Metric-Guided Multi-task Learning

Jinfu Ren, Yang Liu, and Jiming Liu(✉)

Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong SAR, People's Republic of China
{jinfuren,csygliu,jiming}@comp.hkbu.edu.hk

Abstract. Multi-task learning (MTL) aims to solve multiple related learning tasks simultaneously so that the useful information in one specific task can be utilized by other tasks in order to improve the learning performance of all tasks. Many representative MTL methods have been proposed to characterize the relationship between different learning tasks. However, the existing methods have not explicitly quantified the distance or similarity of different tasks, which is actually of great importance in modeling the task relation for MTL. In this paper, we propose a novel method called Metric-guided MTL (M^2TL), which explicitly measures the task distance using a metric learning strategy. Specifically, we measure the distance between different tasks using their projection parameters and learn a distance metric accordingly, so that the similar tasks are close to each other while the uncorrelated tasks are faraway from each other, in terms of the learned distance metric. With a metric-guided regularizer incorporated in the proposed objective function, we open a new way to explore the related information among tasks. The proposed method can be efficiently solved via an alternative method. Experiments on both synthetic and real-world benchmark datasets demonstrate the superiority of the proposed method over existing MTL methods in terms of prediction accuracy.

Keywords: Multi-task learning · Task relation · Metric learning · Metric-guided multi-task learning

1 Introduction

Multi-task learning (MTL), inspired by human learning behavior and patterns of applying the knowledge and experience learned from some tasks to help learn others, solves multiple learning tasks simultaneously and improves the learning performance of all tasks by exploring the task similarities or commonalities [3,20]. MTL has attracted extensive research interest and has been widely applied to various real-world problems such as human facial recognition and pose estimation [4], traffic flow forecasting [5], climate prediction [6], and disease modeling [14].

One of the key issues in MTL is to characterize the relationship between different learning tasks. Some representative methods have been proposed to

D. Helic et al. (Eds.): ISMIS 2020, LNAI 12117, pp. 21–31, 2020.
https://doi.org/10.1007/978-3-030-59491-6_3

address this challenging issue. They can be categorized into two main categories [20]: feature-based MTL and parameter-based MTL. Feature-based MTL characterizes the relationships among different tasks by introducing constraints on the feature representations of given tasks to model the task similarity. According to the property of weight matrix, feature-based MTL can be further categorized into multi-task feature extraction [1,22] and multi-task feature selection [13]. Parameter-based MTL models the relationships among tasks by manipulating the parameters in each task. It can be further divided into four subcategories: low-rank methods that model the task-relation by constraining the rank of a parameter matrix [18]; task clustering methods that group similar tasks into subsets and share parameters within the cluster [8]; task-relation learning methods that describe the relationships among tasks using a specific criterion such as covariance or correlation [21]; and decomposition methods that characterize task relationships by decomposing the parameter matrix to a set of component matrices and introducing constraints on them [7].

Although the aforementioned methods have explored the task relationship from different aspects, they have not explicitly quantified the distance or similarity of different tasks, which is, of course, very important in modeling the intrinsic correlation of multiple learning tasks. To address this problem, in this paper, we introduce a new MTL method called Metric-guided MTL (M^2TL), aiming at explicitly measuring the task distance via a metric learning strategy. Specifically, we represent the distance between different learning tasks using their projection parameters. Accordingly, we propose to learn a distance metric, under which the similar learning tasks are close to each other while the uncorrelated ones are apart from each other. With the formulated distance metric, we introduce a metric-guided regularizer into the objective function of M^2TL. By jointly optimizing the loss function and the metric-guided regularizer, the learned task relationship is expected to well reflect the explicitly quantified similarity between different tasks.

The rest of the paper is organized as follows. Section 2 briefly reviews the related work on feature-based MTL. Section 3 introduces the proposed M^2TL, including the formulation of task distance, the objective function of M^2TL, the optimization procedure, and the computational complexity analysis. Section 4 validates the effectiveness of M^2TL on both synthetic and real-world benchmark datasets, demonstrating the superiority of the proposed method over existing MTL methods in terms of prediction accuracy. Section 5 draws the conclusion of the paper.

2 Related Work

The proposed M^2TL aims to learn the common feature transformation among different tasks. This section, therefore, briefly reviews some related work in feature-based MTL. Generally, the feature-based MTL approaches, including both feature extraction and feature selection methods, can be formulated under the following regularization framework:

$$\arg\min_{\mathbf{W}} Loss(\mathbf{W}) + \mu\mathcal{R}(\mathbf{W}), \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{d\times T}$ denotes the weight matrix, which is column stacked by each task's weight vector $\mathbf{w}_i$ $(i = 1, \cdots, T)$. Here d and T denote the dimension of features and the number of tasks, respectively. Moreover, $Loss(\mathbf{W})$ denotes the total loss on T learning tasks that we want to minimize, $\mathcal{R}(\mathbf{W})$ denotes the regularizer that characterize the relationship among different tasks, and $\mu \geq 0$ denotes the balancing parameter that adjusts the importance of $Loss(\mathbf{W})$ and that of $\mathcal{R}(\mathbf{W})$.

Different feature-based MTL approaches adopt their own ways to formulate the loss function $Loss(\mathbf{W})$ and the regularizer $\mathcal{R}(\mathbf{W})$, so that the relationship between different tasks can be modeled and the learning performance of all tasks can be optimized simultaneously. In [1], Argyriou et al. aimed to learn a square transformation matrix for features, which lies in the assumption that transformed feature space is more powerful than the original. They further proposed a convex formulation [2] to solve the optimization problem. In [11,13], the $L_{2,1}$-norm was used as the regularizer to select common features shared across different tasks. A more general form, the $L_{p,q}$-norm, can also be utilized to select the common or shared features as it owns the property of sparsity as well [19]. However, the $L_{p,q}$-norm may perform even worse when the value of shared features are highly uneven [9]. To address this issue, Jalali et al. proposed a method called dirty MTL in which the weight matrix is decomposed into two components, one to ensure block-structured row-sparsity and the other for element-wise sparsity with different regularizers imposed [9].

The regularizers adopted in the above methods have explored the relationship between different tasks from various perspectives. However, none of them has explicitly quantified the distance/similarity between different tasks, which is of vital importance in modeling the task correlation in MTL. This observation motivates us to propose a new MTL method, which can measure the distance between different tasks in an explicit way. Metric learning [16], which aims to learn a distance metric to reflect the intrinsic similarity/correlation between data samples, becomes a natural choice to model the task distance in our case. In recent years, metric learning has been widely used in various applications, such as healthcare [12], person re-identification [17], and instance segmentation [10]. Different from existing metric learning methods that work on data samples, the nature of MTL problem requires us to define the distance metric over tasks, so that the correlation between different tasks can be explicitly measured and integrated into the learning objectives.

3 Proposed Method

In this section, we introduce the proposed M^{2}TL. First, we define the formulation of task distance metric. Based on that, we propose the objective function of M^{2}TL and describe the optimization procedure. Finally, we analyze the computational complexity of the proposed method.

3.1 Task Distance Metric

For each task, the weight vector is a meaningful index to represent the information learned from the corresponding task. Therefore, we define a task distance metric based on weight vectors of different tasks:

$$D(t_i, t_j) = (\mathbf{w}_i - \mathbf{w}_j)^T \mathbf{M} (\mathbf{w}_i - \mathbf{w}_j),\ i, j = 1, \cdots, T, \tag{2}$$

where t_i and t_j denote the i-th task and the j-th task, respectively; $\mathbf{w}_i$ and $\mathbf{w}_j$ denote the weight vector learned for the i-th task and that for the j-th task, respectively; and $D(t_i, t_j)$ denotes the distance between task i and task j. Similar to the typical distance metric learning, we use the Mahalanobis matrix $\mathbf{M}$ to flexibly adjust the importance of different dimensions. Note that if $\mathbf{M}$ equals to the identity matrix, then the defined distance metric will reduce to the Euclidean distance.

3.2 Objective Function of M^{2}TL

With the above definition on task distance metric, we can formulate the objective function of the proposed M^{2}TL. Given T regression tasks and the corresponding datasets $\{(\mathbf{X}_1, \mathbf{y}_1), (\mathbf{X}_2, \mathbf{y}_2), \cdots, (\mathbf{X}_T, \mathbf{y}_T)\}$, where $\mathbf{X}_i \in \mathbb{R}^{n_i \times d}$ and $\mathbf{y}_i \in \mathbb{R}^{n_i}$ $(i = 1, \cdots, T)$ denote the training samples and the corresponding labels in the i-th task respectively, and n_i denotes the number of samples in the i-th task. In this paper, we use regression tasks as an example to show the formulation of the proposed M^{2}TL. In fact, the idea of the proposed method can be extended to other supervised/unsupervised learning tasks, such as classification tasks and clustering tasks, in a straightforward way. The proposed M^{2}TL specifies the general formulation in Eq. (1) as follows:

$$\arg\min_{\mathbf{W}, \mathbf{M}} Loss(\mathbf{W}) + \frac{\mu}{T^2} \sum_{i=1}^{T} \sum_{j=1}^{T} D(t_i, t_j) + \|\mathbf{M} - \mathbf{Q}\|_F^2. \tag{3}$$

The first term in Eq. (3) represents the total loss of T tasks as mentioned previously. Without loss of generality, we utilize the least squares formulation as the loss function in our model: $Loss(\mathbf{W}) = \sum_{i=1}^{T} \frac{1}{2n_i} \|\mathbf{X}_i \mathbf{w}_i - \mathbf{y}_i\|_2^2$. Note that $Loss(\mathbf{W})$ in Eq. (3) can be any loss function according to different learning requirements. The second term in Eq. (3) is the task distance formulated in the previous subsection. By minimizing the summation of all task distances, we expect that the commonality/similarity among different tasks, which is generally hidden in the original feature space, can be extracted to the maximum extent. In addition to the loss function and the regularizer, we further introduce the

last term in Eq. (3) to avoid the trivial solution on $\mathbf{M}$ by constraining it using a task correlation/covariance matrix $\mathbf{Q}$. Here $\mathbf{Q}$ can be any correlation/covariance matrix that captures the relationship between different tasks. In our paper, we use the Pearson correlation coefficient matrix calculated by the initialization of $\mathbf{W}$ because of its universality.

3.3 Optimization Procedure

To the best of our knowledge, there is no closed-form solution for the optimization problem in Eq. (3). Therefore, we use an alternating method to find the optimal $\mathbf{M}$ and $\mathbf{W}$ iteratively, which guarantees the optimality in each iteration as well as the local optimum of the solution. The detailed optimization procedure is described as follows:

Fix W and update M: With the fixed $\mathbf{W}$, the objective function in Eq. (3) can be rewritten as follows:

$$\begin{aligned}
&\arg\min_{\mathbf{W},\mathbf{M}} Loss(\mathbf{W}) + \frac{\mu}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T} D(t_i,t_j) + \|\mathbf{M}-\mathbf{Q}\|_F^2 \\
= \;& \arg\min_{\mathbf{M}} \frac{\mu}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T}(\mathbf{w}_i-\mathbf{w}_j)^T\mathbf{M}(\mathbf{w}_i-\mathbf{w}_j) + \|\mathbf{M}-\mathbf{Q}\|_F^2 \\
= \;& \arg\min_{\mathbf{M}} \frac{\mu}{T^2} tr(\mathbf{M}\sum_{i,j}^{T}(\mathbf{w}_i-\mathbf{w}_j)(\mathbf{w}_i-\mathbf{w}_j)^T) + \|\mathbf{M}-\mathbf{Q}\|_F^2 \\
= \;& \arg\min_{\mathbf{M}} \frac{2\mu}{T^2} tr(\mathbf{M}(T\sum_{i=1}^{T}\mathbf{w}_i\mathbf{w}_i^T - \sum_{i,j}^{T}\mathbf{w}_i\mathbf{w}_j^T)) + \|\mathbf{M}-\mathbf{Q}\|_F^2 \\
= \;& \arg\min_{\mathbf{M}} \frac{2\mu}{T^2} tr(\mathbf{MWLW}^T) + \|\mathbf{M}-\mathbf{Q}\|_F^2,
\end{aligned} \tag{4}$$

where $\mathbf{L}$ is a $T\times T$ Lagrange matrix defined as: $\mathbf{L} = T\mathbf{I}_T - \mathbf{1}_T\mathbf{1}_T^T$, with $\mathbf{I}_T$ and $\mathbf{1}_T$ being the T-dimensional identity matrix and all one column vector, respectively. Taking the derivative of the objective function in Eq. (4) with respect to $\mathbf{M}$ and set it to zero, i.e.,

$$\frac{\partial\left[\frac{2\mu}{T^2}tr(\mathbf{MWLW}^T) + \|\mathbf{M}-\mathbf{Q}\|_F^2\right]}{\partial\mathbf{M}} = 0, \tag{5}$$

then we can obtain the update of $\mathbf{M}$:

$$\mathbf{M} = \mathbf{Q} - \frac{\mu}{T^2}\mathbf{WLW}^T. \tag{6}$$

Algorithm 1: Metric-guided Multi-Task Learning (M^{2}TL)

Input: Training set for T learning tasks: $\{\mathbf{X}_i \in \mathbb{R}^{n_i \times d}, \mathbf{y}_i \in \mathbb{R}^{n_i}\}_{i=1}^{T}$
Output: Weight matrix $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_T] \in \mathbb{R}^{d \times T}$
1: **for** $i \leftarrow 1 : T$ **do**
2: Initialize $\mathbf{w}_i$: $\mathbf{w}_i \leftarrow \mathbf{X}_i^T \mathbf{y}_i$;
3: **end for**
4: **while** *not convergence* **do**
5: Update $\mathbf{M}$ using Eq. (6);
6: **for** $i \leftarrow 1 : T$ **do**
7: Update $\mathbf{w}_i$ using Eq. (9);
8: **end for**
9: **end while**

Fix M and update W: With the fixed $\mathbf{M}$, the objective function in Eq. (3) can be rewritten as follows:

$$\begin{aligned} & \arg\min_{\mathbf{W},\mathbf{M}} Loss(\mathbf{W}) + \frac{\mu}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T} D(t_i, t_j) + \|\mathbf{M} - \mathbf{Q}\|_F^2 \\ = & \arg\min_{\mathbf{W}} Loss(\mathbf{W}) + \frac{\mu}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T}(\mathbf{w}_i - \mathbf{w}_j)^T\mathbf{M}(\mathbf{w}_i - \mathbf{w}_j) \\ = & \arg\min_{\mathbf{W}} \sum_{i=1}^{T}\frac{1}{2n_i}\|\mathbf{X}_i\mathbf{w}_i - \mathbf{y}_i\|_2^2 + \frac{\mu}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T}(\mathbf{w}_i - \mathbf{w}_j)^T\mathbf{M}(\mathbf{w}_i - \mathbf{w}_j) \\ = & \arg\min_{\mathbf{W}} \sum_{i=1}^{T}(\frac{1}{2n_i}\|\mathbf{X}_i\mathbf{w}_i - \mathbf{y}_i\|_2^2 + \frac{\mu}{T^2}\sum_{j=1}^{T}(\mathbf{w}_i - \mathbf{w}_j)^T\mathbf{M}(\mathbf{w}_i - \mathbf{w}_j)). \end{aligned} \tag{7}$$

Note that in the above formulation, each task can be updated individually. Therefore, for the i-th task, we fix the $1, ..., i-1, i+1, ...T$-th tasks, take the derivative with respect to $\mathbf{w}_i$, and set it to zero, then we have:

$$\frac{1}{n_i}\mathbf{X}_i^T(\mathbf{X}_i\mathbf{w}_i - \mathbf{y}_i) + \frac{4\mu}{T^2}\sum_{\substack{j=1 \\ j\neq i}}^{T}\mathbf{M}(\mathbf{w}_i - \mathbf{w}_j) = 0. \tag{8}$$

From Eq. (8), we can obtain the updating rule of $\mathbf{w}_i$:

$$\mathbf{w}_i = (\frac{1}{n_i}\mathbf{X}_i^T\mathbf{X}_i + \frac{4\mu(T-1)}{T}\mathbf{M})^{-1}(\frac{1}{n_i}\mathbf{X}_i^T\mathbf{y}_i + \frac{4\mu}{T}\mathbf{M}\sum_{\substack{j=1 \\ j\neq i}}^{T}\mathbf{w}_j). \tag{9}$$

The details of the proposed M^{2}TL are described in Algorithm 1.

3.4 Computational Complexity Analysis

In Algorithm 1, the most time-consuming steps are steps 4–9. The time complexity of step 5, i.e., updating $\mathbf{M}$, is $O(dT^2)$. The time complexity of step 7,

i.e., updating $\mathbf{w}_i$, is $O(d^3)$. Assume that t is the number of iterations in the outside *while* loop for convergence, then the total computational complexity of the proposed M^{2}TL is $O(t(dT^2 + d^3T))$.

4 Experimental Results

In this section, we validate the performance of the proposed method on both synthetic and real-world datasets. We use two standard criteria in MTL for performance evaluation: the root mean squared error (RMSE) [23] and the normalized mean squared error (NMSE), which are defined as follows:

$$RMSE = \frac{\sum_{i=1}^{m} \|\mathbf{X}_i^T \mathbf{w}_i - \mathbf{y}_i\|_2 \times n_i}{\sum_{i=1}^{m} n_i}, \; NMSE = \frac{\sum_{i=1}^{m} MSE_i / var(\mathbf{y}_i) \times n_i}{\sum_{i=1}^{m} n_i}, \tag{10}$$

where n_i is the number of samples in i-th task, and $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i\mathbf{w}_i)^2$ denotes the mean square error. We select the following baselines for performance comparison:

- **STL** [15]: the classical single-task learning method, which learns each task independently without modeling the task relationship. Here we employ Lasso as the STL model.
- **L21** [13]: a typical multi-task feature selection method, which uses $L_{2,1}$-norm to achieve the row sparsity of weight matrix.
- **DirtyMTL** [9]: a representative dirty multi-task learning method, which decomposes the weight matrix into two components and regularizes these two components separately to overcome the shortage of $L_{q,p}$-norm.
- **MTFSSR** [19]: a state-of-the-art multi-task feature selection method with sparse regularization, which extends the $L_{1,2}$-norm regularization to the multi-task setting for capturing common features and extracting task-specific features simultaneously.

4.1 Experiments on Synthetic Dataset

In this subsection, we examine the convergence and the prediction accuracy of the proposed method on a synthetic dataset. We generate the data of T regression tasks. Each task includes N data samples. The data of the i-th task are generated from the normal distribution $\mathcal{N}(i/10, 1)$. The ground truth of the weight matrix, $\mathbf{W}^{(truth)}$, is generated from $\mathcal{N}(1, 1)$. The labels are then generated by: $\mathbf{y}_i = \mathbf{X}_i\mathbf{w}_i^{(truth)} + \epsilon$ accordingly, where ϵ is the Gaussian noise generated from $\mathcal{N}(0, 0.1)$. In the experiments, we assume that $\mathbf{W}^{(truth)}$ is unknown and aim to learn it from the training sets $\{\mathbf{X}_i, \mathbf{y}_i\}$ $(i = 1, \cdots, T)$. We set $T = 10$, $d = 10$, and $N = 30$ in our experiments.

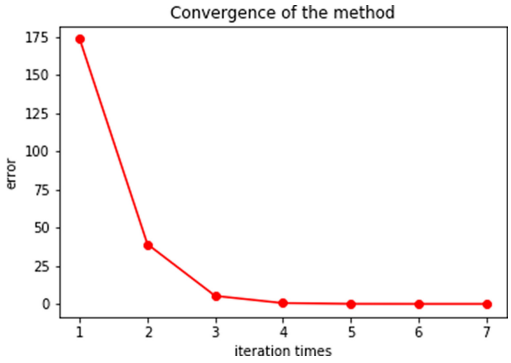

Fig. 1. Convergence speed of the proposed method on the synthetic dataset. The horizontal axis represents the number of iterations while the vertical axis represents the objective value of $\|\mathbf{W}^t - \mathbf{W}^{t-1}\|_F$. The red curve in the figure shows that M^{2}TL can converge to a stable solution quickly. (Color figur online)

We first examine the convergence speed of the proposed method. We determine the stopping point by evaluating the difference between $\mathbf{W}^t$ and $\mathbf{W}^{t-1}$. Specifically, we consider the algorithm as convergent and stop the iteration once the following inequality is satisfied: $\|\mathbf{W}^t - \mathbf{W}^{t-1}\|_F \leq \epsilon$. In this experiment, we set $\epsilon = 0.001$. Figure 1 shows the objective value of $\|\mathbf{W}^t - \mathbf{W}^{t-1}\|_F$ versus the number of iterations. The objective value decreases dramatically and satisfies the stopping criterion within only 7 iterations, demonstrating that the proposed method can converge to a stable solution quickly.

Table 1. The performance (in terms of RMSE) of STL [15], L21 [13], DirtyMTL [9], MTFSSR [19], and the proposed M^{2}TL on the synthetic dataset, with three different training ratios. The best performances are highlighted in bold.

Training No.	Methods				
	STL	L21	DirtyMTL	MTFSSR	M^{2}TL
2	60.49 ± 12.45	31.07 ± 9.24	29.56 ± 4.64	38.90 ± 8.95	$\mathbf{25.40 \pm 12.26}$
4	53.36 ± 12.02	25.22 ± 3.23	25.69 ± 5.60	25.55 ± 7.24	$\mathbf{24.93 \pm 13.75}$
6	46.63 ± 19.95	18.70 ± 2.15	20.33 ± 5.53	18.56 ± 4.64	$\mathbf{18.44 \pm 8.39}$

In the second experiment, we compare the performance of the proposed method with that of four aforementioned baselines. We select 2, 4, and 6 samples from the 30 samples for training and use the rest for testing. We repeat the experiment for 10 times on randomly selected training samples and report the average RMSE as well as the standard deviation of each method. Table 1 lists the results of all methods. Obviously, with the exploration on the relationship between different tasks, the MTL methods (including L21, DirtyMTL, MTFSSR, and the proposed M^{2}TL) achieve the lower RMSE than the STL method.

By further modeling the task distance in an explicit way, the proposed method outperforms other three MTL methods in all scenarios.

4.2 Experiments on Real-World Multi-task Datasets

In this subsection, we conduct experiments on two real-world multi-task datasets: the School dataset (https://github.com/jiayuzhou/MALSAR/tree/master/data) and the Sarcos dataset (http://www.gaussianprocess.org/gpml/data/). The School dataset is commonly used in many MTL literature. It is provided by the Inner London Education Authority and contains 15,362 data samples from 139 schools where each data sample has 27 attributes. We treat each school as a task and the learning target is to predict the exam score. The Sarcos dataset is about the inverse dynamic problem. The learning target is to predict the 7 joint torques given the 7 joint positions, 7 joint velocities and 7 joint accelerations. Here we treat prediction of one joint torque as a task so we have 7 tasks in total. For all 7 tasks, the 21 features (7 joint positions, 7 joint velocities and 7 joint accelerations) are used as the input, so the training data for different tasks are the same. We select 200 samples from each task to conduct our experiment. For both School and Sarcos datasets, we 20%, 30%, 40% of data for training and use the rest for testing. Similar to the synthetic experiments, we repeat the experiment for 10 times on randomly selected training samples. We report the average NMSE as well as the standard deviation of each method.

Tables 2 and 3 report the performance of all five methods on the School dataset and the Sarcos dataset, respectively. With the increase of training ratio, the NMSE of all methods decreases (except the L21 method from 20% to 30%), which is consistent with the common observation that providing more training data is generally beneficial to the learning task. Moreover, similar to the observations in the synthetic experiments, the MTL methods perform better than the STL method while our method again achieves the lowest prediction error among all five methods, demonstrating the necessity of exploring the task relationship in MTL and the effectiveness of the proposed formulation.

Table 2. The performance (in terms of NMSE) of STL [15], L21 [13], DirtyMTL [9], MTFSSR [19], and the proposed M^{2}TL on the School dataset, with three different training ratios. The best performances are highlighted in bold.

Training ratio	Methods				
	STL	L21	DirtyMTL	MTFSSR	M^{2}TL
20%	2.118 ± 0.104	0.963 ± 0.005	$\mathbf{0.924 \pm 0.001}$	0.929 ± 0.000	0.931 ± 0.001
30%	2.052 ± 0.008	1.061 ± 0.007	0.858 ± 0.001	0.862 ± 0.000	$\mathbf{0.855 \pm 0.007}$
40%	2.013 ± 0.001	1.016 ± 0.001	0.821 ± 0.003	0.820 ± 0.000	$\mathbf{0.809 \pm 0.003}$

Table 3. The performance (in terms of NMSE) of STL [15], L21 [13], DirtyMTL [9], MTFSSR [19], and the proposed M^2TL on the Sarcos dataset, with three different training ratios. The best performances are highlighted in bold.

Training ratio	Methods				
	STL	L21	DirtyMTF	MTFSSR	M^2TL
20%	2.180 ± 0.037	0.309 ± 0.113	0.294 ± 0.001	0.300 ± 0.004	$\mathbf{0.291 \pm 0.004}$
30%	2.052 ± 0.026	0.216 ± 0.001	0.226 ± 0.001	0.216 ± 0.002	$\mathbf{0.211 \pm 0.002}$
40%	2.000 ± 0.091	0.208 ± 0.001	0.202 ± 0.000	0.194 ± 0.000	$\mathbf{0.189 \pm 0.001}$

5 Conclusions

In this paper, we proposed a novel multi-task learning method called Metric-guided Multi-Task Learning (M^2TL), which learns a task distance metric to explicitly measure the distance between different tasks and uses the learned metric as a regularizer to model the multi-task correlation. In the future, we plan to extend our experimental evaluation on large-scale datasets with distributed strategy in order to overcome the issues of quadratic time compelxity with respect to the task number and cubic time complexity with respect to the feature dimension. Moreover, we will investigate more sophisticated ways to model and learn the task distance metric.

Acknowledgement. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the Grants from the Research Grant Council of Hong Kong SAR under Projects RGC/HKBU12201318 and RGC/HKBU12201619.

References

1. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: Proceedings 19th NIPS, pp. 41–48 (2007)
2. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Mach. Learn. **73**(3), 243–272 (2008)
3. Caruana, R.: Multitask learning. Mach. Learn. **28**(1), 41–75 (1997)
4. Chu, X., Ouyang, W., Yang, W., Wang, X.: Multi-task recurrent neural network for immediacy prediction. In: Proceedings 15th ICCV, pp. 3352–3360 (2015)
5. Deng, D., Shahabi, C., Demiryurek, U., Zhu, L.: Situation aware multi-task learning for traffic prediction. In: Proceedings 17th ICDM, pp. 81–90 (2017)
6. Goncalves, A., Banerjee, A., Zuben, F.V.: Spatial projection of multiple climate variables using hierarchical multitask learning. In: Proceedings 31th AAAI, pp. 4509–4515 (2017)
7. Gong, P., Ye, J., Zhang, C.: Robust multi-task feature learning. In: Proceedings 18th SIGKDD, pp. 895–903 (2012)
8. Jacob, L., Vert, J.P., Bach, F.R.: Clustered multi-task learning: a convex formulation. In: Proceedings 21th NIPS, pp. 745–752 (2009)
9. Jalali, A., Sanghavi, S., Ruan, C., Ravikumar, P.K.: A dirty model for multi-task learning. In: Proceedings 22th NIPS, pp. 964–972 (2010)

10. Lahoud, J., Ghanem, B., Pollefeys, M., Oswald, M.R.: 3d instance segmentation via multi-task metric learning (2019). arXiv preprint arXiv:1906.08650
11. Liu, J., Ji, S., Ye, J.: Multi-task feature learning via efficient l 2, 1-norm minimization. In: Proceedings 25th UAI, pp. 339–348 (2009)
12. Ni, J., Liu, J., Zhang, C., Ye, D., Ma, Z.: Fine-grained patient similarity measuring using deep metric learning. In: Proceedings 26th CIKM, pp. 1189–1198. ACM (2017)
13. Obozinski, G., Taskar, B., Jordan, M.: Multi-task feature selection. University of California, Berkeley, Technical report (2006)
14. Pei, H., B.Yang, Liu, J., Dong, L.: Group sparse Bayesian learning for active surveillance on epidemic dynamics. In: Proceedings 32th AAAI, pp. 800–807 (2018)
15. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. Ser. B (Methodological) **58**(1), 267–288 (1996)
16. Xing, E.P., Jordan, M.I., Russell, S.J., Ng, A.Y.: Distance metric learning with application to clustering with side-information. In: Proceedings 15th NIPS, pp. 521–528 (2003)
17. Yu, H.X., Wu, A., Zheng, W.S.: Cross-view asymmetric metric learning for unsupervised person re-identification. In: Proceedings 16th ICCV, pp. 994–1002 (2017)
18. Zhang, J., Ghahramani, Z., Yang, Y.: Learning multiple related tasks using latent independent component analysis. In: Proceedings 18th NIPS, pp. 1585–1592 (2006)
19. Zhang, J., Miao, J., Zhao, K., Tian, Y.: Multi-task feature selection with sparse regularization to extract common and task-specific features. Neurocomputing **340**, 76–89 (2019)
20. Zhang, Y., Yang, Q.: A survey on multi-task learning (2018). arXiv:1707.08114v2
21. Zhang, Y., Yeung, D.Y.: Multi-task boosting by exploiting task relationships. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) Proceedings 22th ECML PKDD, pp. 697–710 (2012)
22. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Facial landmark detection by deep multi-task learning. In: Proceedings 13th ECCV, pp. 94–108 (2014)
23. Zhou, J., Chen, J., Ye, J.: Malsar: multi-task learning via structural regularization. Arizona State University, Technical report (2011)