_i}{2}\alpha_j^i.$$

Substituting $\mathbf{W}$, ξ_j^i back into (42), we obtain the following dual form of problem (41):

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \sum_{j=1}^{n_i} \alpha_j^i y_j^i - \frac{1}{2}\boldsymbol{\alpha}^T \left(\mathbf{K} + \frac{1}{2}\mathbf{V}\right)\boldsymbol{\alpha}$$
$$s.t. \sum_{j=1}^{n_i} \alpha_j^i = 0, \forall i, \tag{44}$$

where $\boldsymbol{\alpha} = (\alpha_1^1, \ldots, \alpha_{n_m}^m)^T$. $\mathbf{K} \in \mathbb{R}^{\sum_{i=1}^m n_i \times \sum_{i=1}^m n_i}$ is the multi-task kernel matrix defined on all training data of all tasks. For any two training samples $(x_{j_1}^{i_1}, x_{j_2}^{i_2})$, we define the corresponding multi-task kernel to be $\mathbf{e}_{i_1}^T \mathbf{S}^{-1} \mathbf{e}_{i_2} \kappa(x_{j_1}^{i_1}, x_{j_2}^{i_2})$ where $\kappa(x_{j_1}^{i_1}, x_{j_2}^{i_2})$ is the kernel function defined by $\kappa(x_{j_1}^{i_1}, x_{j_2}^{i_2}) = \phi(x_{j_1}^{i_1})^T \phi(x_{j_2}^{i_2})$. $\mathbf{V} \in \mathbb{R}^{\sum_{i=1}^m n_i \times \sum_{i=1}^m n_i}$ is a diagonal matrix with diagonal element n_i if the corresponding data point is from the ith task. Similar to SVM, (44) can be solved by using the SMO algorithm [32].

After solving (44), it is straightforward to update $\mathbf{Z}$ in (6) with fixed $\mathbf{W}$ and $\mathbf{b}$. Specifically, $\mathbf{D}_{ik}$ can be calculated as following

$$\mathbf{D}_{ik} = \|w_i - w_k\|_2^2 = \mathbf{e}_i^T \mathbf{H} \mathbf{e}_i - 2\mathbf{e}_i^T \mathbf{H} \mathbf{e}_k + \mathbf{e}_k^T \mathbf{H} \mathbf{e}_k,$$

where

$$\mathbf{H} = \sum_{i_1=1}^{m} \sum_{i_2=1}^{m} \sum_{j_1=1}^{n_{i_1}} \sum_{j_2=1}^{n_{i_2}} \alpha_{j_1}^{i_1} \alpha_{j_2}^{i_2} \mathbf{S}^{-1} \mathbf{e}_{i_1} \mathbf{e}_{i_2}^T \mathbf{S}^{-1} \kappa(x_{j_1}^{i_1}, x_{j_2}^{i_2}).$$

After substituting $\mathbf{D}_{ik}$ back into (6), it can be solved by ADMM as in the linear case.

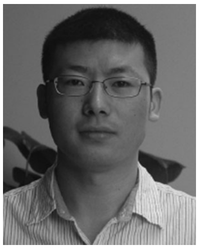
ACKNOWLEDGMENTS

The authors would like to thank Dr. Leon Wenliang Zhong for providing the implementation of [60]. They also thank the anonymous reviewers and associate editor for their constructive suggestions. The research was supported by the Defense Innovative Research Programme (No. 9014100596). Q. Zhao is the corresponding author.

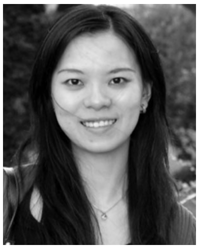
REFERENCES

[1] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, 2005.
[2] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 41–48.
[3] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3. pp. 243–272, 2008.
[4] B. Bakker and T. Heskes, "Task clustering and gating for Bayesian multitask learning," *J. Mach. Learn. Res.*, vol. 4, pp. 83–99, 2003.
[5] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1. pp. 183–202, 2009.
[6] E. V. Bonilla, F. V. Agakov, and C. K. I. Williams, "Kernel multi-task learning using task-specific features," in *Proc. 11th Int. Conf. Artif. Intell. Statist.*, 2007, pp. 43–50.
[7] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams, "Multi-task Gaussian process prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 153–160.
[8] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1. pp. 1–122, 2011.
[9] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[10] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1. pp. 41–75, 1997.
[11] J. Chen, J. Liu, and J. Ye, "Learning incoherent sparse and low-rank patterns from multiple tasks," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2010, pp. 1179–1188.
[12] J. Chen, L. Tang, J. Liu, and J. Ye, "A convex formulation for learning a shared predictive structure from multiple tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5. pp. 1025–1038, May 2013.
[13] J. Chen, J. Zhou, and J. Ye, "Integrating low-rank and group-sparse structures for robust multi-task learning," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2011, pp. 42–50.
[14] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing, "Smoothing proximal gradient method for general structured sparse learning," in *Proc. Conf. Uncertainty Artif. Intell.*, 2011, pp. 105–114.
[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 886–893.
[16] F. Dinuzzo, "Learning output kernels for multi-task problems," *Neurocomputing*, vol. 118, pp. 119–126, 2013.
[17] F. Dinuzzo, G. Pillonetto, and G. D. Nicolao, "Client-server multi-task learning from distributed datasets," *IEEE Trans. Neural Netw.*, vol. 22, no. 2. pp. 290–303, Feb. 2011.
[18] E. Elhamifar, G. Sapiro, and R. Vidal, "Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 19–27.
[19] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 1600–1607.
[20] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Mach. Learn. Res.*, vol. 6, pp. 615–637, 2005.
[21] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2004, pp. 109–117.
[22] D. Goldfarb, S. Ma, and K. Scheinberg, "Fast alternating linearization methods for minimizing the sum of two convex functions," *Math. Program.*, vol. 141, no. 1–2, pp. 349–382, 2013.
[23] P. Gong, J. Ye, and C. Zhang, "Multi-stage multi-task feature learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1997–2005.
[24] P. Gong, J. Ye, and C. Zhang, "Robust multi-task feature learning," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2012, pp. 895–903.
[25] P. Gong, J. Ye, and C. Zhang, "Multi-stage multi-task feature learning," *J. Mach. Learn. Res.*, vol. 14, pp. 2979–3010, 2013.
[26] L. Jacob, F. Bach, and J.-P. Vert, "Clustered multi-task learning: A convex formulation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 745–752.
[27] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in *Proc. 26th Anuu. Int. Conf. Mach. Learn.*, 2009, pp. 433–440.
[28] L. Jacob and J.-P. Vert, "Efficient peptide-mhc-i binding prediction for alleles with few known binders," *Bioinformatics*, vol. 24, no. 3. pp. 358–366, 2008.
[29] A. Jalali, P. D. Ravikumar, S. Sanghavi, and C. Ruan, "A dirty model for multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 964–972.
[30] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 521–528.
[31] T. Kato, H. Kashima, M. Sugiyama, and K. Asai, "Conic programming for multitask learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7. pp. 957–968, Jul. 2010.
[32] S. S. Keerthi and S. K. Shevade, "SMO algorithm for least-squares svm formulations," *Neural Comput.*, vol. 15, no. 2. pp. 487–507, 2003.
[33] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 543–550.
[34] A. Kumar and H. Daumé-III, "Learning task grouping and overlap in multi-task learning," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1383–1390.
[35] S. Lee, J. Zhu, and E. P. Xing, "Adaptive multi-task lasso: With application to eqtl detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1306–1314.
[36] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l2, 1-norm minimization," in *Proc. Conf. Uncertainty Artif. Intell.*, 2009, pp. 339–348.
[37] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, no. 1. pp. 125–161, 2013.

[38] A. Passos, P. Rai, J. Wainer, and H. Daumé-III, "Flexible modeling of latent task structures in multitask learning," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1103–1110.
[39] G. Pillonetto, F. Dinuzzo, and G. D. Nicolao, "Bayesian online multitask learning of Gaussian processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2. pp. 193–205, Feb. 2010.
[40] N. Quadrianto, A. J. Smola, T. S. Caetano, S. V. N. Vishwanathan, and J. Petterson, "Multitask learning without label correspondences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1957–1965.
[41] A. Rakotomamonjy, R. Flamary, G. Gasso, and S. Canu, "ℓ_p-ℓ_q penalty for sparse linear and sparse multiple kernel multitask learning," *IEEE Trans. Neural Netw.*, vol. 22, no. 8. pp. 1307–1320, Aug. 2011.
[42] B. Romera-Paredes, A. Argyriou, N. Berthouze, and M. Pontil, "Exploiting unrelated tasks in multi-task learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2012, pp. 951–959.
[43] B. Romera-Paredes, H. Aung, N. Bianchi-Berthouze, and M. Pontil, "Multilinear multitask learning," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1444–1452.
[44] A. Schwaighofer, V. Tresp, and K. Yu, "Learning Gaussian process kernels via hierarchical Bayes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1209–1216.
[45] M. Solnon, S. Arlot, and F. Bach, "Multi-task regression using minimal penalties," *J. Mach. Learn. Res.*, vol. 13, pp. 2773–2812, 2012.
[46] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.
[47] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5. pp. 854–869, May 2007.
[48] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Instit. of Technol., San Diego, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.
[49] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3360–3367.
[50] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Instit. of Technol., San Diego, CA, USA, Tech. Rep. CNS-TR-2010-001, 2010.
[51] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram, "Multi-task learning for classification with Dirichlet process priors," *J. Mach. Learn. Res.*, vol. 8, pp. 35–63, 2007.
[52] X. Yang, S. Kim, and E. P. Xing, "Heterogeneous multitask learning with joint sparsity constraints," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2151–2159.
[53] K. Yu, V. Tresp, and A. Schwaighofer, "Learning Gaussian processes from multiple tasks," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 1012–1019.
[54] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statist. Soc.: Series B (Statist. Methodol.)*, vol. 68, no. 1. pp. 49–67, 2006.
[55] Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *Proc. 26th Conf. Uncertainty Artif. Intell.*, 2010, pp. 733–742.
[56] Y. Zhang and D.-Y. Yeung, "Multi-task warped Gaussian process for personalized age estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 2622–2629.
[57] Y. Zhang and D.-Y. Yeung, "Learning high-order task relationships in multi-task learning," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1917–1923.
[58] Y. Zhang, D.-Y. Yeung, and Q. Xu, "Probabilistic multi-task feature selection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2559–2567.
[59] W. Zhong and J. T. Kwok, "Efficient sparse modeling with automatic feature grouping," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 9–16.
[60] W. Zhong and J. T.-Y. Kwok, "Convex multitask learning with flexible task clusters," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 49–56.
[61] J. Zhou, J. Chen, and J. Ye, "Clustered multi-task learning via alternating structure optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 702–710.
[62] J. Zhou, L. Yuan, J. Liu, and J. Ye, "A multi-task learning formulation for predicting disease progression," in *Proc. ACM SIGKDD Conf. Knowl. Discovery Data Mining*, 2011, pp. 814–822.



Qiang Zhou received the BEng degree in Computer Science from the Anhui University, Hefei, China, in 2010. Currently, he is working toward the PhD degree at the Department of Electrical and Computer Engineering, National University of Singapore. His research interests include machine learning and optimization.



Qi Zhao received the MSc and PhD degrees in computer engineering from the University of California, Santa Cruz, in 2007 and 2009, respectively. She is an assistant professor in the Electrical and Computer Engineering Department at the National University of Singapore (NUS) and the principal investigator at the Visual Information Processing Lab (http://www.ece.nus.edu.sg/stfpage/eleqiz), working on computational vision and cognitive neuroscience. She also holds an appointment in the Interactive and Digital Media Institute at NUS. Prior to joining NUS, she was a postdoctoral researcher in the Computation & Neural Systems, and Division of Biology at the California Institute of Technology from 2009 to 2011. Her main research interests include computational vision, machine learning, computational cognition, and neuroscience. She has published more than 30 journal and conference papers in top computer vision, cognitive neuroscience, and machine learning venues, and is editing a book with Springer, titled *Computational and Cognitive Neuroscience of Vision*, that provides a systematic and comprehensive overview of vision from various perspectives, ranging from neuroscience to cognition, and from computational principles to engineering developments. She is a member of the IEEE.