Eigenfunction-Based Multitask Learning in a Reproducing Kernel Hilbert Space

Xinmei Tian, *Member, IEEE*, Ya Li, Tongliang Liu, Xinchao Wang, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— Multitask learning aims to improve the performance on related tasks by exploring the interdependence among them. Existing multitask learning methods explore the relatedness among tasks on the basis of the input features and the model parameters. In this paper, we focus on nonparametric multitask learning and propose to measure task relatedness from a novel perspective in a reproducing kernel Hilbert space (RKHS). Past works have shown that the objective function for a given task can be approximated using the top eigenvalues and corresponding eigenfunctions of a predefined integral operator on an RKHS. In our method, we formulate our objective for multitask learning as a linear combination of two sets of eigenfunctions, common eigenfunctions shared by different tasks and unique eigenfunctions in individual tasks, such that the eigenfunctions for one task can provide additional information on another and help to improve its performance. We present both theoretical and empirical validations of our proposed approach. The theoretical analysis demonstrates that our learning algorithm is uniformly argument stable and that the convergence rate of the generalization upper bound can be improved by learning multiple tasks. Experiments on several benchmark multitask learning data sets show that our method yields promising results.**

***Index Terms*— Eigenfunction-based learning, multitask learning, regression, task relatedness.**

I. Introduction

In RECENT years, multitask learning has been widely studied in various fields, such as metric learning [1]–[3], image and video research [4], and disease prediction [5], [6]. The main advantage of multitask learning is the ability to explore the intrinsic interdependence among different tasks, through which all tasks can benefit each other. As a result, multitask learning methods usually achieve better performance than single-task learning methods.

The key challenge of multitask learning is measuring the relationships among different tasks. Existing multitask learning methods can be categorized into two classes based on the methods used to measure the task relationships. In the first class of methods, it is assumed that related tasks share a set of common features [7]–[10], while in the second class of methods, it is assumed that different tasks share some common parameters [10]–[14]. Both classes of methods involve the imposition of direct regularizations, on either features or parameters, to learn the relatedness of multiple tasks. However, some of these regularizations are too strong, making the objective functions difficult to solve.

In this paper, we propose a novel multitask learning algorithm that utilizes a different measure of task relatedness. Unlike previous methods, in which task relatedness is measured using common features or model parameters, our method measures the interdependence among tasks through the relatedness of eigenfunctions. The objective function for a particular task, in a manner similar to regression, can be approximated as a linear combination of the top eigenvalues of a predefined integral operator on a reproducing kernel Hilbert space (RKHS) [15]–[17]. In our method, we assume that related tasks share a set of common eigenfunctions and that each task also has a set of unique eigenfunctions, which are sparse. We formulate our objective function as a linear combination of both sets of eigenfunctions, such that the functions associated with one task may provide additional information to benefit others. Moreover, since the eigenfunctions can be explicitly computed from the input features with an associated kernel function, our method can be readily extended to any type of kernel version. Please note that our method is not suitable for parametric multitask learning problems [18].

We propose an efficient optimization algorithm for solving our objective function, which has two regularization terms. One is an L1-norm regularization to guarantee the sparsity of the task-specific eigenfunctions. The other is an L2-norm regularization on the shared eigenfunctions to constrain the complexity of the trained model. We present a theoretical analysis to show that our learning algorithm is uniformly argument stable, meaning that the output is not sensitive to subtle changes in the input. In addition, we show that the convergence rate of the generalization upper bound is related to the number of training samples and the number of tasks. This means that when either the training set size or the number

Manuscript received February 27, 2018; revised June 14, 2018 and September 8, 2018; accepted September 25, 2018. Date of publication October 29, 2018; date of current version May 23, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1002203, in part by the National Natural Science Foundation of China under Grant 61872329 and Grant 61572451, in part by Fok Ying Tung Education Foundation under Grant WF2100060004, and in part by the Australian Research Council Projects under Grant FL-170100117, Grant DP-180103424, and Grant IH180100002. *(Corresponding author: Xinmei Tian.)*

X. Tian and Y. Li are with the CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application Systems, University of Science and Technology of China, Hefei 230027, China (e-mail: xinmei@ustc.edu.cn; muziyiye@mail.ustc.edu.cn).

T. Liu and D. Tao are with the UBTECH Sydney Artificial Intelligence Centre, School of Information Technologies, Faculty of Engineering and Information Technologies, The University of Sydney, Darlington, NSW 2008, Australia (e-mail: tongliang.liu@sydney.edu.au; dacheng.tao@sydney.edu.au).

X. Wang is with the Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ 07030, USA (e-mail: xinchao.wang@stevens.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2873649

of tasks increases, the generalization error will decrease. Our experimental results obtained on benchmark data sets further validate our proposed approach.

The remainder of this paper is organized as follows. In Section II, we briefly review related works. We present the details of our proposed algorithm and the optimization algorithm in Section III, and we derive a theoretical analysis to demonstrate the effectiveness of the proposed multitask learning method in Section IV. We report experimental results obtained on several landmark data sets in Section V. We conclude this paper and discuss future work in Section VI.

II. Related Work

Recent works have demonstrated the success and development of multitask learning in various domains [1], [7], [19]–[23]. In traditional single-task learning methods, related tasks are learned separately, and the positive interactions among different tasks are ignored, which lead to a loss of valuable information regarding the data distribution. Given the drawbacks of single-task learning methods, multitask learning has been proposed to explore the intrinsic relatedness among different tasks through the joint learning of multiple tasks. Proper measurements of task relatedness can help to gain additional information on all tasks, particularly when the number of training data is insufficient. Additional information gained from other tasks can help to compensate for the lack of training samples. Consequently, multitask learning is applied with the goal of improving the performance on all tasks.

Given the successful applications of multitask learning, various traditional single-task learning methods have been extended to multitask learning algorithms. For example, support vector machine (SVM) [24], as one of the most popular machine learning algorithms, has been investigated in various multitask learning studies [11], [25]–[27]. Evgeniou and Pontil [11] proposed a classic SVM-based multitask learning framework, which has been referenced by many other researchers. In the proposed regularized multitask learning method, it is assumed that the hyperplanes of all tasks are close to one central hyperplane with an offset. Li *et al.* [25] extended the proximal SVM approach into a multitask learning framework to improve the efficiency of multitask learning. The proposed multitask proximal vector machine model can be solved explicitly with high efficiency and comparable performance. Jebara [26] proposed learning a common feature selection and kernel selection for multitask SVMs with maximum entropy discrimination. Metric learning has also been extended to a multitask learning framework. Parameswaran and Weinberger [1] studied multitask large margin nearest neighbor metric learning and achieved much better performance than that achieved with single-task large margin nearest-neighbor metric learning. Ma *et al.* [2] applied multitask distance metric learning for person reidentification and achieved considerably improved performance. In their approach, multiple distance metrics are learned jointly to measure the distances of images from different camera pairs.

In recent years, multitask deep learning has been applied in various research fields. Zhang *et al.* [28] utilized multitask deep learning to improve the robustness of facial landmark detection by simultaneously considering correlated tasks such as head pose estimation and facial attribute inference. Liu *et al.* [29] applied multitask deep learning in video thumbnail selection, using two highly related data sets to explore query-thumbnail relevance. In the work of Zhang *et al.* [30], multitask deep convolutional neural networks were utilized to improve performance in multiview face detection. The constructed multitask deep neural networks were simultaneously trained for face/nonface decisions, face pose estimation, and facial landmark localization.

Due to the good performance of multitask learning in various applications, some researchers have attempted to theoretically demonstrate the merits of multitask learning [21], [23], [31], [32]. Liu *et al.* [31] proposed an algorithm-dependent generalization bound for multitask learning based on algorithmic stability. Subject to a mild assumption regarding the feature structures, the authors observed that the functions associated with other tasks can be viewed as regularizers for a given task. Li *et al.* [21] proposed the use of the RKHS of vector-valued functions as a hypothesis space for multitask classification. They derived an improved empirical Rademacher complexity-based generalization bound and discussed the relationship between a group lasso regularizer and the proposed hypothesis space. An algorithm for multitask learning from unlabeled data was proposed by Ando and Zhang [32]. Their paper presented a general framework for formulating the structural learning problem and analyzed it theoretically. Maurer *et al.* [23] applied sparse coding in multitask learning and transfer learning. Their paper adopted the assumption that the parameters of tasks can be approximated well through a sparse linear combination of the atoms of a high-dimensional dictionary, and a generalization error bound for the proposed approach was given. All these works have presented valid theoretical analyses of multitask learning.

Multitask learning is based on the assumption that the tasks to be learned are indeed related. However, the method for measuring the relatedness among different tasks is always a key problem. Common feature representation sharing and common parameter sharing are two popular methods of exploring the relatedness among multiple tasks. Among methods based on feature sharing, Argyriou *et al.* [7] proposed a convex multitask feature learning (CMTL) algorithm with L21-norm regularization of the parameters. This regularization has the ability to ensure the learning of a sparse feature representation shared across different tasks. Zhang and Yeung [33] proposed a convex formulation for multitask learning that can be used to estimate task relationships automatically. Ciliberto *et al.* [34] proposed a general computational framework for multitask learning in which *a priori* knowledge of the task structure is encoded with a convex penalty. In the setting of that paper, some previous proposals could be recovered as special cases. A nonconvex multitask sparse feature learning method was proposed by Gong *et al.* [9]. The authors noted the drawbacks of previous convex formulations of multitask feature learning and argued that the proposed method could achieve a better parameter estimation error bound that could be achieved with a convex formulation.

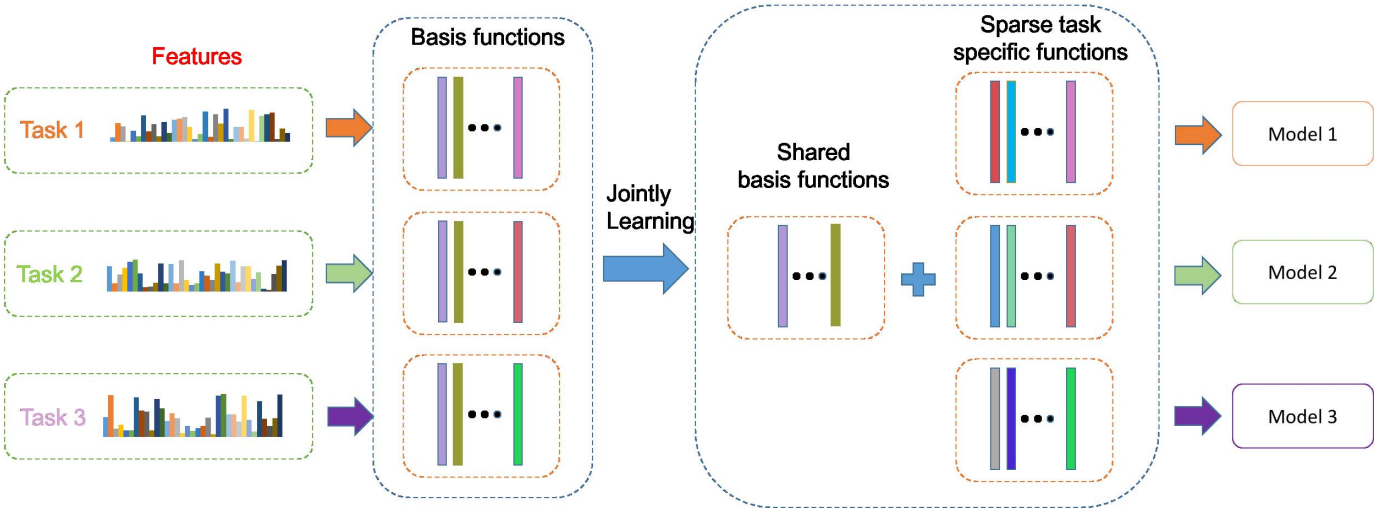


Fig. 1. Framework of our proposed EMTL method. Three different tasks are considered in this figure. We first explicitly learn the top eigenfunctions from the input features for each task. Then, all tasks are learned jointly to identify a set of shared eigenfunctions from among those for all tasks. Because of the uniqueness of each task, each task also has another set of sparse basis eigenfunctions that represent its unique characteristics. The final model for each task is approximated as a combination of the shared eigenfunctions and the particular eigenfunctions in each task.

Considering methods based on parameter sharing, Rai and Daume [14] proposed a nonparametric Bayesian model that captures task relatedness under the assumption that the task parameters share a latent subspace. In addition, the proposed method can use both labeled and unlabeled data to assist in learning this subspace, leading to further improvement in performance. Xue *et al.* [12] proposed an efficient multitask learning algorithm based on a Dirichlet process-based statistical model. The proposed algorithm can automatically group similar tasks whose training data might be drawn from similar distributions.

In most of these methods, task relatedness is measured directly by means of regularizations applied to features from the training data or to the model parameters. However, some regularizations are too strict, and the objective functions are difficult to optimize. For example, in the method proposed in [7], the features are regularized with an L21-norm regularization, which assumes that all tasks share a subset of features. This assumption is too strong because it ignores the possibility that some tasks may have features that are not shared with other tasks. In addition, the objective function is nonconvex because of the L21-norm regularization. It is difficult to solve such nonconvex problems directly. Instead, such a problem must be transformed into an equivalent convex optimization problem for efficient computation. In addition, the extension of such methods to kernel methods is usually complicated, and thus, the resulting methods are difficult to implement. For example, the method proposed in [33] has a convex formulation for multitask learning. However, when it is extended to a kernel version, the objective function must be changed, and the optimization procedure becomes more complex.

III. Eigenfunction-Based Multitask Learning Method

In this section, we present the details of our eigenfunction-based multitask learning (EMTL) method. We first introduce the algorithm for approximating the target function using eigenfunctions and then describe our EMTL method, followed by an iterative optimization algorithm for optimizing the objective function.

The framework of our EMTL method is illustrated in Fig. 1, where we consider three different tasks. Each task is associated with a set of features that are used to learn the eigenfunctions for that task. All the tasks are then learned jointly using the eigenfunctions from all tasks. We assume that all tasks share a set of eigenfunctions and that each task also has a sparse set of task-specific eigenfunctions that represent its characteristics. The final model for each task can be approximated as a combination of the shared eigenfunctions and its task-specific eigenfunctions.

A. Explicit Eigenfunction Learning

Here, we give a brief introduction to the algorithm for learning explicit eigenfunctions from features to approximate the target function, which is mainly inspired by [15] and [16]. Suppose that we have a data set of n samples, $D_t = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i \in \mathcal{X}$ denotes the ith input feature from a compact manifold in the Euclidean space $\mathbb{R}^m$ and $y_i \in \mathcal{Y}$ is the corresponding output in the Euclidean space $\mathbb{R}$. Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, with a Borel probability measure ρ. In addition, let $\rho_{\mathcal{X}}$ be the marginal probability on $\mathcal{X}$, and let $\rho(y|x)$ be the conditional probability of y given x. In this paper, we mainly focus on the regression problem $y = f(x)$, where $f(\cdot)$ is our target function. Then, the regression function $f_\rho(x)$ can be formulated as follows:

$$f_\rho(x) = \int_Y y d\rho(y|x). \tag{1}$$

Our goal is to approximate an accurate prediction function $f_\rho(x)$ using the given training data in an RKHS. Let $K(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a Mercer kernel, and let $\mathcal{H}_K$ be an RKHS associated with the Mercer kernel $K(\cdot,\cdot)$. An integral operator

L_K on $\mathcal{H}_K$ is defined as follows:

$$L_K(f) = \int_{\mathcal{X}} K(\cdot, x) f(x) d\rho_{\mathcal{X}}(x), \quad f \in \mathcal{H}_K. \tag{2}$$

Let $(\Phi_i(x), \lambda_i), i = 1, 2, \ldots, n$, be the eigenfunctions and eigenvalues of L_K ranked in descending order of the eigenvalues, where the eigenfunctions $\Phi_i(x), i = 1, 2, \ldots, n$, form an orthonormal basis of $\mathcal{H}_K$. The regression function $f(x)$ can be approximated by a linear combination of the top m eigenfunctions with nonzero eigenvalues of L_K [15]. This function can be formulated as follows:

$$f(x) = \sum_{i=1}^{m} C_i \Phi_i(x) \tag{3}$$

where m is the number of top eigenfunctions that are used to approximate the target regression function and can be determined empirically. C_i is the coefficient of the ith eigenfunction $\Phi_i(\cdot)$. The eigenpairs $(\lambda_i, \Phi_i(\cdot))$ can be explicitly found from the given features of the training data as follows. Let $\mathbb{K}$: $(K(x_i, x_j))_{i,j=1}^n$ be the Gramian matrix formed by the kernel $K(\cdot, \cdot)$ with the training data, and let $d \leq n$ be the rank of the Gramian matrix. The eigenvalues are arranged in descending order as $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_d \geq \hat{\lambda}_{d+1} = \cdots = \hat{\lambda}_n$, and the corresponding eigenvectors are $\{\hat{u}_i\}_{i=1}^n$, which form an orthonormal basis of $\mathbb{R}^n$. We thus have

$$\lambda_i = \frac{\hat{\lambda}_i}{n}$$
$$\Phi_i(\cdot) = \frac{1}{\sqrt{\hat{\lambda}_i}} \sum_{j=1}^{n} (\hat{u}_i)_j K(\cdot, x_j), \quad \text{for } i = 1, \ldots, d. \tag{4}$$

Since the eigenfunctions can be explicitly computed, our goal is to obtain their corresponding coefficients. In Section III-B, we propose our EMTL algorithm, which uses these explicitly computed eigenfunctions. The coefficients for all tasks are jointly learned by means of their shared set of eigenfunctions.

B. Eigenfunction-Based Multitask Learning Algorithm

Suppose that we have T different tasks, each of which is related to a set of data $D_t = \{(x_{t1}, y_{t1}), (x_{t2}, y_{t2}), \ldots, (x_{tn_t}, y_{tn_t})\}$, where n_t is the number of training samples for task t. We first compute the eigenpairs $\{\lambda_{ti}, \Phi_{ti}(\cdot)\}$ for task t according to (4), where λ_{ti} is the eigenvalue and Φ_{ti} is the corresponding eigenfunction. For clarity of notation, we rewrite the eigenpairs for task t as follows. Let $d_t \leq n_t$ be the rank of the Gramian matrix $(K(x_{ti}, x_{tj})_{i,j=1}^{n_t})$. We order the eigenvalues as $\hat{\lambda}_{t1} \geq \cdots \geq \hat{\lambda}_{d_t} \geq \hat{\lambda}_{d_t+1} = \cdots = \hat{\lambda}_{n_t} = 0$, and the associated eigenvectors are $\{\hat{\mu}_{ti}\}_{ti=1}^{n_t}$. We have

$$\lambda_{ti} = \frac{\hat{\lambda}_{ti}}{n_t}$$
$$\Phi_{ti}(\cdot) = \frac{1}{\sqrt{\hat{\lambda}_{ti}}} \sum_{j=1}^{n_t} (\mu_{ti})_j K(\cdot, x_{tj}), \quad \text{for } i = 1, \ldots, d_t. \tag{5}$$

Our method measures the relatedness of different tasks through the eigenfunctions $\{\Phi_{ti}(\cdot)\}$. We assume that some eigenfunctions are shared among the tasks and that the eigenfunctions for one task may benefit the others. To prevent all tasks from being performed similarly due to the influence of the shared eigenfunctions, our model maintains a set of nonshared eigenfunctions for each task. The objective of our EMTL method is formulated as follows:

$$\min_{C_t, C_0} \frac{1}{T} \sum_{t=1}^{T} \frac{1}{n_t} \sum_{i=1}^{n_t} \left(\sum_{j=1}^{d} (C_{tj} + C_{0j}) \Phi_j(x_{ti}) - y_{ti} \right)^2 + \gamma \|C\|_1 + \beta \|\Phi^\top C_0\|_2^2 \tag{6}$$

where $C = [C_1, C_2, \ldots, C_T]$ and $d = d_1 + d_2 + \cdots + d_T$, with the latter denoting the total number of eigenfunctions identified from all tasks. The eigenfunctions from all tasks, $\{\Phi_{ti}(\cdot)\}_{i=1}^{d_t}, t = 1, \ldots, T$, are combined into a single complete set, $\{\Phi_j(\cdot)\}_{j=1}^{d}$.

To consider the effects of $(1/T)$ and $(1/n_t)$, we adopt the notation $\Phi(x_{ti}) = [\Phi_1(x_{ti}) \times (1/\sqrt{Tn_t}), \Phi_2(x_{ti}) \times (1/\sqrt{Tn_t}), \ldots, \Phi_d(x_{ti}) \times (1/\sqrt{Tn_t})]^\top \in \mathbb{R}^d$ to represent a vector of the values of all eigenfunctions given the input x_{ti}. In addition, $X_t = [x_{t1}, x_{t2}, \ldots, x_{tn_t}] \in \mathbb{R}^{m \times n_t}$ denotes the data matrix from task t, and $\Phi(X_t) = [\Phi(x_{t1}), \Phi(x_{t2}), \ldots, \Phi(x_{tn_t})] \in \mathbb{R}^{d \times n_t}$ is a matrix of the values of all eigenfunctions given the training data for task t. Let Φ denote a matrix with entries corresponding to the values of all eigenfunctions given the inputs from all tasks, $\Phi = [\Phi(X_1), \Phi(X_2), \ldots, \Phi(X_T)]$. $C_0 \in \mathbb{R}^d$ is the vector of the coefficients of the shared eigenfunctions, and C_{0j} denotes the jth entry of the vector C_0. $C_t \in \mathbb{R}^d$ is the coefficient vector for the task-specific eigenfunctions of task t. The first term in the objective function is the loss between the prediction output and the ground truth. The second term is the regularization of the coefficients C_t. We constrain C_t using an L1-norm regularization, which leads to a sparse set of coefficients. The third term is a Tikhonov regularization of C_0 with a Tikhonov matrix Φ, which controls the complexity of the model. γ and β are two tradeoff parameters, which can be determined empirically. If (γ/β) is set at a large value, then the coefficient vector C_t will tend toward zero; in this case, all tasks are closely related and tend to share most eigenfunctions, with few or no task-specific eigenfunctions. By contrast, when the value (γ/β) approaches zero, we obtain small values of the coefficients in C_0 that correspond to the shared eigenfunctions, in which case, the above-mentioned objectives can be viewed as T separate single-task learning problems that are very weakly related.

Let $\{\hat{C}_t\}_{t=1}^{d}$ and $\hat{C}_0$ be the solutions to the above-mentioned objective functions. Our target regression function for task t can be written as follows:

$$f_t(x_{ti}) = \sum_{j=1}^{d} (\hat{C}_{tj} + \hat{C}_{0j}) \Phi_{tj}(x_{ti}) \tag{7}$$

where x_{ti} is the ith input for task t and $f_t(x_{ti})$ is the predicted value for task t with input x_{ti}.

C. Optimization Algorithm

In this section, we present our iterative algorithm for optimizing the above-mentioned objective function (6) with

respect to C_0 and $\{C_t\}_{t=1}^T$. The details of the algorithm are given in Algorithm 1.

It is difficult to obtain the closed-form solution $(\{C_t\}_{t=1}^T, C_0)$ to the objective function because of the shared coefficients C_0. We therefore iteratively optimize the objective function with respect to C_0 and $\{C_t\}_{t=1}^T$. We first optimize the objective function with respect to the parameter vector C_0 by fixing the parameter vectors $\{C_t\}_{t=1}^T$. For simplicity and clarity of notation, we introduce some additional variables and rewrite the formulation given in (6). Recall that $\Phi(x_{ti}) = [\Phi_1(x_{ti}) \times (1/\sqrt{Tn_t}), \Phi_2(x_{ti}) \times (1/\sqrt{Tn_t}), \ldots, \Phi_d(x_{ti}) \times (1/\sqrt{Tn_t})]^\top \in \mathbb{R}^d$ is a vector of the values of all eigenfunctions given the input x_{ti}, that $X_t = [x_{t1}, x_{t2}, \ldots, x_{tn_t}] \in \mathbb{R}^{m \times n_t}$, and that $\Phi(X_t) = [\Phi(x_{t1}), \Phi(x_{t2}), \ldots, \Phi(x_{tn_t})] \in \mathbb{R}^{d \times n_t}$ is a matrix of the values of all eigenfunctions given the training data for task t. The optimization with respect to C_0 requires the training data for all tasks. Therefore, we adopt the notation $\Phi(X) = \text{bdiag}(\Phi(X_1), \Phi(X_2), \ldots, \Phi(X_T)) \in \mathbb{R}^{dT \times N}$, where $\text{bdiag}(\Phi(X_1), \Phi(X_2), \ldots, \Phi(X_T))$ is a block diagonal matrix whose diagonal entries correspond to the outputs of all eigenfunctions given the data for task t, that is,

$$\Phi(X) = \begin{pmatrix} \Phi(X_1) & & & & \\ & \Phi(X_2) & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \Phi(X_T) \end{pmatrix}.$$

N denotes the total number of training samples for all tasks, as follows:

$$N = n_1 + n_2 + \cdots + n_T.$$

The output for all tasks is denoted by $Y = [Y_1^\top, Y_2^\top, \ldots, Y_T^\top]^\top \in \mathbb{R}^N$, where $Y_t = [y_{t1} \times (1/\sqrt{Tn_t}), y_{t3} \times (1/\sqrt{Tn_t}), \ldots, y_{tn_t} \times (1/\sqrt{Tn_t})]^\top$, considering the effects of $(1/T)$ and $(1/n_t)$ in (6). In addition, let $\hat{C} = [C_1^\top, C_2^\top, \ldots, C_T^\top]^\top \in \mathbb{R}^{dT}$, let I be the identity matrix of dimension d, and let $I_0 = [\underbrace{I, I, \ldots, I}_{T}]^\top \in \mathbb{R}^{dT \times d}$. We introduce a new variable $\hat{C}_0 = I_0 \times C_0$. By ignoring the regularization term $\gamma \|C\|_1$, the formulation given in (6) can then be reformulated as follows:

$$\min_{C_0} \|Y - \Phi(X)^\top (\hat{C}_0 + \hat{C})\|_2^2 + \beta \|\Phi^\top C_0\|_2^2. \tag{8}$$

We replace $\hat{C}_0$ with $I_0 \times C_0$ and rewrite the above-mentioned objective as a standard L2-norm regularized regression problem

$$\min_{C_0} \|Y - \Phi(X)^\top (I_0 \times C_0 + \hat{C})\|_2^2 + \beta \|\Phi^\top C_0\|_2^2. \tag{9}$$

The solution to this L2-norm regularized problem can be explicitly obtained as follows:

$$C_0 = \left(I_0^\top \Phi(X)\Phi(X)^\top I_0 + \beta \Phi^\top \Phi\right)^{-1} \times \left(I_0^\top \Phi(X) Y - I_0^\top \Phi(X)\Phi(X)^\top \hat{C}\right). \tag{10}$$

With the explicit solution for the shared coefficients C_0 obtained by fixing $\{C_t\}_{t=1}^T$, we then optimize $\{C_t\}_{t=1}^T$ by fixing C_0. The optimization of $\{C_t\}_{t=1}^T$ can be separated into T different tasks when C_0 is fixed. For task t, the optimization problem can be reformulated with the additional variables as follows:

$$\min_{C_t} \|\Phi(X_t)^\top (C_t + C_0) - Y_t\|_2^2 + \gamma \|C_t\|_1 \tag{11}$$

which is a standard L1-norm regularized regression problem. Such L1-norm regularized regression problems have been extensively investigated in the past and can be solved using various methods, such as those presented in [35]–[38]. The final iterative optimization algorithm is given in Algorithm 1.

D. Time Complexity Analysis

We now present an analysis of the computational complexity of Algorithm 1. In Algorithm 1, the computational cost mainly arises from the optimization of (9) and (11). Note that the eigenpairs in the first step of Algorithm 1 can be computed ahead of time and stored for the following optimization steps. Equation (10) is the closed-form solution to (9), and it can be reformulated as follows:

$$C_0 = \left(I_0^\top \Phi(X)\Phi(X)^\top I_0 + \beta \Phi^\top \Phi\right)^{-1} \times I_0^\top \Phi(X) \times (Y - \Phi(X)^\top \hat{C}). \tag{12}$$

The computation of $(I_0^\top \Phi(X)\Phi(X)^\top I_0 + \beta \Phi^\top \Phi)$ has a time complexity of $O(NTd^2)$, and the computation of $I_0^\top \Phi(X)$ also has a time complexity of $O(NTd^2)$. In addition, the computation of $(Y - \Phi(X)^\top \hat{C})$ has a time complexity of $O(NTd)$. Considering that the inversion of the matrix $(I_0^\top \Phi(X)\Phi(X)^\top I_0 + \beta \Phi^\top \Phi)$ has a time complexity of $O(d^3)$, the final computational time complexity for solving problem (9) is $O(NTd^2 + d^3)$, which depends on the number of training data for all tasks, the number of tasks and the number of selected top eigenfunctions. For (11), the computational time complexity is $O(n_t d^2 + d^3)$ if we solve it using the least angle regression algorithm [38]. The optimization of (11) must be performed for all tasks; therefore, the total time complexity of solving $\{C_t\}_{t=1}^T$ is $O(Nd^2 + d^3)$. Suppose that Algorithm 1 runs for M iterations; then, the final time complexity is $M \times O(NTd^2 + d^3)$. From this time complexity analysis, we can conclude that the time complexity of our proposed algorithm is independent of the dimensionality of the original data. We can control the time complexity by varying the number of selected top eigenfunctions. Consequently, our proposed method is more suitable than other methods for high-dimensional multitask learning problems.

IV. Theoretical Analysis

In this section, we present a theoretical analysis to demonstrate how the proposed method can better learn shared information. Since $C_0^\top \Phi$ represents the commonly shared parameters and $C_t^\top \Phi$ represents the specific parameters for the tth task, we will focus on analyzing the learning properties for $C_0^\top \Phi$. Specifically, we show that the proposed method is

Algorithm 1 Iterative Optimization Algorithm for EMTL

Input: Input data sets $D_t, t = 1, 2, \ldots, T$; initialize the variables C_0 and $\{C_t\}_{t=1}^{T}$ and the trade-off parameters γ and β.
Output: Shared coefficients C_0 and coefficients for each specific task $\{C_t\}_{t=1}^{T}$.
1: Explicitly compute the eigenpairs $\{\lambda_{ti}, \Phi_{ti}\}_{i=1}^{d_t}$ for all tasks using formulation (5)
2: **while** (not converged) **do**
3: Compute $C_0 = \text{argmin}\|Y - \Phi(X)^\top (I_0 \times C_0 + \hat{C})\|_2^2 + \beta\|\Phi^\top C_0\|_2^2$.
4: **for** t =1 to T **do**
5: Compute $C_t = \text{argmin}\ \|\Phi(X_t)^T (C_t + C_0) - Y_t\|_2^2 + \gamma\,\|C_t\|_1$.
6: **end for**
7: **end while**

argument stable [39] for learning $C_0^\top \Phi$ and that the generalization bound for learning $C_0^\top \Phi$ has a convergence rate of $O(1/\sqrt{nT})$, which enables the proposed learning algorithm to generalize quickly and accurately from a small training sample when the number of tasks is large.

We first introduce the notion of argument stability [39], which measures the impact of changing a single training example on the function selected by the learning algorithm. Intuitively, the learning algorithm is stable if its outputs are not sensitive to subtle changes in the input, or in other words, if the outputs do not change much when the changes in the input training samples are small.

Definition 1 (Uniform Argument Stability [39]): Let $D = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ be a training set consisting of n pairs of independent random variables. Let $C_{0,D}^\top \Phi_D$ denote the output of a learning algorithm obtained by exploiting the input training set D. We say that the learning algorithm is $\alpha(n)$-uniformly argument stable if for all $i \in \{1, \ldots, n\}$, it holds that

$$\left\| C_{0,D}^\top \Phi_D - C_{0,D^i}^\top \Phi_{D^i} \right\| \le \alpha(n) \tag{13}$$

where $\alpha(n) \in \mathbb{R}_+$ and $D^i = \{(X_1, Y_1), \ldots, (X_{i-1}, Y_{i-1}), (X_i', Y_i'), (X_{i+1}, Y_{i+1}), \ldots, (X_n, Y_n)\}$ represents the training set D with the ith sample replaced with an independent copy of (X_i', Y_i').

We show that the algorithm for learning the commonly shared parameter C_0 in (6) is uniformly argument stable.

Theorem 1: If we assume that the variables $\|C\|_2$, $\|\Phi(x)\|_2$, $K(x, x)$, and Y are upper bounded by $\wedge_C$, $\wedge_\phi$, $\wedge_K^2$, and $\wedge_Y$, respectively, then the algorithm for learning C_0 in (6) is uniformly argument stable, that is,

$$\begin{aligned}&\left\| C_{0,D}^\top \Phi_D - C_{0,D^i}^\top \Phi_{D^i} \right\| \\ &\quad \le \frac{2(2\wedge_C \wedge_\Phi + \wedge_Y)\wedge_K}{\beta \min\{n_1, \ldots, n_T\} T} + \sqrt{\frac{4(2\wedge_C \wedge_\Phi + \wedge_Y)\wedge_C \wedge_\Phi}{\beta \min\{n_1, \ldots, n_T\} T}}.\end{aligned} \tag{14}$$

To prove Theorem 1, we first introduce the notion of the Bregman divergence.

Definition 2 (Bregman Divergence): Let f be a convex function. For any s and t in its domain, the Bregman divergence is defined as

$$B_f(s\|t) = f(s) - f(t) - \langle s - t, \nabla f(t)\rangle \tag{15}$$

where $\nabla f(t)$ denotes the gradient of f at t.

It is easy to prove that the Bregman divergence is additive and nonnegative. For example, if $f = f_1 + f_2$ and both f_1 and f_2 are convex, then for any s and t in the domain, we have

$$B_f(s\|t) = B_{f_1}(s\|t) + B_{f_2}(s\|t) \tag{16}$$

and

$$B_f(s\|t) \ge 0. \tag{17}$$

We are now ready to prove Theorem 1. Let

$$\begin{aligned}&L_D\big(C_0^\top \Phi\big) \\ &\quad = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{n_t} \sum_{i=1}^{n_t} \left(\sum_{j=1}^{d} (C_{tj} + C_{0j})\Phi_j(X_{ti}) - Y_{ti} \right)^2\end{aligned} \tag{18}$$

and

$$P_D\big(C_0^\top \Phi\big) = \beta \left\| C_0^\top \Phi \right\|_2^2 + \gamma\,\|C\|_1. \tag{19}$$

The objective in (6) can be written as

$$O_D\big(C_0^\top \Phi\big) = L_D\big(C_0^\top \Phi\big) + P_D\big(C_0^\top \Phi\big). \tag{20}$$

Note that O_D and P_D are both convex with respect to $C_0^\top \Phi$. Using the nonnegative and additive properties of the Bregman divergence, we have

$$\begin{aligned}&B_{O_D}\big(C_{0,D^i}^\top \Phi_{D^i} \big\| C_{0,D}^\top \Phi_D\big) + B_{O_{D^i}}\big(C_{0,D}^\top \Phi_D \big\| C_{0,D^i}^\top \Phi_{D^i}\big) \\ &\quad \ge B_{P_D}\big(C_{0,D^i}^\top \Phi_{D^i} \big\| C_{0,D}^\top \Phi_D\big) + B_{P_{D^i}}\big(C_{0,D}^\top \Phi_D \big\| C_{0,D^i}^\top \Phi_{D^i}\big).\end{aligned}$$

We attempt to upper bound $B_{O_D}(C_{0,D^i}^\top \Phi_{D^i} \| C_{0,D}^\top \Phi_D) + B_{O_{D^i}}(C_{0,D}^\top \Phi_D \| C_{0,D^i}^\top \Phi_{D^i})$ and lower bound $B_{P_D}(C_{0,D^i}^\top \Phi_{D^i} \| C_{0,D}^\top \Phi_D) + B_{P_{D^i}}(C_{0,D}^\top \Phi_D \| C_{0,D^i}^\top \Phi_{D^i})$. Specifically, let $P2_D(C_0^\top \Phi) = \beta \|C_0^\top \Phi\|_2^2$; then, we have

$$\begin{aligned}&B_{P_D}\big(C_{0,D^i}^\top \Phi_{D^i} \big\| C_{0,D}^\top \Phi_D\big) + B_{P_{D^i}}\big(C_{0,D}^\top \Phi_D \big\| C_{0,D^i}^\top \Phi_{D^i}\big) \\ &\ge B_{P2_D}\big(C_{0,D^i}^\top \Phi_{D^i} \big\| C_{0,D}^\top \Phi_D\big) + B_{P2_{D^i}}\big(C_{0,D}^\top \Phi_D \big\| C_{0,D^i}^\top \Phi_{D^i}\big) \\ &= \beta \left\| C_{0,D^i}^\top \Phi_{D^i} \right\|_2^2 - \beta \left\| C_{0,D}^\top \Phi_D \right\|_2^2 \\ &\quad - \left\langle C_{0,D^i}^\top \Phi_{D^i} - C_{0,D}^\top \Phi_D, 2\beta C_{0,D}^\top \Phi_D \right\rangle + \beta \left\| C_{0,D}^\top \Phi_D \right\|_2^2 \\ &\quad - \beta \left\| C_{0,D^i}^\top \Phi_{D^i} \right\|_2^2 - \left\langle C_{0,D}^\top \Phi_D - C_{0,D^i}^\top \Phi_{D^i}, 2\beta C_{0,D^i}^\top \Phi_{D^i} \right\rangle \\ &= 2\beta \left\| C_{0,D^i}^\top \Phi_{D^i} - C_{0,D}^\top \Phi_D \right\|_2^2.\end{aligned}$$

We further upper bound $B_{O_D}(C_{0,D^i}^\top \Phi_{D^i} \| C_{0,D}^\top \Phi_D) + B_{O_{D^i}}(C_{0,D}^\top \Phi_D \| C_{0,D^i}^\top \Phi_{D^i})$

$$\begin{aligned}
&B_{O_D}\left(C_{0,D^i}^\top \Phi_{D^i} \middle\| C_{0,D}^\top \Phi_D\right) \\
&\quad + B_{O_{D^i}}\left(C_{0,D}^\top \Phi_D \middle\| C_{0,D^i}^\top \Phi_{D^i}\right) \\
&\quad = O_D\left(C_{0,D^i}^\top \Phi_{D^i}\right) - O_D\left(C_{0,D}^\top \Phi_D\right) \\
&\quad\quad - \left\langle C_{0,D^i}^\top \Phi_{D^i} - C_{0,D}^\top \Phi_D, \nabla O_D\left(C_{0,D}^\top \Phi_D\right)\right\rangle \\
&\quad\quad + O_{D^i}\left(C_{0,D}^\top \Phi_D\right) - O_{D^i}\left(C_{0,D^i}^\top \Phi_{D^i}\right) \\
&\quad\quad - \left\langle C_{0,D}^\top \Phi_D - C_{0,D^i}^\top \Phi_{D^i}, \nabla O_{D^i}\left(C_{0,D^i}^\top \Phi_{D^i}\right)\right\rangle \\
&\quad = O_D\left(C_{0,D^i}^\top \Phi_{D^i}\right) - O_D\left(C_{0,D}^\top \Phi_D\right) + O_{D^i}\left(C_{0,D}^\top \Phi_D\right) \\
&\quad\quad - O_{D^i}\left(C_{0,D^i}^\top \Phi_{D^i}\right). \qquad (21)
\end{aligned}$$

The second equality holds because $C_{0,D}^\top \Phi_D$ and $C_{0,D^i}^\top \Phi_{D^i}$ are the minimizers of $O_D(C_{0,D}^\top \Phi_D)$ and $O_{D^i}(C_{0,D^i}^\top \Phi_{D^i})$, respectively, and $\nabla O_D(C_{0,D}^\top \Phi_D) = \nabla O_{D^i}(C_{0,D^i}^\top \Phi_{D^i}) = 0$. Thus

$$\begin{aligned}
&B_{O_D}\left(C_{0,D^i}^\top \Phi_{D^i} \middle\| C_{0,D}^\top \Phi_D\right) \\
&\quad + B_{O_{D^i}}\left(C_{0,D}^\top \Phi_D \middle\| C_{0,D^i}^\top \Phi_{D^i}\right) \\
&\quad = O_{D^i}\left(C_{0,D}^\top \Phi_D\right) - O_D\left(C_{0,D}^\top \Phi_D\right) + O_D\left(C_{0,D^i}^\top \Phi_{D^i}\right) \\
&\quad\quad - O_{D^i}\left(C_{0,D^i}^\top \Phi_{D^i}\right) \\
&\quad = \frac{1}{Tn_t}\left(\sum_{j=1}^{d}(C_{tj} + C_{0j,D})\Phi_{j,D}\left(X'_{ti}\right) - Y'_{ti}\right)^2 \\
&\quad\quad - \frac{1}{Tn_t}\left(\sum_{j=1}^{d}(C_{tj} + C_{0j,D})\Phi_{j,D}(X_{ti}) - Y_{ti}\right)^2 \\
&\quad\quad + \frac{1}{Tn_t}\left(\sum_{j=1}^{d}(C_{tj} + C_{0j,D^i})\Phi'_{j,D^i}(X_{ti}) - Y_{ti}\right)^2 \\
&\quad\quad - \frac{1}{Tn_t}\left(\sum_{j=1}^{d}(C_{tj} + C_{0j,D^i})\Phi'_{j,D^i}\left(X'_{ti}\right) - Y'_{ti}\right)^2 \\
&\quad \le \frac{2(2\wedge_C \wedge_\Phi + \wedge_Y)}{Tn_t}\left|\sum_{j=1}^{d}(C_{tj} + C_{0j,D})\Phi_{j,D}\left(X'_{ti}\right)\right. \\
&\quad\quad \left. - \sum_{j=1}^{d}(C_{tj} + C_{0j,D^i})\Phi'_{j,D^i}\left(X'_{ti}\right)\right| + \frac{2(2\wedge_C \wedge_\Phi + \wedge_Y)}{Tn_t} \\
&\quad\quad \times \left|\sum_{j=1}^{d}(C_{tj} + C_{0j,D})\Phi_{j,D}(X_{ti})\right. \\
&\quad\quad\quad \left. - \sum_{j=1}^{d}(C_{tj} + C_{0j,D^i})\Phi'_{j,D^i}(X_{ti})\right| \\
&\quad \le \frac{4(2\wedge_C \wedge_\Phi + \wedge_Y)\wedge_K}{Tn_t} \\
&\quad\quad \times \left\|\sum_{j=1}^{d}(C_{0j,D}\Phi_D - C_{0j,D^i}\Phi_{D^i})\right\|_2 \\
&\quad\quad + \frac{8(2\wedge_C \wedge_\Phi + \wedge_Y)\wedge_C \wedge_\Phi}{Tn_t} \\
&\quad \le \frac{4(2\wedge_C \wedge_\Phi + \wedge_Y)\wedge_K}{Tn_t}\left\|C_{0,D}^\top \Phi_{j,D} - C_{0,D^i}^\top \Phi_{j,D^i}\right\|_2 \\
&\quad\quad + \frac{8(2\wedge_C \wedge_\Phi + \wedge_Y)\wedge_C \wedge_\Phi}{Tn_t}. \qquad (22)
\end{aligned}$$

Combining (21) and (22), we obtain

$$\begin{aligned}
&2\beta\left\|C_{0,D}^\top \Phi_{j,D} - C_{0,D^i}^\top \Phi_{j,D^i}\right\|_2^2 \\
&\quad \le \frac{4(2\wedge_C \wedge_\Phi + \wedge_Y)\wedge_K}{Tn_t}\left\|C_{0,D}^\top \Phi_{j,D} - C_{0,D^i}^\top \Phi_{j,D^i}\right\|_2 \\
&\quad\quad + \frac{8(2\wedge_C \wedge_\Phi + \wedge_Y)\wedge_C \wedge_\Phi}{Tn_t}. \qquad (23)
\end{aligned}$$

We then have

$$\begin{aligned}
&\left\|C_{0,D}^\top \Phi_{j,D} - C_{0,D^i}^\top \Phi_{j,D^i}\right\|_2 \\
&\quad \le \frac{2(2\wedge_C \wedge_\Phi + \wedge_Y)\wedge_K}{\beta T n_t} + \sqrt{\frac{4(2\wedge_C \wedge_\Phi + \wedge_Y)\wedge_C \wedge_\Phi}{\beta T n_t}}. \qquad (24)
\end{aligned}$$

■

Theorem 1 implies that when the training set is changed by one example, the change in the output $C_0^\top \Phi$ will vanish as the training set size n or the number of tasks T goes to infinity. This is the property of algorithmic stability, which can be used to derive the generalization bound [40]. By employing the result of Liu *et al.* [39] (Theorem 2 therein), we can easily derive a deformed generalization bound for the proposed algorithm with respect to the parameter $C_{0,D}^\top \Phi_D$. This deformed generalization upper bound will have a convergence rate of $O(1/\sqrt{nT})$ with respect to the training set size n and the number of tasks T, which implies that with an increase in either the training set size n or the number of tasks T, the generalization error will decrease. Specifically, in the proof of Theorem 1, we can see that the convergence rate of $O(1/\sqrt{nT})$ is introduced because of C_t. If $C_t = 0$, then the generalization bound for learning the commonly shared parameter will converge faster, with a rate of $O(1/nT)$. The advantage of multitask learning has thus been demonstrated for learning $C_{0,D}^\top \Phi_D$. The empirical validations presented in Section V also support these theoretical results.

The generalization error measures the difference between the training and testing errors. A small generalization error bound does not imply a small test error. A small testing error should additionally be based on a small training error. The choice of Φ in this paper also essentially guarantees a small training error in (6) because it guarantees a small reconstruction error in the feature space. Then, (7) functions similar to a representer theorem but with a clear structure of commonly shared parameters in the multitask learning setting.

V. Experiments

In this section, we present and analyze experimental results obtained on three benchmark multitask learning data sets to demonstrate the effectiveness of our proposed multitask

TABLE I
COMPARISON OF RESULTS OBTAINED ON THE SCHOOL DATA SET USING THE MSE

Training Ratio	KRR	FSTL	FSTL_M	CMTL	CMTS	MTRL	EMTL
10%	11.68 ± 0.18	11.81 ± 0.17	12.03 ± 0.19	11.66 ± 0.17	11.81 ± 0.14	11.39 ± 0.14	$\mathbf{10.36 \pm 0.13}$
20%	11.19 ± 0.13	11.31 ± 0.16	11.20 ± 0.14	10.89 ± 0.13	10.90 ± 0.12	10.78 ± 0.13	$\mathbf{10.29 \pm 0.12}$
30%	10.89 ± 0.14	11.01 ± 0.18	10.85 ± 0.15	10.67 ± 0.14	10.54 ± 0.13	10.59 ± 0.14	$\mathbf{10.19 \pm 0.13}$

TABLE II
COMPARISON OF RUNNING TIMES (SECONDS) ON THE SCHOOL DATA SET

Training ratio	KRR	FSTL	FSTL_M	CMTL	CMTS	MTRL	EMTL
10%	0.023 ± 0.002	0.462 ± 0.003	3.892 ± 0.193	4.203 ± 0.061	1.003 ± 0.057	59.871 ± 0.760	3.875 ± 0.038
20%	0.035 ± 0.003	0.599 ± 0.008	5.670 ± 0.198	30.174 ± 0.137	6.273 ± 0.149	134.330 ± 1.061	15.520 ± 0.579
30%	0.048 ± 0.002	0.675 ± 0.005	8.167 ± 0.303	96.401 ± 0.386	17.450 ± 0.265	224.959 ± 4.611	22.695 ± 0.039

learning algorithm. The three data sets used in our experiments are the School data set,[1] the Computer data set [41], and the Isolet data set.[2] These three data sets have been widely used for evaluating the effectiveness of multitask learning in various works [1], [7], [9], [11], [25]. The experimental results of our proposed EMTL method are compared with those of three single-task learning algorithms and several state-of-the-art multitask learning methods. The first two single-task learning methods are kernel ridge regression (KRR) and a single-task learning method based on explicitly learned eigenfunctions feature-based single task learning (FSTL) [15]. The third single-task learning method is FSTL_multiple (FSTL_M), which is based on FSTL but considers the eigenfunctions learned from all tasks. The multitask learning methods are CMTL [7], multitask relationship learning (MTRL) [33], and the convex learning of multiple tasks and their structure (CMTS) [34]. These multitask learning methods are representative methods that have achieved promising performance on various multitask learning data sets. Consequently, comparisons with these methods can sufficiently demonstrate the effectiveness of our proposed method.

A. School Data Set

The School data set is one of the most widely used multitask learning data sets. It was collected from the Inner London Education Authority. This data set consists of 139 tasks, each of which corresponds to the prediction of examination scores at one secondary school. The provided data include the examination scores of 15 362 students from 139 secondary schools in 1985, 1986, and 1987. Each sample includes four school-dependent features, three student-dependent features, and the year of the examination. The four school-dependent features are the percentage of students eligible for free school meals, the percentage of students in voltage regulator (VR) band one, the school denomination, and the school's gender composition. The three student-dependent features are gender, ethnic group, and VR band. To ensure fair comparisons with the other methods, we considered 27-dimensional binary variables for each sample, following the same setup as in previous multitask learning studies [7], [33], [42].

To evaluate the effectiveness of our proposed multitask learning method, we randomly selected 10%, 20%, or 30% of the data for each task as the training data. The remaining samples were split into the validation set and the test set. To avoid statistical outliers, we repeated this selection process $10\times$ for all methods, and we reported the mean performance and the standard deviation across the 10 trials. All methods utilized an radial basis function (RBF) kernel, and the best parameters for different tasks were selected based on the validation set. The number of top eigenfunctions was also empirically selected based on the validation set. If all nonzero eigenvalues and their corresponding eigenvectors were used, the best performance would be achieved. However, this approach would also increase the computation time of our proposed algorithm. We therefore attempted to reduce the number of eigenfunctions used for each task in our proposed method while guaranteeing its performance. For the School data set, the top 10 eigenvectors were used in our method. We evaluated the performances of all regression methods using mean squared error (MSE). The results are shown in Table I.

From the results shown in Table I, we can conclude that all multitask learning methods outperformed two of the single-task learning methods, further demonstrating the effectiveness of multitask learning compared with single-task learning. Notably, our proposed EMTL algorithm consistently performed the best as the training ratio increased from 10% to 30%. The FSTL_M method, which used the eigenfunctions from all tasks without considering how the eigenfunctions were shared among tasks, performed the worst. This poor performance might be caused by the introduction of noise from other tasks. We can also conclude that the proposed method effectively measures the relatedness among tasks, particularly when the number of training samples is insufficient. In single-task learning, sufficient information about the distribution of the training data cannot be obtained when limited data are provided. By contrast, our proposed method extracts more information by considering the relatedness among different tasks. In addition, our proposed method achieves better performance using 10% of the training data than other methods achieve using 30% of the training data.

To better illustrate the computational efficiencies, we compared the running times of all methods on a PC with a 4.0-GHz Intel Core CPU and 16 GB of memory. The results

[1] http://ttic.uchicago.edu/~argyriou/code/

[2] https://archive.ics.uci.edu/ml/datasets/ISOLET

TABLE III
COMPARISON OF RESULTS OBTAINED ON THE COMPUTER DATA SET USING THE MSE

Method	MSE
KRR	2.03
FSTL	1.98
FSTL_M	1.98
CMTL	**1.85**
CMTS	1.88
MTRL	1.92
EMTL	**1.84**

TABLE IV
COMPARISON OF RUNNING TIMES (SECONDS) ON THE COMPUTER DATA SET

Method	Running Time (s)
KRR	0.1080 ± 0.0181
FSTL	0.9018 ± 0.0234
FSTL_M	0.9018 ± 0.0234
CMTL	17.2972 ± 0.1657
CMTS	1.0830 ± 0.0592
MTRL	19.4653 ± 0.1030
EMTL	1.4678 ± 0.1536

are shown in Table II. Note that the running time is the total time required to solve all tasks and that the experiments were repeated $10\times$. The mean time and standard deviation are reported. We can conclude that multitask learning algorithms require much more time than single-task learning algorithms. Our proposed EMTL method has a much lower time cost than CMTL and MTRL. CMTS is the most efficient method. This result is mainly because CMTS has closed-form solutions for each step. However, the cost time of our EMTL method is comparable to that of CMTS as the number of training samples increases, and our proposed EMTL method achieves much better performance.

B. Computer Data Set

In this section, we report on the experiments conducted on the Computer data set, which contains people's ratings of computer products [41]. This data set includes the results of a survey of 180 people who rated their tendency to buy 20 different computers. Each person is regarded as one task, following the same experimental setup as in previous works. Each computer is represented by a 13-dimensional binary feature vector, which includes telephone hotline availability, amount of memory, screen size, CPU speed, hard disk, CD-ROM/multimedia, cache, color, availability, warranty, software, guarantee, and price. The output is an integer rating that scales from 0 to 10. To facilitate computation, one dimension representing the bias term was added. Following the same setup used in previous works, the first eight examples from each person were used as training data, and the last four examples were used as test data. We chose the top eight eigenfunctions from each task to approximate the final regression function. We used the same evaluation measurement, the MSE, as used on the School data set to evaluate the performance. All methods used an RBF kernel, and the best parameters were selected based on the validation set.

The results are reported in Table III, based on which we conclude that the multitask learning methods outperform the single-task learning methods. Our proposed EMTL method and CMTL exhibit the best performance, with similar MSE values. In addition, we show the running times of all methods on the Computer data set in Table IV, and conclusions similar to those obtained on the School data set can be drawn.

We also illustrate the learned shared coefficients C_0 and task-specific coefficients $C = [C_1, C_2, \ldots, C_T]$. For the Computer data set, the training data for each task are the same. Consequently, the explicitly learned eigenfunctions for each task are also the same. This is the reason why FSTL and FSTL_M achieve the same MSE and the same computation time. We only have to learn the coefficients for all tasks from the selected top eight eigenfunctions. The absolute values of the learned coefficients are shown in Figs. 2 and 3.

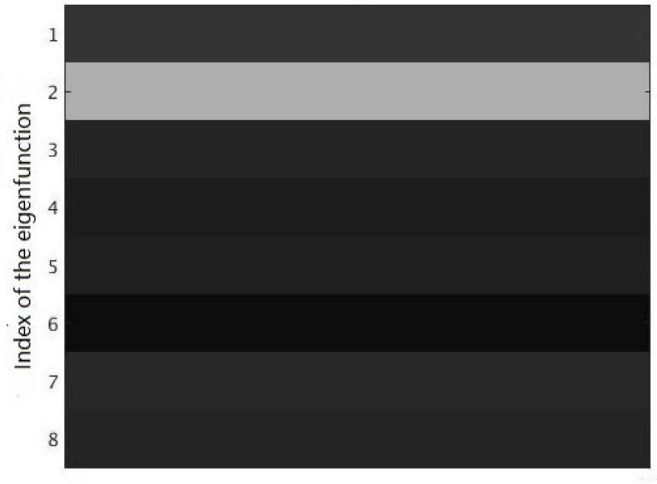


Fig. 2. Illustration of the absolute values of the eight shared coefficients (C_0) learned from the Computer data set. Black areas denote zero values, and the value increases as the color changes from black to white. Only the coefficient of the 6th eigenfunction is close to zero, which means that all tasks share seven eigenfunctions, leading to high relatedness.

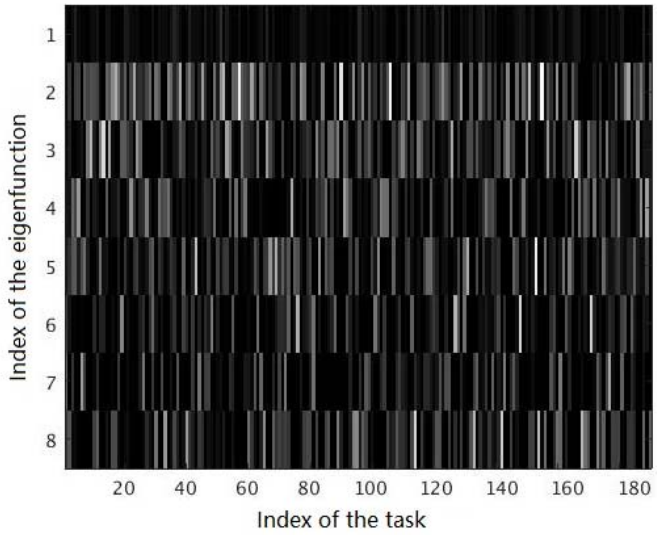


Fig. 3. Illustration of the absolute values of the nonshared coefficients for individual tasks learned from the Computer data set. The coefficients of the task-specific eigenfunctions are sparse.

TABLE V
COMPARISON OF RESULTS OBTAINED ON THE ISOLET DATA SET USING THE MSE

Training ratio	KRR	FSTL	FSTL_M	CMTL	CMTS	MTRL	EMTL
10%	5.03 ± 0.19	5.17 ± 0.15	5.14 ± 0.07	5.47 ± 0.19	4.61 ± 0.13	4.17 ± 0.05	$\mathbf{3.98 \pm 0.04}$
20%	4.40 ± 0.06	4.71 ± 0.06	4.73 ± 0.11	4.84 ± 0.10	4.07 ± 0.06	3.96 ± 0.02	$\mathbf{3.80 \pm 0.02}$
30%	4.14 ± 0.05	4.58 ± 0.06	4.52 ± 0.06	4.60 ± 0.07	3.84 ± 0.04	3.87 ± 0.05	$\mathbf{3.70 \pm 0.08}$

TABLE VI
COMPARISON OF RUNNING TIMES (SECONDS) ON THE ISOLET DATA SET

Training ratio	KRR	FSTL	FSTL_M	CMTL	CMTS	MTRL	EMTL
10%	0.005 ± 0.001	0.111 ± 0.021	0.163 ± 0.011	1.415 ± 0.019	0.169 ± 0.027	11.245 ± 0.060	0.721 ± 0.033
20%	0.0162 ± 0.003	0.281 ± 0.025	0.506 ± 0.021	4.847 ± 0.191	0.880 ± 0.020	24.080 ± 0.099	3.4175 ± 0.096
30%	0.054 ± 0.006	0.6001 ± 0.027	0.628 ± 0.013	13.90 ± 0.079	1.849 ± 0.033	38.854 ± 0.068	7.925 ± 0.170

TABLE VII
COMPARISON OF RESULTS OBTAINED ON FIVE ISOLET TASKS USING THE MSE

	KRR	FSTL	FSTL_M	CMTL	CMTS	MTRL	EMTL
Task 1	3.83 ± 0.12	4.22 ± 0.10	4.24 ± 0.12	4.26 ± 0.12	$\mathbf{3.65 \pm 0.13}$	3.77 ± 0.12	3.67 ± 0.17
Task 2	3.92 ± 0.13	4.50 ± 0.18	4.40 ± 0.15	4.45 ± 0.20	$\mathbf{3.71 \pm 0.10}$	3.92 ± 0.07	3.75 ± 0.10
Task 3	4.36 ± 0.14	4.68 ± 0.14	4.67 ± 0.18	4.87 ± 0.16	3.98 ± 0.10	3.93 ± 0.08	$\mathbf{3.70 \pm 0.07}$
Task 4	4.41 ± 0.09	4.81 ± 0.12	4.70 ± 0.15	4.83 ± 0.10	4.01 ± 0.07	3.89 ± 0.05	$\mathbf{3.74 \pm 0.07}$
Task 5	4.15 ± 0.19	4.68 ± 0.15	4.56 ± 0.29	4.59 ± 0.23	3.85 ± 0.17	3.81 ± 0.10	$\mathbf{3.68 \pm 0.15}$

Black areas denote zero values, and the value increases as the color changes from black to white. From Fig. 2, we find that the coefficient of the 6th eigenfunction is close to zero, which means that seven of the eight eigenfunctions are shared. From Fig. 3, we find that the coefficients of the task-specific eigenfunctions are sparse. We can conclude that the different tasks in the Computer data set are closely related. The reason the coefficient value for the second eigenfunction appears quite high compared to the rest of the values may be that the second eigenfunction is important to all tasks. This can also be seen in the results in Fig. 3. The task-specific coefficient for the second eigenfunction appears to be large for almost all tasks. Therefore, we can conclude that the second eigenfunction is the most important of the eigenfunctions to most of the tasks and should be shared among the tasks with a large coefficient.

C. Isolet Data Set

We report the results of testing the performance of our proposed multitask learning method on the Isolet data set in this section. This data set concerns the pronunciation of the letters in the alphabet by 150 speakers. Each speaker spoke each letter twice; thus, 52 examples were collected from each speaker. The speakers are grouped into five groups: Isolet-1, Isolet-2, Isolet-3, Isolet-4, and Isolet-5. Thus, there are five tasks, one corresponding to each of these five groups, which contain 1560, 1560, 1560, 1558, and 1559 samples, respectively. Each letter is related to a label (1–26), and we treat all tasks as regression problems, following [9]. We randomly selected 10%, 20%, or 30% of the data as the training set, and the rest of the data was split into the validation set and the test set. To avoid statistical outliers in the experimental results, we repeated all experiments five times, and we reported the mean performance with the standard deviation. An RBF kernel was used in all methods, and the best parameters were selected based on the validation set. We again used the MSE to evaluate the performance of each method.

Based on the results shown in Table V, we can again conclude that all of the multitask learning methods except CMTL outperform the single-task learning methods. This result is because considering the sharing of common features across tasks is not an adequate means of measuring the relatedness among tasks in the Isolet data set. In addition, the performance of FSTL_M is similar to that of FSTL. This finding indicates that FSTL_M cannot learn additional information about the data from the eigenfunctions of other tasks without considering the shared eigenfunctions. Our proposed EMTL method consistently achieved the best performance across the different training ratios. These findings demonstrate that measuring task relatedness through eigenfunctions enables better exploration of the information contained in this data set that can be achieved with the other methods. In addition, we present the running times of all methods in Table VI, using the same settings as for the School data set. We can again obtain conclusions similar to those found in the School data set and the Computer data set.

In Table VII, we present additional experimental results to enable an analysis of the performance improvement on each task when our proposed method is used. This experiment was conducted with 30% of the data as the training set and the remaining data split into the validation set and the test set. All experiments were repeated five times to avoid statistical outliers, and the best parameters were selected based on the validation set. Based on the performance of the single-task learning methods, we can conclude that the difficulty varies among the different tasks and that task 4 is the most difficult one. Compared with the single-task learning methods, all of the multitask learning methods except CMTL showed

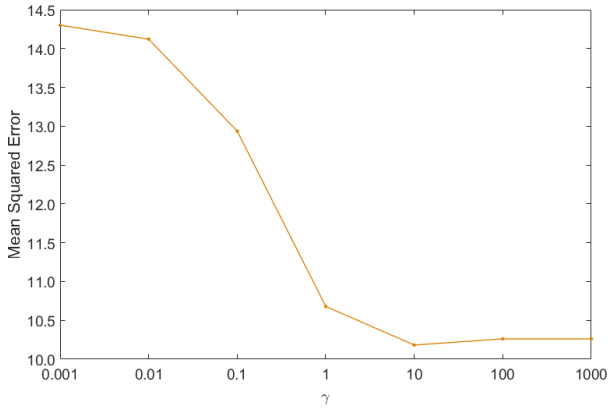


Fig. 4. Sensitivity analysis of EMTL with respect to the parameter γ.

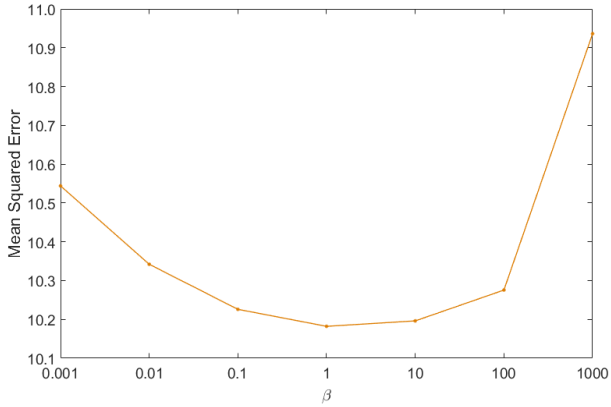


Fig. 5. Sensitivity analysis of EMTL with respect to the parameter β.

improved performance on all tasks. For the more difficult tasks, limited information about the distribution of the data was available from their training data. However, multitask learning methods can extract more shared information relevant to these tasks, leading to performance improvements. Our proposed multitask learning method effectively measures the relatedness among tasks and significantly improves the performance on all tasks, particularly difficult ones.

D. Sensitivity Analysis of EMTL

In this section, we report experiments conducted to analyze the sensitivity of our proposed EMTL method to the regularization parameters β and γ. All experiments were conducted on the School data set with a training ratio of 10%.

All parameters, including β and γ, were selected to achieve the best performance on the validation set for all experiments in this paper. We conducted a grid search of β and γ within the set $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$. For the School data set, we used an RBF kernel with a bandwidth of 10. As seen from experiments on the validation set, the best performance was achieved with parameter values of $\gamma = 100$ and $\beta = 1$. Consequently, we analyzed the sensitivity of EMTL to γ with a fixed value of $\beta = 1$ and analyzed the sensitivity of EMTL to β with a fixed value of $\gamma = 100$. The results are shown in Figs. 4 and 5. From these results, we can conclude that the performances in the experiments were better for γ values larger than 10 and β values near 1. These findings indicate that the coefficients $\{C_t\}_{t=1}^{T}$ of the task-specific eigenfunctions tend to be smaller than the coefficients C_0 of the shared eigenfunctions. Therefore, the shared eigenfunctions play a more important role, and for each task, additional valuable information can be obtained from the training data associated with other tasks. The performances on all tasks should improve in such a situation.

TABLE VIII

COMPARISONS BETWEEN OUR PROPOSED METHOD AND THE SECOND BEST METHOD IN TERMS OF P-VALUES

Training ratio	School Dataset	Isolet Dataset
10%	1.25×10^{-13}	2.49×10^{-4}
20%	3.73×10^{-9}	2.42×10^{-4}
30%	1.08×10^{-5}	9.35×10^{-4}

E. Analysis of P-Values

In this section, we present an analysis of p-values obtained using the t-test to show that our proposed method is statistically significantly better than the next best method. We performed t-tests only on the School data set and the Isolet data set because the training and test samples in the Computer data set are fixed.

From Table I, we can see that on the School data set, MTRL performs the second best when the training ratio is 10% or 20% and that CMTS performs the second-best when the training ratio is 30%. We therefore compare our EMTL method with MTRL for training ratios of 10% and 20% and with CMTS for a training ratio of 30%. Similarly, we compare our EMTL method with MTRL for training ratios of 10% and 20% and with CMTS for a training ratio of 30% on the Isolet data set. The results are shown in Table VIII. We can conclude that our proposed method performs significantly better than the second-best methods, as the p-values are substantially smaller than 0.05 for all training ratios on both data sets.

VI. CONCLUSION

In this paper, we propose a method for learning multiple tasks from a new perspective. Unlike previous multitask learning methods, in which task relatedness is measured through parameter sharing or feature sharing, our proposed multitask learning method learns task relationships by considering a shared set of eigenfunctions. These eigenfunctions can be explicitly learned and easily extended to any kernel type. Consequently, we only have to learn a set of shared coefficients for all tasks and a set of task-specific coefficients for each task. The objective function can be optimized by means of an iterative algorithm, which divides the optimization problem into two subproblems: L2-norm regularized regression and L1-norm regularized regression. We also present a detailed theoretical analysis to demonstrate that our proposed algorithm is uniformly argument stable and that the convergence rate of the generalization upper bound is related to the number of training samples and the number of tasks. The findings

imply that learning multiple tasks simultaneously can help improve performance. Various experiments were conducted on several multitask learning data sets, and the experimental results demonstrate the effectiveness of our proposed method.

REFERENCES

[1] S. Parameswaran and K. Q. Weinberger, "Large margin multi-task metric learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1867–1875.

[2] L. Ma, X. Yang, and D. Tao, "Person re-identification over camera networks using multi-task distance metric learning," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3656–3670, Aug. 2014.

[3] Y. Luo, Y. Wen, and D. Tao, "Heterogeneous multitask metric learning across multiple domains," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 154–167, Sep. 2018.

[4] X. Wang, C. Zhang, and Z. Zhang, "Boosted multi-task learning for face verification with applications to Web image and video search," in *Proc. Comput. Vis. Pattern Recognit.*, 2009, pp. 142–149.

[5] D. Zhang and D. Shen, "Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease," *NeuroImage*, vol. 59, no. 2, pp. 895–907, 2012.

[6] L. Nie, L. Zhang, L. Meng, X. Song, X. Chang, and X. Li, "Modeling disease progression via multisource multitask learners: A case study with Alzheimer's disease," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1508–1519, Jul. 2017.

[7] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, 2008.

[8] G. Obozinski, B. Taskar, and M. I. Jordan, "Joint covariate selection and joint subspace selection for multiple classification problems," *Statist. Comput.*, vol. 20, no. 2, pp. 231–252, Apr. 2010.

[9] P. Gong, J. Ye, and C.-S. Zhang, "Multi-stage multi-task feature learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1988–1996.

[10] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 339–348.

[11] T. Evgeniou and M. Pontil, "Regularized multi–task learning," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 109–117.

[12] 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, May 2007.

[13] K. Yu, V. Tresp, and A. Schwaighofer, "Learning Gaussian processes from multiple tasks," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 1012–1019.

[14] P. Rai and H. Daumé, III, "Infinite predictor subspace models for multitask learning," in *Proc. AISTATS*, 2010, pp. 613–620.

[15] X. Guo and D.-X. Zhou, "An empirical feature-based learning algorithm producing sparse approximations," *Appl. Comput. Harmon. Anal.*, vol. 32, no. 3, pp. 389–400, 2012.

[16] L. Zwald, G. Blanchard, P. Massart, and R. Vert, "Kernel projection machine: A new tool for pattern recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1649–1656.

[17] M. Ji, T. Yang, B. Lin, R. Jin, and J. Han. (2012). "A simple algorithm for semi-supervised learning with improved generalization error bound." [Online]. Available: https://arxiv.org/abs/1206.6412

[18] I. Takeuchi, T. Hongo, M. Sugiyama, and S. Nakajima, "Parametric task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1358–1366.

[19] X. Mei, Z. Hong, D. Prokhorov, and D. Tao, "Robust multitask multiview tracking in videos," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2874–2890, Nov. 2015.

[20] X. Chang and Y. Yang, "Semisupervised feature analysis by mining correlations among multiple tasks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2294–2305, Oct. 2017.

[21] C. Li, M. Georgiopoulos, and G. C. Anagnostopoulos, "Multitask classification hypothesis space with improved generalization bounds," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1468–1479, Jul. 2015.

[22] Y. Kong, M. Shao, Y. Fu, and K. Li, "Probabilistic low-rank multitask learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 670–680, Mar. 2018.

[23] A. Maurer, M. Pontil, and B. Romera-Paredes, "Sparse coding for multitask and transfer learning," *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 343–351.

[24] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[25] Y. Li, X. Tian, M. Song, and D. Tao, "Multi-task proximal support vector machine," *Pattern Recognit.*, vol. 48, no. 10, pp. 3249–3257, 2015.

[26] T. Jebara, "Multi-task feature and kernel selection for SVMs," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 55–63.

[27] J. Tang, Y. Tian, P. Zhang, and X. Liu, "Multiview privileged support vector machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3463–3477, Aug. 2018.

[28] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 94–108.

[29] W. Liu, T. Mei, Y. Zhang, C. Che, and J. Luo, "Multi-task deep visual-semantic embedding for video thumbnail selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3707–3715.

[30] C. Zhang and Z. Zhang, "Improving multiview face detection with multi-task deep convolutional neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2014, pp. 1036–1041.

[31] T. Liu, D. Tao, M. Song, and S. J. Maybank, "Algorithm-dependent generalization bounds for multi-task learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 227–241, Feb. 2017.

[32] 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, Nov. 2005.

[33] Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *Proc. 26th Conf. Uncertainty Artif. Intell. (UAI)*, 2010, pp. 733–742.

[34] C. Ciliberto, Y. Mroueh, T. Poggio, and L. Rosasco, "Convex learning of multiple tasks and their structure," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1548–1557.

[35] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[36] M. Schmidt, G. Fung, and R. Rosales, "Optimization methods for ℓ_1-regularization," Univ. Brit. Columbia, Vancouver, BC, Canada, Tech. Rep. TR-2009, 2009, vol. 19.

[37] S. 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, Jan. 2011.

[38] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[39] T. Liu, G. Lugosi, G. Neu, and D. Tao, "Algorithmic stability and hypothesis complexity," in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 2159–2167.

[40] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan, "Learnability, stability and uniform convergence," *J. Mach. Learn. Res.*, vol. 11, pp. 2635–2670, Oct. 2010.

[41] P. J. Lenk, W. S. DeSarbo, P. E. Green, and M. R. Young, "Hierarchical bayes conjoint analysis: Recovery of partworth heterogeneity from reduced experimental designs," *Marketing Sci.*, vol. 15, no. 2, pp. 173–191, 1996.

[42] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Mach. Learn. Res.*, vol. 6, pp. 615–637, Apr. 2005.

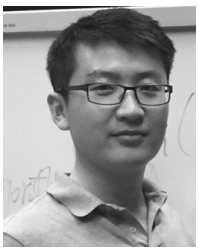
Xinmei Tian (M'13) received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively.

She is currently an Associate Professor with the CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, University of Science and Technology of China. Her current research interests include multimedia information retrieval and machine learning.

Dr. Tian was a recipient of the Excellent Doctoral Dissertation of Chinese Academy of Sciences Award in 2012 and the Nomination of National Excellent Doctoral Dissertation Award in 2013.

Ya Li received the B.S. and Ph.D. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China, in 2013 and 2018, respectively.

His current research interests include machine learning and computer vision.



Tongliang Liu received the B.Eng. degree from the University of Science and Technology of China, Hefei, China, and the Ph.D. degree from the University of Technology Sydney, Ultimo, NSW, Australia.

He is currently a Lecturer with the School of Information Technologies, Faculty of Engineering and Information Technologies, The University of Sydney, Darlington, NSW, where he is a Core Member with the UBTECH Sydney Artificial Intelligence Centre. He has authored or co-authored more than 40 research papers including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, International Conference on Machine Learning, IEEE Conference on Computer Vision and Pattern Recognition, and Knowledge Discovery in Database. His current research interests include statistical learning theory, computer vision, and optimization.

Xinchao Wang received the first class honorable degree from The Hong Kong Polytechnic University, Hong Kong, in 2010, and the Ph.D. degree from the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2015.

He was a Post-Doctoral Fellow with the University of Illinois at Urbana–Champaign, Champaign, IL, USA, with Prof. T. S. Huang. He is currently a Tenure-Track Assistant Professor with the Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ, USA. He has authored or co-authored various venues including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON MEDICAL IMAGING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE Conference on Computer Vision and Pattern Recognition, European Conference on Computer Vision, IEEE International Conference on Computer Vision, Neural Information Processing Systems, and International Conference On Medical Image Computing & Computer Assisted Intervention. His current research interests include computer vision, machine learning, and artificial intelligence.

Dr. Wang is an Associate Editor of the *Journal of Visual Communication and Image Representation*.

Dacheng Tao (F'15) is currently a Professor of computer science and an ARC Laureate Fellow with the School of Information Technologies, Faculty of Engineering and Information Technologies, and the Inaugural Director of the UBTECH Sydney Artificial Intelligence Centre, The University of Sydney, Darlington, NSW, Australia. He mainly applies statistics and mathematics to Artificial Intelligence and Data Science. He has authored or co-authored 1 monograph and more than 200 publications at prestigious journals and prominent conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *International Journal of Computer Vision*, *Journal of Machine Learning Research*, Neural Information Processing Systems, International Conference on Machine Learning, IEEE Conference on Computer Vision and Pattern Recognition, IEEE International Conference on Computer Vision, European Conference on Computer Vision, IEEE International Conference on Data Mining (ICDM), and ACM Special Interest Group on Knowledge Discovery and Data Mining.

Mr. Tao is a Fellow of the Australian Academy of Science, AAAS, IAPR, OSA, and SPIE. He was a recipient of several best paper awards, such as the Best Theory/Algorithm Paper Runner-Up Award in IEEE ICDM in 2007, the Best Student Paper Award in IEEE ICDM in 2013, the Distinguished Paper Award in the 2018 International Joint Conference on Artificial Intelligence, the 2014 ICDM 10-Year Highest-Impact Paper Award, and the 2017 IEEE Signal Processing Society Best Paper Award.