Multi-Task Learning: Multiple Kernels for Multiple Tasks

Wei Wu[a], Hang Li[b], Yunhua Hu[b], Rong Jin[c]

[a]*Department of Probability and Statistics, Peking University, No.5 Yiheyuan Road, Beijing, 100871, P. R. China*
[b]*Microsoft Research Asia, 4F Sigma Building, No. 49 Zhichun Road, Beijing, 100190, P. R. China*
[c]*Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48824, USA*

Abstract

Many kernel based methods for multi-task learning have been proposed, which leverage relations among tasks to enhance the overall learning accuracies. Most of the methods assume that the learning tasks share the same kernel [e.g., 13], which could limit their applications because in practice different tasks may need different kernels. In this paper, we consider utilizing multiple kernels for multiple tasks. The main challenge of introducing multiple kernels into multiple tasks is that functions from different Reproducing Kernel Hilbert Spaces (RKHSs) are not comparable, making it difficult to exploit relations among tasks. This paper addresses the challenge by defining the problem in the *Square Integrable Space* (SIS). Specially, it proposes a kernel based method which makes use of a regularization term defined in the SIS to represent task relations. We prove a new representer theorem for the proposed approach in SIS. We further derive a practical method for solving the learning problem and conduct consistency analysis of the method. We discuss the relations between our method and the existing method by showing the inequality relation between the two regularization terms in the two methods. We also give an SVM based implementation of our method for multi-label classification. Experiments on an artificial example and three real-world data sets show significant improvements of the proposed method over existing methods.

Key words: kernel methods, multi-task learning, multi-label classification, square integrable space, representer theorem, convergence analysis, Support Vector Machines

1. Introduction

We consider the kernel based approaches to multi-task learning in this paper. One commonly adopted strategy is to exploit the relations between tasks (classes) to enhance the performance of learning [cf., 8, 7]. [13], as well as [17], proposed using task relations as regularization terms in kernel methods by assuming that the tasks share the same kernel. We point out that in practice besides employing a single kernel for multiple tasks (SKMT), it is also necessary to employ multiple kernels for multiple tasks (MKMT), depending on applications.

Figure 1 illustrates the importance of MKMT with an artificial example on multi-label classification (special case of multi-task learning). There are three classes, and classification of instances to one class corresponds to one task. The instances (circle points) in the square area on the right side belong to class 1, the instances (diamond points) in the outer circle area and the instances (cross points) in the inner circle area on the left side belong to classes 2 and 3, respectively. The instances (square points) in the middle belong to both classes 1 and 2 (i.e., they

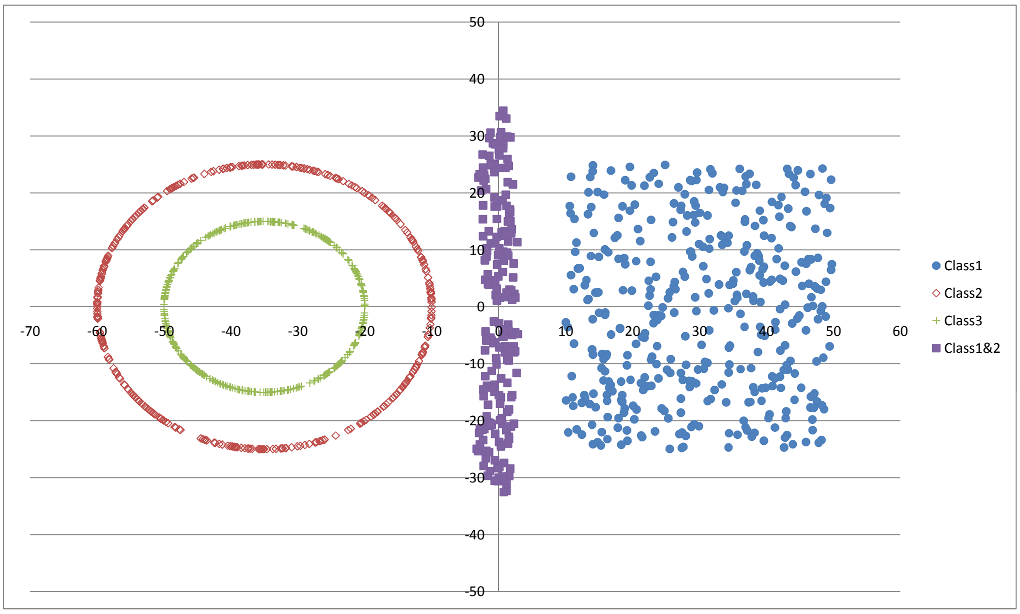

Figure 1: Artificial Data

have multiple labels). The goal of the learning problem is to train classifiers from the training data that can classify new instances as accurately as possible. It is easy to verify that to separate the instances in class 1 from the others, using a linear classifier is sufficient, while to separate the instances in class 2 or the instances in class 3 from the others, it is better to use a nonlinear classifier. Moreover, to handle those instances with double labels, it is more preferable to exploit task relations (e.g., co-occurrence information). We conducted experiments on this artificial data set. The results given in Section 6 show that our method can effectively utilize multiple kernels as well as task relations to outperform the baselines.

In this paper, we propose a general approach for multi-kernel multi-task learning. The major challenge for employing multiple kernels for multiple tasks is that models for different tasks may come from different Reproducing Kernel Hilbert Spaces (RKHSs), making their comparison infeasible. Thus, we formulate multi-kernel multi-task learning in a new space, named the Square Integrable Space (SIS). Since SIS includes RKHSs for different tasks as subspaces, the task relations can be naturally incorporated into a regularization term in the SIS. We then present a new representer theorem which provides the form of solutions to the proposed kernel method. We derive a practical method for solving the learning problem and further prove the convergence of the practical solution to the ideal solution. We discuss the relations between our method and Evgeniou et al's method by showing the inequality relation between the regularization terms in the two methods. We give a specific algorithm of our method based on SVM technique. Experiments of multi-label classification on the artificial example and three real-world data sets show the effectiveness of our approach on handling MKMT problems.

Our contribution in this paper is primarily theoretical, and it consists of three fold, (1) proposal of a method of multi-task learning in Square Integrable Space, particularly for MKMT,

(2) theoretical analysis of the method, (3) practical implementation of the method and empirical verification of its effectiveness.

The rest of the paper is organized as follows: after a survey of related work in Section 2, we introduce the notations used in this paper, and the background knowledge on RKHS and Square Integrable Space (SIS) in Section 3. Then, we propose our approach in Section 4, including showing the new representer theorem, deriving the practical solution, and analyzing the convergence of the practical solution to the ideal solution. We give an implementation of our approach based on SVM technique in Section 5, and empirically verify the effectiveness of our method through several experiments in Section 6. We finally conclude the whole paper with some remarks in Section 7. Proofs of theorems are given in Appendix.

2. Related Work

Multi-task learning aims to perform learning for multiple problems simultaneously in order to achieve better performance for all the problems. It has been verified both theoretically and empirically that it is feasible if one can properly leverage information across the tasks in the learning process, and many methods have been proposed [cf., 8, 7]. One group of methods attempt to use task relations. For example, [14], [21], [13], and [17] proposed presenting task relations as regularization terms in the objective functions to be optimized. The regularization terms can make closer the parameters of models for similar tasks. Another group of methods manage to find the common structure for multi-task learning. For instance, [1], as well as [2] proposed methods for multi-task learning by finding the common structure from data, and then utilizing the learned structure. Our multi-task learning method belongs to the first group, and it is more generally applicable than the existing methods (MKMT v.s. SKMT).

Kernel methods are a principled and powerful approach in machine learning [23, 15]. Conventional kernel methods are defined in the Reproducing Kernel Hilbert Space (RKHS). In our paper, we extend kernel methods to the Square Integrable Space (SIS).

One issue in kernel methods is to choose a proper kernel from a set of candidate kernels. A common practice is to heuristically determine a set of kernels, compare the performances of the kernels, and choose the best one. Multiple Kernel Learning (MKL) aims to solve the kernel selection problem in a principled way. Specifically, it employs a linear combination of kernels and learns the model (classifier) as well as the optimal weights of the linear combination at the same time [cf., 18, 4]. MKMT is different from MKL; the former is about learning for multiple tasks, while the latter is about kernel selection in a single task. We could adopt MKL in selection of the best kernel for each task (the best linear combination of kernels) in our method. In this paper, we simply use the heuristic way of kernel selection and consider integration of MKL into our approach as future work.

The following recent work is also related to, but different from our work. [24] proposed a method of simultaneously learning multiple kernels for multiple tasks, but they did not utilize task relations. [16] proposed to embed data into a low dimensional space by exploiting label correlation. They focus on learning of a better feature representation by employing MKL, while we try to solve a multi-task learning problem. [11] proposed learning classifiers in different function spaces for different tasks in domain adaptation (similar to MKMT). They trained classifiers separately, rather than collectively as in multi-task learning.

3. Premise

Before proposing our approach, we first introduce some notations used in this paper, and review the background knowledge on RKHS and Square Integrable Space (SIS).

3.1. Notations

We consider multi-task learning, specifically, multi-label learning. Suppose that there are T tasks. For each task t, data (x_t, y_t) is generated from $\mathcal{X}_t \times \mathcal{Y}_t$ according to a distribution $P_t(x, y)$. In this paper, we suppose that $\mathcal{X}_t$ is a compact subset in $\mathbb{R}^d$ and $\mathcal{Y}_t = \{+1, -1\}$. Moreover, we have a training data set $S_t = \{(x_{ti}, y_{ti})\}_{i=1}^n$ for each task t and our goal is to learn a classifier $f_t(x)$: $\mathcal{X}_t \to \mathbb{R}$ that can assign a label to a new instance x_t. Following the proposal in [13], we assume that $\mathcal{X}_t$=$\mathcal{X}$ for all tasks and x_{ti} is independent from t[1]. Different tasks share a common marginal distribution $P(x)$ but have different conditional distributions $P_t(y|x)$ (i.e., we consider multi-label classification). In MKMT, the function spaces $\mathcal{F}_t$ ($f_t(x) \in \mathcal{F}_t$) of different tasks are assumed to be different from each other.

We further assume a matrix $\Delta \in \mathbb{R}_+^{T \times T}$ is provided as prior knowledge in training, where element $\delta(s, t) \in [0, 1]$ represents the similarity between tasks s and t. In this paper, we give a heuristic way to learn Δ from training data in Section 6 . We use Δ in learning of the classifiers $\{f_t(x)\}_{t=1}^T$.

3.2. RKHS and Square Integrable Space

We review the theory on Reproducing Kernel Hilbert Space (RKHS) [cf., 3, 9] and show the relationship between RKHS and Square Integrable Space (SIS).

A reproducing kernel Hilbert space $\mathcal{H}$ is a Hilbert space of functions $f(\cdot) : \mathcal{X} \to \mathbb{R}$ that satisfies that $\forall x \in \mathcal{X}$, $f \to f(x)$ is continuous. Suppose the inner product is $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, according to Riesz Representation Theorem, there is a function $K(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined such that $f(x) = \langle f(\cdot), K(\cdot, x) \rangle_{\mathcal{H}}$. $K(x, y)$ is called a reproducing kernel. Moreover, since $K(x, y) = \langle K(\cdot, x), K(\cdot, y) \rangle_{\mathcal{H}}$, the reproducing kernel K satisfies 1) $K(x, y) = K(y, x)$, $\forall x, y \in \mathcal{X}$; 2) $\forall \{\alpha_i\}_{i=1}^n \subset \mathbb{R}, \{x_i\}_{i=1}^n \subset \mathcal{X}$, $\sum_{i,j=1}^n \alpha_i \alpha_j K(x_i, x_j) \geqslant 0$. Inversely, any function $K(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies conditions 1) and 2) can uniquely determine a Hilbert space $\mathcal{H}$ endowed with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ [2], such that 1) $K(\cdot, x) \in \mathcal{H}$, $\forall x \in \mathcal{X}$; 2) $\forall f \in \mathcal{H}$, $f(x) = \langle f(\cdot), K(\cdot, x) \rangle_{\mathcal{H}}$. Thus, $\mathcal{H}$ is a reproducing kernel Hilbert space and K is a reproducing kernel.

A continuous reproducing kernel K is a Mercer kernel. Suppose that $\mathcal{X}$ is endowed with a measure μ, $\mu(\mathcal{X}) < \infty$. We use $\mathcal{L}^2(\mathcal{X}, \mu)$ to denote square integrable function space of $\mathcal{X}$ in which each function $f(x)$ satisfies $\int f^2(x)\mu(\mathrm{d}x) < \infty$.

Given a Mercer kernel K, suppose that the reproducing kernel Hilbert space associated with K is $\mathcal{H}_K$, and the corresponding inner product is given by $\langle \cdot, \cdot \rangle_K$. Let us consider the following operator from $\mathcal{L}^2(\mathcal{X}, \mu)$ to $\mathcal{L}^2(\mathcal{X}, \mu)$:

$$\begin{aligned} L_K : \mathcal{L}^2(\mathcal{X}, \mu) &\to \mathcal{L}^2(\mathcal{X}, \mu) \\ f(x) &\mapsto \int f(y) K(x, y) \mu(\mathrm{d}y). \end{aligned}$$

[1] We can take $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_T$ as input space.

[2] 'unique' is in the sense of isomorphism

It is easy to verify that L_K is a self-adjoint compact operator. According to Hilbert-Schmidt theorem [22], there is a sequence of real eigenvalues $\{\lambda_i\}_{i=1}^N$, $\lambda_1 \geqslant \lambda_2 \geqslant \cdots$, $\lim_{i\to+\infty} \lambda_i = 0$ if $N = +\infty$, and an orthonormal basis $\{e_i(x)\}_{i=1}^N$ of $\mathcal{L}^2(\mathcal{X}, \mu)$, such that $L_K(e_i(x)) = \lambda_i e_i(x)$, $\forall i$, $e_i(x)$ is a continuous function. Moreover, $K(x, y) = \sum_{i=1}^N \lambda_i e_i(x)e_i(y)$, and the convergence is absolute and uniform.

Given any function $f(x) \in \mathcal{H}_K$, since

$$\begin{aligned}\int f^2(x)\mu(\mathrm{d}x) &= \int (\langle f(\cdot), K(\cdot, x)\rangle_K)^2 \mu(\mathrm{d}x)\\ &\leqslant \int (\langle f(\cdot), f(\cdot)\rangle_K)^2 K(x, x)\mu(\mathrm{d}x) < \infty,\end{aligned}$$

$f(x) \in \mathcal{L}^2(\mathcal{X}, \mu)$. Thus, RKHS $\mathcal{H}_K$ is a subspace of SIS $\mathcal{L}^2(\mathcal{X}, \mu)$, for any Mercer kernel K.

Since $\forall f(x) \in \mathcal{H}_K$, $f(x) \in \mathcal{L}^2(\mathcal{X}, \mu)$, $f(x)$ can be represented as $f(x) = \sum_{i=1}^N a_i e_i(x)$, where $a_i = \int f(x)e_i(x)\mu(\mathrm{d}x)$. Furthermore, we have the following theorem, which characterizes RKHS $\mathcal{H}_K$ by $\{\lambda_i\}_{i=1}^N$ and the orthonormal basis of SIS $\{e_i(x)\}_{i=1}^N$ [9]:

Theorem 3.1. $\mathcal{H}_K = \{f \in \mathcal{L}^2(\mathcal{X}, \mu) \mid f(x) = \sum_{i=1}^N a_i e_i(x),\ \sum_{i=1}^N \frac{a_i^2}{\lambda_i} < \infty\}$. *The convergence is absolute and uniform, thus* $\forall f(x) \in \mathcal{H}_K$, $f(x) \in C(\mathcal{X})$, *where* $C(\mathcal{X})$ *is continuous function space of* $\mathcal{X}$. *Moreover, suppose that* $f(x), g(x) \in \mathcal{H}_K$, $f(x) = \sum_{i=1}^N a_i e_i(x)$ *and* $g(x) = \sum_{i=1}^N b_i e_i(x)$, *the inner product* $\langle f(\cdot), g(\cdot)\rangle_K$ *in* $\mathcal{H}_K$ *is given by* $\sum_{i=1}^N \frac{a_i b_i}{\lambda_i}$.

4. Our Approach

We propose a novel and general kernel approach to multi-task learning using task relations. Formally, suppose that RKHS $\mathcal{H}_t$ is generated by kernel κ_t for task t. We learn function (model) f_t from $\mathcal{H}_t$. Since kernels κ_t may be different from each other, $f_1, f_2, \cdots, f_T$ may be no longer in the same space (i.e., RKHS). We consider using Square Integrable Space (SIS) as the space containing all the RKHSs $\mathcal{H}_t$, which is supported by Theorem 3.1. We conduct multi-task learning in SIS $\mathcal{L}^2(\mathcal{X}, \mu)$.

One advantage of the approach is that we can naturally use task relations in $\mathcal{L}^2(\mathcal{X}, \mu)$, since SIS contains the RKHSs for different tasks. More importantly, we can offer a theoretical justification to the approach by proving the representer theorem and the convergence of the practical solution.

4.1. Ideal Solution

Multi-task learning is then defined as the following optimization problem:

$$\arg\min_{f_t\in\mathcal{H}_t} \frac{1}{n}\sum_{t=1}^T\sum_{i=1}^n L(x_i, y_{ti}, f_t) + \gamma_1 \sum_{t=1}^T \|f_t\|_{\kappa_t}^2 + \frac{\gamma_2}{2}\sum_{s,t=1}^T \delta(s, t) \int (f_s(x) - f_t(x))^2 \mu(\mathrm{d}x), \quad (1)$$

where the second term is a normal regularization term which control the complexity of models in their own RKHSs, and the third term is a regularization term which measure difference of models in the common space (i.e.,$\mathcal{L}^2(\mathcal{X}, \mu)$). The underlying assumption is that if two tasks s and t are similar ($\delta(s, t)$ is large), then the corresponding models should also be similar in the common space.

To guarantee the existence of solution and identify the form of solution, we need a new Representer Theorem for the new kernel method:

Theorem 4.1 (**Representer Theorem**). *Suppose that loss function $L(\cdot,\cdot,\cdot)$ is convex and continuous, the solution to the optimization problem* (1) *exists and has the following form:*

$$f_t^\star(x) = \sum_{i=1}^{n} \alpha_{ti}\kappa_t(x_i, x) + \int \theta_t(y)\kappa_t(y, x)\mu(\mathrm{d}y),$$

where $\alpha_{ti} \in \mathbb{R}$ and $\theta_t(\cdot) \in \mathcal{L}^2(\mathcal{X}, \mu)$. Note that here we require loss function L to be convex and continuous. This condition appears to be slightly strong, but is satisfied by most of the commonly used loss functions.

We can use the marginal distribution $P(x)$ as μ. In this way, the solution of problem (1) becomes

$$f_t^\star(x) = \sum_{i=1}^{n} \alpha_{ti}\kappa_t(x_i, x) + \int \theta_t(y)\kappa_t(y, x)P(\mathrm{d}y). \tag{2}$$

Formula (2) offers an ideal solution to our approach. The proof of Theorem 4.1 can be found in Appendix A.

4.2. Practical Solution

To obtain the ideal solution, we need to know marginal distribution $P(x)$, and then solve a functional optimization problem (Note θ_t is a $\mathcal{L}^2$ integrable function).

In practice, we use the empirical distribution $\hat{P}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(x = x_i)$ [3] to estimate $P(x)$. Thus, the solution of problem (1) becomes

$$\hat{f}_t(x) = \sum_{i=1}^{n} \alpha_{ti}\kappa_t(x_i, x) + \frac{1}{n}\sum_{i=1}^{n} \theta_{ti}\kappa_t(x_i, x). \tag{3}$$

We need to answer the question whether the practical solution converges to the ideal solution when training size goes to infinity. The difficulty is that we need to verify the convergence of the optimal parameters $\{\alpha_{ti}\}$ in formula (3) to the optimal parameters in formula (2), and the convergence of optimal parameters $\{\theta_{ti}\}$ in formula (3) to the optimal functions $\{\theta_t(x)\}$ in formula (2). Here, the optimal parameters are those obtained by solving the optimization problem (1) with respect to $P(x)$ and $\hat{P}(x)$, respectively. As shown below, we are able to prove the convergence, by analyzing the relationship between the two minimums of the optimization problem (1) under $P(x)$ and $\hat{P}(x)$ respectively.

4.3. Convergence of Practical Solution

In this section, we further assume that L is differentiable and prove the convergence of practical solution under the condition.[4] Theorem 4.5 gives a bound on the difference between the two solutions. The result indicates that the practical solution given by formula (3) converges to the ideal solution given by formula (2) in probability. Moreover, it also gives the convergence speed, which will enable us to estimate the number of instances necessary to make the difference

[3] $\mathbb{1}(x = x_i) = 1$, if $x = x_i$, otherwise, $\mathbb{1}(x = x_i) = 0$

[4] Whether the same conclusion holds under a weaker condition is still an open question, in which L is only continuous. Our hypothesis is that it may be the case, because we can use differentiable functions to approximate a continuous function.

between the two solutions small enough with high probability. We briefly explain how to obtain the bound and present the proofs of theorems in Appendix B.

Define $D_t = \sup_{x \in \mathcal{X}} |f_t^\star(x) - \hat{f}_t(x)|$. Suppose that $\max_{1 \leqslant t \leqslant T} \sup_{x \in \mathcal{X}} \kappa_t(x, x) \leqslant B$. Define $h(\mathcal{F}, X) = \sup_{f \in \mathcal{F}} |Ef^2 - \frac{1}{n} \sum_{i=1}^n f^2(x_i)|$, where $X = \{x_i\}_{i=1}^n$, $\mathcal{F} = \{f | f = \sum_{t=1}^T f_t, f_t \in \mathcal{H}_t, \|f_t\|_{\kappa_t} \leqslant R^*\}$. Since $\{f_t^\star(x)\}_{t=1}^T$ minimize (1) with respect to $P(x)$, $\sum_{t=1}^T \|f_t^\star\|_{\kappa_t}^2 \leqslant \frac{1}{n\gamma_1} \sum_{t=1}^T \sum_{i=1}^n L(x_i, y_{ti}, 0)$, where $\frac{1}{n} \sum_{t=1}^T \sum_{i=1}^n L(x_i, y_{ti}, 0)$ is the value of the objective function (1) on 0. Since L is differentiable, and thus continuous, we can suppose that $\max_{x,y} L(x, y, 0) \leqslant U$. Thus, $\sum_{t=1}^T \|f_t^\star\|_{\kappa_t}^2 \leqslant \frac{TU}{\gamma_1}$. Similarly, $\sum_{t=1}^T \|\hat{f}_t\|_{\kappa_t}^2 \leqslant \frac{TU}{\gamma_1}$. We can let $R^* = \sqrt{\frac{TU}{\gamma_1}}$, thus, $\forall t, s$, $f_t^\star - f_s^\star$ and $\hat{f}_t - \hat{f}_s$ are in $\mathcal{F}$. We first bound $h(\mathcal{F}, X)$. Using the McDiarmid inequality [5], we obtain the following theorem:

Theorem 4.2. *Given an arbitrary small positive number δ, with probability more than $1 - \delta$, the following inequality holds:*

$$|h(\mathcal{F}, X) - Eh(\mathcal{F}, X)| \leqslant (TR^*)^2 B \sqrt{\frac{2\ln(2/\delta)}{n}}.$$

With this theorem, we only need to bound $Eh(\mathcal{F}, X)$. Using the conclusion given by Theorem 12.6 and techniques in the proof of Theorem 8 and Lemma 22 in [5], we obtain

Theorem 4.3.

$$E(h(\mathcal{F}, X)) \leq \frac{8(TR^*)^2 B}{\sqrt{n}}$$

Combining the results in Theorem 4.2 and 4.3, we finally obtain the bound for $h(\mathcal{F}, X)$:

Theorem 4.4. *With probability more than $1 - \delta$, we have the following inequality hold:*

$$h(\mathcal{F}, X) \leq (TR^*)^2 B \sqrt{\frac{2\ln(2/\delta)}{n}} + \frac{8(TR^*)^2 B}{\sqrt{n}} \triangleq g(n).$$

With the results above, we reach our conclusion:

Theorem 4.5. *With probability more than $1 - \delta$, the following inequality holds:*

$$D_t = \sup_{x \in \mathcal{X}} |f_t^\star(x) - \hat{f}_t(x)| \leqslant O(1/n^{\frac{1}{4}}) \quad 1 \leqslant t \leqslant T.$$

4.4. Relation with Existing Approach

[13] formalize multi-task learning as the following optimization problem :

$$\arg\min_{\{f_t\} \subset \mathcal{H}} \frac{1}{n} \sum_{t=1}^T \sum_{i=1}^n L(x_i, y_{ti}, f_t) + \gamma_1 \sum_{t=1}^T \|f_t\|_\kappa^2 + \frac{\gamma_2}{2} \sum_{s,t=1}^T \delta(s, t) \|f_s - f_t\|_\kappa^2.$$

where the second term is a regularization term in RKHS, and the third term is a regularization term based on task relations, also defined in the RKHS. In their approach , in order to make comparison between models for different tasks *and* exploit task relations, models are assumed to be in the RKHS generated by the same kernel κ, which is the major difference from our approach. The following theorem shows the relationship between the two regularization terms of our approach and Evgeniou et al.'s approach when all tasks share the same kernel.

Theorem 4.6. *Suppose all tasks share the same kernel κ, $\forall s, t$, the following inequality holds:*

$$\delta(s,t)\|f_s - f_t\|_\kappa^2 \geqslant C(\kappa,\mu)\delta(s,t)\int (f_s(x) - f_t(x))^2\mu(\mathrm{d}x),$$

where $C(\kappa,\mu)$ is a positive constant related to kernel κ and measure μ. Moreover, if κ satisfies $\kappa(x,y) \geqslant 0$ and $\int \kappa(x,y)\mu(\mathrm{d}x) = 1$ $\forall y$ (e.g., Gaussian kernel), $C(\kappa,\mu) \leqslant 1$.

The proof of Theorem 4.6 can be found in Appendix C. Theorem 4.6 indicates that for SKMT cases our approach can work as well as existing approach.

5. Implementation

We give a specific algorithm of our approach. We define the loss function L in the objective function (1) as hinge loss, and thus define $\{f_t\}_{t=1}^T$ as SVM classifiers.

The learning problem becomes:

$$\operatorname*{arg\,min}_{f_t\in\mathcal{H}_t} \frac{1}{n}\sum_{t=1}^{T}\sum_{i=1}^{n}(1 - y_{ti}(f_t(x_i) + b_t))_+ + \gamma_1\sum_{t=1}^{T}\|f_t\|_{\kappa_t}^2 + \frac{\gamma_2}{2n}\sum_{s,t=1}^{T}\sum_{i=1}^{n}\delta(s,t)(f_s(x_i) - f_t(x_i))^2.$$

Note that here following the convention we add a bias b_t into each classifier $f_t(x)$. By introducing slack variables, we obtain the prime problem:

$$\begin{aligned}&\operatorname*{arg\,min}_{f_t\in\mathcal{H}_t} \sum_{t=1}^{T}\sum_{i=1}^{n}\xi_{ti} + \gamma_1'\sum_{t=1}^{T}\|f_t\|_{\kappa_t}^2 + \frac{\gamma_2}{2}\sum_{s,t=1}^{T}\sum_{i=1}^{n}\delta(s,t)(f_s(x_i) - f_t(x_i))^2\\ &\text{subject to}\quad y_{ti}(f_t(x_i) + b_t) \geqslant 1 - \xi_{ti} \quad \xi_{ti} \geqslant 0\end{aligned}$$

Equation (3) gives the solution to the prime problem, and we combine parameters α_{ti} and θ_{ti} together as α'_{ti} here:

$$\hat{f}_t(x) = \sum_{i=1}^{n}\alpha'_{ti}\kappa_t(x_i, x) \quad 1 \leqslant t \leqslant T$$

Substituting this solution into the prime problem, we obtain the following dual problem:

$$\begin{aligned}&\operatorname*{arg\,max}_{\beta\in\mathbb{R}^{nT}} \sum_{t=1}^{T}\sum_{i=1}^{n}\beta_{ti} - \frac{1}{4\gamma_1'}\beta^\top Y\mathcal{K}\left(\mathcal{K} + \frac{\gamma_2}{\gamma_1'}\mathcal{K}(\mathcal{L}\otimes I)\mathcal{K}\right)^{-1}\mathcal{K}Y\beta.\\ &\text{subject to}\quad \sum_{i=1}^{n}\beta_{ti}y_{ti} = 0 \quad 1 \leqslant t \leqslant T, \quad 0 \leqslant \beta_{ti} \leqslant 1\end{aligned}$$

Here, $\beta = (\beta_1^\top, \beta_2^\top, \cdots, \beta_T^\top)^\top$ is the dual variable, where $\beta_t = (\beta_{t1}, \beta_{t2}, \cdots, \beta_{tn})^\top$ $t = 1, 2\cdots T$. Y is a diagonal matrix whose $t \times i$-th element is y_{ti}. $\mathcal{L}$ is task graph Laplacian which is constructed by taking $\delta(s,t)$ as the weight of edge connecting task s and t on an undirected graph. $\mathcal{K}$ is a block diagonal matrix with each block $\mathcal{K}_t = \left(\kappa_t(x_i, x_j)\right)_{n\times n}$.

After getting the optimal β^*, we can compute the optimal $\alpha'^* = ({\alpha_1}'^{*\top}, {\alpha_2}'^{*\top}, \cdots, {\alpha_T}'^{*\top})^\top$ where ${\alpha_t}'^* = ({\alpha_{t1}}'^*, {\alpha_{t2}}'^*, \cdots, {\alpha_{tn}}'^*)^\top$ through the following equation:

$$\alpha'^* = \frac{1}{2\gamma_1'}\left(\mathcal{K} + \frac{\gamma_2}{\gamma_1'}\mathcal{K}(\mathcal{L}\otimes I)\mathcal{K}\right)^{-1}\mathcal{K}Y\beta^*. \tag{4}$$

Algorithm 1

```
1: Input: training data $\{x_i\}_{i=1}^n$,$\{y_{ti}\}_{i=1}^n$ $1 \leqslant t \leqslant T$, task similarity matrix $\Delta$
2: Choose a proper kernel $\kappa_t$ for each task $t$
3: Choose proper $\gamma_1'$ and $\gamma_2$.
4: Compute matrix $\left(\mathcal{K} + \frac{\gamma_2}{\gamma_1'}\mathcal{K}(\mathcal{L} \otimes I)\mathcal{K}\right)^{-1}$
5: Compute $\beta^*$ by solving the dual problem
6: Compute $\alpha'^*$ by using equation (4).
7: Output: $f_t^*(x) = \sum_{t=1}^n \alpha_{ti}'^* \kappa_t(x_i, x) + b_t^*$, $1 \leqslant t \leqslant T$
```

The details of the algorithm are shown in Algorithm 1. At step 2, we empirically find a proper kernel for each task, from a number of kernels. At step 5, we solve a QP problem. We specifically employ Franke and Wolf's method [see 12], which is a gradient descent based method. At step 4, we need to compute inverse of matrix, which is of order $O(n^3)$. We focus on problem formulation and theoretical analysis in this paper, and leave to future work the improvements of our method on efficiency and kernel selection.

6. Experiments

We conducted experiments to verify the effectiveness of our approach. We used an artificial data set given in Figure 1 and three classification data sets: protein classification data set, music classification data set, and video classification data set. We considered two baselines: Individual and Single (Kernel). In the former, SVM classifiers for the tasks are trained individually (i.e., task relations are ignored), and in the latter, SVM classifiers for tasks are trained using Evgeniou et al.'s method. We denote our method Multiple (Kernel). For the artificial data set, we used MicroF1 value, MacroF1 value [20], and error rate as evaluation measures. For other real-world data sets, we utilized ROC score [19] as evaluation measure.

In our experiments, we employ the following heuristic method to learn $\{\delta(s,t)\}$ from training data. We create a vector for each task (class) based on training data. Each element of the vector corresponds to an instance; and if the instance belongs to the class (task), then we set the value of the element as 1, otherwise we set it as 0. Finally, we take the *cosine* of the vectors of two tasks s and t as $\delta(s,t)$. We actually use positive co-occurrence of tasks (classes) as similarity between them.

6.1. Artificial Data Classification

As shown in Figure 1, there are three classes, and classification of instances to one class corresponds to one task. The 400 instances (circle points) in the square area on the right side belong to class 1, the 400 instances (diamond points) in the outer circle area and the 400 instances (cross points) in the inner circle area on the left side belong to classes 2 and 3, respectively. The 200 instances (square points) in the middle belong to both classes 1 and 2 (i.e., they have multiple labels). Totally, classes 1, 2 and 3 have 600, 600 and 400 instances respectively. We randomly chose 5%, 5%, and 90% of instances in each class as training, validation, and testing data respectively. In many real world problems usually there are small amount of training data and large amount of testing data. We tried to simulate such kind of situation.

We used the heuristic method referred above to set the similarity between tasks 1 and 2 as 0.67, similarity between tasks 1 and 3 as 0, and similarity between tasks 2 and 3 as 0.

Table 1: Accuracies of methods on artificial data classification

	Individual	Single	Multiple
Average MicroF1			
	0.9	0.83	**0.92**
Average MacroF1			
	0.89	0.83	**0.92**
Average Error Rates			
class 1	**0.4%**	14%	0.8%
class 2	12.4%	12.8%	**9.6%**
class 3	8%	**7.6%**	8%

We chose the best kernel for each task, from linear kernel and Gaussian kernel $\exp^{-\gamma\|x-y\|^2}$ (including parameter). At the same time, we also determined the best value for parameter γ_1'. The result indicates that for classes 1, 2, 3, the best kernels are linear kernel, Gaussian kernel with $\gamma = 0.1$, and Gaussian kernel with $\gamma = 0.1$, respectively. The results combined over tasks are actually those for Individual.

Besides, we tuned parameters γ_2 in Single and Multiple. We first selected the best kernel and the best parameter γ_1' for Individual, and then fixed them for Single and Multiple, and selected the best parameters γ_2 for Single and Multiple. Parameter selection was conducted with validation data. γ_1' was in $\{0.1, 0.5, 1, 5\}$, and γ_2 was in $\{0.00075, 0.00375, 0.0075, 0.0375, 0.075, 0.375, 0.75, 3.75, 7.5, 37.5\}$. For Single which employs only one single kernel, we used the Gaussian kernel which performs the best for all tasks.

We repeated the process ten times, and Table 1 gives the average results of the three methods in terms of MicroF1 and MacroF1. Table 1 also shows the error rates for each individual task by each method.

Multiple performs as well as Individual on classes 1 and 3, but better than Individual on class 2. (Note that the two methods employ the same kernels while Multiple uses task relations but Individual does not.) It seems that the use of task relations can help Multiple to achieve better performance. Multiple works as well as Single on class 3, but significantly better than Single on classes 1 and 2. (Note that the two methods rely on the same task relation information, but Single makes use of the same kernel for all tasks while Multiple makes use of different kernels for different tasks). It is obvious that for class 1 employing a linear kernel is better, and that is why for this class Multiple performs better than Single. For class 2, there are instances in both the outer circle area and the middle area. Multiple seems to be able to leverage task relation and multiple kernels to achieve high accuracy (Note that Multiple performs well on class 1), while it is hard for Single to do so.

6.2. Protein Data Classification

In this experiment, a benchmark data set [5] on classification of protein functions was used. The data set contains 3,588 proteins with 13 function classes, and each protein may be associated with one or more functions. The average number of functions per protein is 1.53. For more details, see [19].

[5] http://noble.gs.washington.edu/proj/yeast/

Table 2: Comparison of three methods on protein data

	Individual	Single (K_{sw})	Single (K_{pfam_E})	Multiple
class 1	0.721	0.697	0.736	$\mathbf{0.740}^{1,2}$
class 2	**0.666**	0.652	0.638	0.657^{3}
class 3	0.654	0.655	0.653	$\mathbf{0.676}^{1,3}$
class 4	0.728	0.745	0.741	$\mathbf{0.753}^{1,3}$
class 5	0.779	0.790	0.779	$\mathbf{0.797}^{1,3}$
class 6	0.682	0.650	0.686	$\mathbf{0.688}^{2}$
class 7	0.671	0.653	0.673	$\mathbf{0.687}^{1,2,3}$
class 8	0.635	**0.641**	0.621	$0.636^{1,3}$
class 9	0.605	0.584	**0.611**	0.607^{2}
class 10	0.649	0.646	0.600	$\mathbf{0.651}^{3}$
class 11	0.541	**0.557**	0.542	0.541
class 12	0.881	0.826	0.887	$\mathbf{0.891}^{1,2,3}$
class 13	0.579	0.590	0.594	$\mathbf{0.602}^{1}$

We randomly chose 500 instances as training data, and evenly divided the rest into two subsets for validation and testing. We used the heuristic method above to calculate task similarities from the training data.

We selected the best kernel for each task heuristically by using the validation set. We tuned the parameters for Individual, Single, and Multiple. In this data set, no information on features is available, and the kernels are provided as kernel matrices. There are in total 8 kernel matrices (8 kernels). We then tuned the parameters for the three methods. We followed the parameter setting in [19], that is, we fixed γ_1' as 1 for all the three methods (equivalent to having $C = 1$). γ_2 for Single and Multiple was chosen from $\{0.01, 0.05, 0.1, 0.5, 1, 5\}$.

We repeated the above process ten times, and the final results are averaged over ten trials. For classes 2, 8, 10, and 13 the best performing kernel for Individual is Smith-Waterman kernel (K_{sw}), and for the other classes the best performing kernel is Enriched Pfam kernel (K_{pfam_E}).

Table 2 gives the results of the three methods. Superscripts 1, 2, and 3 stand for that the improvements of our methods are statistically significant over Individual, Single using kernel K_{sw} and Single using kernel K_{pfam_E}, respectively.

We can see that Multiple is better than Individual on eleven classes among thirteen classes and for eight of the outperforming cases, the improvement is statistically significant.

The results in Table 1 indicate that kernel selection is crucial for Single. With a good kernel selection, SKMT's performance can be comparable to Multiple, which agrees with the theoretical result in Theorem 4.6, but with a bad kernel selection, the performance can be worse than Individual. In contrast, Multiple can exploit different kernels as well as the task relations to achieve a good performance.

6.3. Music Data Classification

We test our method on another data set [6] on music emotion classification (multi-label classification). The data contains 593 pieces of music and 6 types of emotion. Each piece of music

[6] http://mulan.sourceforge.net/datasets.html

Table 3: Comparison of three methods on music data

	Individual	Single (Gaussian)	Single (Polynomial)	Multiple
class 1	0.723	0.675	0.727	**0.732**$^{\mathbf{1,2}}$
class 2	0.570	0.543	**0.584**	0.580^{2}
class 3	0.764	**0.777**	0.643	0.762^{3}
class 4	0.855	0.816	0.863	**0.872**$^{\mathbf{2}}$
class 5	0.748	0.653	0.755	**0.760**$^{\mathbf{1,2}}$
class 6	0.736	0.706	0.747	**0.752**$^{\mathbf{1,2}}$

expresses one or more types of emotion. The average number of emotion types per piece of music is 1.87.

We randomly chose 100 instances, 243 instances, and 250 instances as training data, validation data, and testing data, respectively. We created Gaussian kernels ($exp^{-\sigma\|x-y\|^2}$) with σ varying in $\{0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$, linear kernel, and polynomial kernels with degree $2-5$ as kernel candidates. We chose γ_1' from $\{0.1, 0.5, 1, 5\}$, calculated task similarities, selected kernels, and tuned model parameters, in the same way in the protein data experiment.

We also repeated the process ten times. Table 3 gives the average results. For classes except 3, polynomial kernel with degree 5 performs best, and for class 3, Gaussian kernel with $\sigma = 0.01$ is the best performing one. Superscripts 1, 2, and 3 mean that the improvement of our method is statistically significant when compared with Individual, Single using Gaussian kernel, and Single using Polynomial kernel, respectively. We can make the same conclusions as in the protein data experiment. That is, our method can effectively utilize the task relation and different kernels to handle MKMT problems.

6.4. Video Data Classification

To further verify the effectiveness of our method, we conducted another experiment [7] on video data classification. The data set contains 43,907 videos and totally 101 semantic concepts. Each video is labeled with at least one concept (i.e., the data set is a multi-label data set). The average number of concepts per video is 4.37.

To control the scale, we calculated the frequencies of the 101 concepts in the entire data set, and chose the top 10 most frequent concepts as classes in our experiment. After removing the videos without the top 10 concepts, there were 41,583 videos left, and the average number of classes per video became 3.17. We randomly chose 833 videos, 750 videos, and 40,000 videos as training data, validation data, and testing data, respectively.

As in music data classification, we created Gaussian kernels ($exp^{-\sigma\|x-y\|^2}$) with σ varying in $\{0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$, linear kernel, and polynomial kernels with degree $2-5$, as kernel candidates. We chose γ_1' from $\{0.1, 0.5, 1, 5\}$, calculated task similarities, selected kernels, and tuned model parameters, in the same way in the music data experiment.

We repeated the process ten times. Table 4 gives the average results. There are totally 3 optimal kernels. For classes $1-5$, Gaussian kernel with $\sigma = 1$ is optimal. For classes 6 and 7, Gaussian kernel with $\sigma = 5$ is the one with the best performance. For classes $8-10$, Gaussian kernel with $\sigma = 0.01$ is the best choice. Superscripts 1, 2, 3, and 4 indicate that the improvement

[7] http://mulan.sourceforge.net/datasets.html

Table 4: Comparison of three methods on video data

	Individual	Single	Single	Single	Multiple
		(Gaussian $\sigma = 1$)	(Gaussian $\sigma = 5$)	(Gaussian $\sigma = 0.01$)	
class 1	**0.805**	0.804	0.799	0.766	0.787^{4}
class 2	**0.812**	**0.812**	0.798	0.730	**0.812**^{3,4}
class 3	0.757	**0.758**	0.734	0.725	**0.758**^{3,4}
class 4	0.778	0.778	0.758	0.727	**0.789**^{1,2,3,4}
class 5	0.750	0.753	0.755	0.753	**0.759**^{1}
class 6	0.645	0.634	**0.648**	0.615	**0.648**^{1,2,4}
class 7	**0.619**	0.608	**0.619**	0.553	**0.619**^{2,4}
class 8	0.816	0.825	**0.831**	0.823	0.822^{1}
class 9	0.661	0.677	0.677	0.680	**0.695**^{1,2,3}
class 10	0.707	0.723	0.738	0.720	**0.754**^{1,2,3,4}

of our method is statistically significant over Individual, Single using Gaussian kernel with $\sigma = 1$, Single using Gaussian kernel with $\sigma = 5$, and Single using Gaussian kernel with $\sigma = 0.01$, respectively. We can see that by exploiting task relations, our method significantly outperforms Individual on 6 out of 10 classes. If Single uses the best kernel, its performance is comparable with our method, but if not, our method usually significantly outperforms Single. We can make the same conclusions as in the other experiments. That is, our method can effectively leverage both task relations and different kernels to handle MKMT problems.

7. Conclusion

In this paper, we have proposed a new kernel approach to multi-task learning using task relations. The main characteristics of the approach is to formalize the problem in the Square Integrable Space in order to employ multiple kernels for multiple tasks. Specifically, our method incorporates task relations into a regularization term in the objective function. We have addressed the theoretical issues of the learning method, specifically, proved the representer theorem, derived a practical solution, proved the convergence of the practical solution to the ideal solution, and verified the relation between our method and an existing method. We have also proposed an algorithm for implementing our approach for multi-label classification, on the basis of SVM. We have conducted experiments and empirically verified that our method is indeed very effective for multi-task learning.

As future work, we plan (1) to study the generalization ability of our method, (2) to study principled ways of selecting kernels in our approach, (3) to develop more efficient learning algorithms, and (4) to conduct experiments on other data sets.

References

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

[2] Argyriou, A., Evgeniou, T., Pontil, M., 2007. Multi-task feature learning. Advances in Neural Information Processing Systems 19, 41.

[3] Aronszajn, N., 1950. Theory of Reproducting Kernels. Transactions of the American Mathematical Society 68 (3), 337–404.

[4] Bach, F., Lanckriet, G., Jordan, M., 2004. Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the twenty-first international conference on Machine learning. ACM, p. 6.

[5] Bartlett, P., Mendelson, S., 2003. Rademacher and Gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research 3 (3), 463–482.

[6] Belkin, M., Niyogi, P., Sindhwani, V., 2006. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. Journal of Machine Learning Research 7, 2399–2434.

[7] Ben-David, S., Schuller, R., 2003. Exploiting task relatedness for multiple task learning. Lecture notes in computer science, 567–580.

[8] Caruana, R., 1997. Multitask learning. Machine Learning 28 (1), 41–75.

[9] Cucker, F., Smale, S., 2002. On the mathematical foundations of learning. Bulletin-American Mathematical Society 39 (1), 1–50.

[10] Dieudonné, J., 1960. Foundations of modern analysis.

[11] Duan, L., Tsang, I., Xu, D., Chua, T., 2009. Domain adaptation from multiple sources via auxiliary classifiers. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM, pp. 289–296.

[12] Elisseeff, A., Weston, J., 2002. Kernel methods for Multi-labelled classification and Categorical regression problems. In: Advances in Neural Information Processing Systems.

[13] Evgeniou, T., Micchelli, C., Pontil, M., 2006. Learning multiple tasks with kernel methods. Journal of Machine Learning Research 6 (1), 615.

[14] Evgeniou, T., Pontil, M., 2004. Regularized multi–task learning. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM New York, NY, USA, pp. 109–117.

[15] Hofmann, T., Scholkopf, B., Smola, A., 2008. Kernel methods in machine learning. Annals of Statistics 36 (3), 1171.

[16] Ji, S., Sun, L., Jin, R., Ye, J., 2009. Multi-label multiple kernel learning. NIPS.

[17] Kato, T., Kashima, H., Sugiyama, M., Asai, K., 2008. Multi-task learning via conic programming. Advances in Neural Information Processing Systems 20, 737–744.

[18] Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., Jordan, M., 2004. Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research 5, 27–72.

[19] Lanckriet, G., Deng, M., Cristianini, N., Jordan, M., Noble, W., 2004. Kernel-based data fusion and its application to protein function prediction in yeast. In: Proceedings of the Pacific Symposium on Biocomputing. Vol. 9. pp. 300–311.

[20] Lewis, D., 1991. Evaluating text categorization. In: Proceedings of Speech and Natural Language Workshop. Morgan Kaufmann, pp. 312–318.

[21] Micchelli, C., Pontil, M., 2005. Kernels for multi–task learning. Advances in Neural Information Processing Systems 17, 921–928.

[22] Renardy, M., Rogers, R., 2004. An introduction to partial differential equations. Springer Verlag.

[23] Schölkopf, B., Smola, A., 2002. Learning with kernels: Support vector machines, regularization, optimization, and beyond. MIT press.

[24] Tang, L., Chen, J., Ye, J., 2009. On Multiple Kernel Learning with Multiple Labels. In: IJCAI'09.

Appendix A.

Proof of Theorem 4.1 (Representer Theorem)

To prove Theorem 4.1, we need two lemmas:

Lemma .1. *Given Mercer kernels $\{\kappa_t\}_{t=1}^T$, $\mathcal{H}_t$ is the reproducing kernel Hilbert space generated by κ_t. Ball $\mathcal{B}_r = \{\vec{f} = (f_1, f_2, \cdots, f_T) \mid f_t \in \mathcal{H}_t, \sum_{t=1}^T \|f_t\|_{\kappa_t}^2 \leqslant r\}$ is a compact subset of $C(\mathcal{X})$, where $r \geqslant 0$ and $C(\mathcal{X})$ is continuous function space of $\mathcal{X}$.*

Proof. We first consider an embedding:

$$\begin{aligned} &\mathcal{H} \to C(\mathcal{X}) \\ &\vec{f} \mapsto \vec{f}, \end{aligned}$$

where $\mathcal{H} = \mathcal{H}_1 \times \mathcal{H}_2 \times \cdots \mathcal{H}_T$, and the corresponding norm is defined as $\|\vec{f}\|_{\mathcal{H}} = \sqrt{\sum_{t=1}^T \|f_t\|_{\kappa_t}^2}$. As the first step, we try to prove that the embedding above is a compact embedding. To achieve this goal, we only have to prove that for any sequence $\{\vec{f_n} = (f_{n1}, f_{n2}, \cdots, f_{nT})\}$ that satisfies $\|\vec{f_n}\|_{\mathcal{H}} \leqslant M$, there exists a subsequence that converges in $C(\mathcal{X})$.

Suppose that $\max_{1 \leqslant t \leqslant T} \sup_{x \in \mathcal{X}} \kappa_t(x, x) \leqslant B$. This is possible because $\mathcal{X}$ is compact and κ_t is continuous. Since $\forall x \in \mathcal{X}$, $|f_{nt}(x)| = |\langle f_{nt}(\cdot), \kappa_t(\cdot, x)\rangle_{\kappa_t}| \leqslant \|f_{nt}\|_{\kappa_t} \sqrt{\kappa_t(x, x)} \leqslant M\sqrt{B}$, we have $\|\vec{f_n}\|_C \leqslant TM\sqrt{B}$. Thus, $\{\vec{f_n}\}$ is uniformly bounded.

For any fixed $x_0 \in \mathcal{X}$,

$$\begin{aligned} |f_{nt}(x) - f_{nt}(x_0)| &= |\langle f_{nt}(\cdot), \kappa_t(\cdot, x) - \kappa_t(\cdot, x_0)\rangle_{\kappa_t}| \\ &\leqslant \|f_{nt}\|_{\kappa_t} \sqrt{\kappa_t(x, x) - 2\kappa_t(x, x_0) + \kappa_t(x_0, x_0)} \\ &\leqslant M\sqrt{\kappa_t(x, x) - 2\kappa_t(x, x_0) + \kappa_t(x_0, x_0)}. \end{aligned}$$

Since $\forall t$, κ_t is continuous, thus, $\forall \epsilon$, $\exists \delta > 0$, when $|x - x_0| < \delta$, $\sum_{t=1}^T \sqrt{\kappa_t(x, x) - 2\kappa_t(x, x_0) + \kappa_t(x_0, x_0)} < \frac{\epsilon}{M}$. Thus, when $|x - x_0| < \delta$, $\sum_{t=1}^T |f_{nt}(x) - f_{nt}(x_0)| < \epsilon$, $\forall n$. This means $\{\vec{f_n}\}$ is equicontinuous.

Since $\{\vec{f_n}\}$ is uniformly bounded and equicontinuous, according to Ascoli Theorem [10], we know that there is a subsequence of $\{\vec{f_n}\}$ that converges uniformly. The uniformly convergent subsequence is the subsequence that converges in $C(\mathcal{X})$. Thus, the embedding $\mathcal{H} \to C(\mathcal{X})$ is a compact embedding.

Since $\mathcal{H}$ is compactly embedded in $C(\mathcal{X})$, to prove $\mathcal{B}_r$ is compact in $C(\mathcal{X})$, we only need to prove that $\mathcal{B}_r$ is closed in $C(\mathcal{X})$.

$\forall \{\vec{f_n}\} \subset \mathcal{B}_r$, if $\vec{f_n}$ converges to $\vec{f}$ in $C(\mathcal{X})$, we try to prove that $\vec{f} \in \mathcal{B}_r$. According to Theorem 3.1, we know $f_{nt}(x) = \sum_{i=1}^{N_t} a_i^{nt} e_i^t(x)$, where $\{e_i^t(x)\}$ is the orthonormal basis of square integrable space associated with kernel κ_t, and N_t is finite or infinite. Moreover, $\|f_{nt}\|_{\kappa_t}^2 = \sum_{i=1}^{N_t} \frac{(a_i^{nt})^2}{\lambda_i^t}$. Suppose $f_t(x) = \sum_{i=1}^{N_t} a_i^t e_i^t(x)$, using the fact that $a_i^t = \int f_t(x) e_i^t(x) \mu(\mathrm{d}x)$, $a_i^{nt} = \int f_{nt}(x) e_i^t(x) \mu(\mathrm{d}x)$, and $f_{nt} \to f_t$ in $C(\mathcal{X})$ (i.e., uniform convergence), we know that $\lim_{n \to \infty} a_i^{nt} = a_i^t$. Thus , we have

$$\sum_{i=1}^{N_t} \frac{(a_i^t)^2}{\lambda_i^t} = \sum_{i=1}^{N_t} \lim_{n \to \infty} \frac{(a_i^{nt})^2}{\lambda_i^t} \leqslant \lim_{n \to \infty} \sum_{i=1}^{N_t} \frac{(a_i^{nt})^2}{\lambda_i^t}.$$

Using the inequality above, we have

$$\sum_{t=1}^{T}\|f_t\|_{\kappa_t}^2=\sum_{t=1}^{T}\sum_{i=1}^{N_t}\frac{(a_i^t)^2}{\lambda_i^t}=\sum_{t=1}^{T}\sum_{i=1}^{N_t}\varliminf_{n\to\infty}\frac{(a_i^{nt})^2}{\lambda_i^t}\leqslant\varliminf_{n\to\infty}\sum_{t=1}^{T}\sum_{i=1}^{N_t}\frac{(a_i^{nt})^2}{\lambda_i^t}\leqslant r.$$

Thus, we know $\vec{f}\in\mathcal{B}_r$. We proved that $\mathcal{B}_r$ is a compact subset of $C(\mathcal{X})$. □

Lemma .2. *Given a Mercer kernel κ, the corresponding reproducing kernel Hilbert space is given by $\mathcal{H}$. Suppose that the eigenvalues and the orthonormal basis associated with κ are given by $\{\lambda_i\}_{i=1}^N$ and $\{e_i(x)\}_{i=1}^N$, respectively. For any $f\in\mathcal{H}$ and $f(x)=\sum_{i=1}^N a_ie_i(x)$, there exists a function $g\in\mathcal{L}^2(\mathcal{X},\mu)$ such that $f(x)=\int g(y)\kappa(x,y)\mu(\mathrm{d}y)$ if and only if*

$$\sum_{i=1}^{N}\frac{a_i^2}{\lambda_i^2}<\infty.$$

This lemma is given by [6].

With the two lemmas above, we can prove Theorem 4.1:

Proof. To prove the existence of the minimizer of optimization problem (1) (denote the objective function as $H(\vec{f})$), we equivalently consider the following optimization problem:

$$\begin{aligned}&\operatorname*{arg\,min}_{f_t\in\mathcal{H}_t}\frac{1}{n}\sum_{t=1}^{T}\sum_{i=1}^{n}L(x_i,y_{ti},f_t)+\frac{\gamma_2}{2}\sum_{s,t=1}^{T}\delta(s,t)\int(f_s(x)-f_t(x))^2\mu(\mathrm{d}x)\\&\sum_{t=1}^{T}\|f_t\|_{\kappa_t}^2\leqslant M,\end{aligned}\tag{5}$$

where M is a constant related to γ_1. From Lemma .1, we know that $\{\vec{f}|f_t\in\mathcal{H}_t,\sum_{t=1}^T\|f_t\|_{\kappa_t}^2\leqslant M\}$ is compact in $C(\mathcal{X})$. Since $L(\cdot,\cdot,\cdot)$ is continuous, the objective function (5) is continuous in $C(\mathcal{X})$. Thus, the minimizer exists. We denote it as $\vec{f}^\star$.

To get the form of $\vec{f}^\star$, we first assume that L is differentiable. The "differentiable" condition can ultimately be eliminated by approximating a non-differentiable function appropriately and passing to the limit. By using Theorem 3.1, $f_t^\star=\sum_j a_j^{t\star}e_j^t(x)$. Substituting this formula to $H(\vec{f})$ and differentiating $H(\vec{f})$ with respect to $a_j^{t\star}$, we have:

$$\begin{aligned}\frac{\partial H(\vec{f}^\star)}{\partial a_j^{t\star}}&=\frac{1}{n}\sum_{i=1}^{n}\partial_3L(x_i,y_{ti},\sum_k a_k^{t\star}e_k^t(x))e_j^t(x_i)+\frac{2\gamma_1a_j^{t\star}}{\lambda_j^t}\\&+2\gamma_2\int\sum_{s=1}^{T}\delta(t,s)(f_t^\star(x)-f_s^\star(x))e_j^t(x)\mu(\mathrm{d}x)=0.\end{aligned}$$

Thus,

$$a_j^{t\star}=-\frac{\lambda_j^t}{2n\gamma_1}\sum_{i=1}^{n}\partial_3L(x_i,y_{ti},\sum_k a_k^{t\star}e_k^t(x))e_j^t(x_i)-\frac{\gamma_2}{\gamma_1}\lambda_j^t\int\sum_{s=1}^{T}\delta(t,s)(f_t^\star(x)-f_s^\star(x))e_j^t(x)\mu(\mathrm{d}x).$$

Using the equation above, we have

$$f_t^\star(x) = \sum_j -\frac{\lambda_j^t}{2n\gamma_1} \sum_{i=1}^n \partial_3 L(x_i, y_{ti}, \sum_k a_k^{t\,\star} e_k^t(x))e_j^t(x_i)e_j^t(x)$$

$$- \sum_j \frac{\gamma_2}{\gamma_1}\lambda_j^t \int \sum_{s=1}^T \delta(t,s)(f_t^\star(x) - f_s^\star(x))e_j^t(x)\mu(\mathrm{d}x)e_j^t(x).$$

By defining $-\frac{1}{2n\gamma_1}\partial_3 L(x_i, y_{ti}, \sum_k a_k^{t\,\star} e_k^t(x))$ as α_{ti}, the equation above can be represented as

$$f_t^\star(x) = \sum_j \sum_{i=1}^n \alpha_{ti}\lambda_j^t e_j^t(x_i)e_j^t(x) - \sum_j \frac{\gamma_2}{\gamma_1}\lambda_j^t \int \sum_{s=1}^T \delta(t,s)(f_t^\star(x) - f_s^\star(x))e_j^t(x)\mu(\mathrm{d}x)e_j^t(x)$$

$$= \sum_{i=1}^n \alpha_{ti}\kappa_t(x_i, x) - \sum_j \frac{\gamma_2}{\gamma_1}\lambda_j^t \int \sum_{s=1}^T \delta(t,s)(f_t^\star(x) - f_s^\star(x))e_j^t(x)\mu(\mathrm{d}x)e_j^t(x).$$

Finally, since $\sum_{s=1}^T \delta(t,s)(f_t^\star(x) - f_s^\star(x)) \in \mathcal{L}^2(\mathcal{X}, \mu)$, we have

$$\sum_j \frac{[\lambda_j^t \int \sum_{s=1}^T \delta(t,s)(f_t^\star(x) - f_s^\star(x))e_j^t(x)\mu(\mathrm{d}x)]^2}{{\lambda_j^t}^2} = \sum_j [\int \sum_{s=1}^T \delta(t,s)(f_t^\star(x) - f_s^\star(x))e_j^t(x)\mu(\mathrm{d}x)]^2$$

$$= \int [\sum_{s=1}^T \delta(t,s)(f_t^\star(x) - f_s^\star(x))]^2 \mu(\mathrm{d}x) < \infty.$$

By using Lemma .2, we know that there is a $\theta_t(x) \in \mathcal{L}^2(\mathcal{X}, \mu)$ such that

$$f_t^\star(x) = \sum_{i=1}^n \alpha_{ti}\kappa_t(x_i, x) + \int \theta_t(y)\kappa_t(x, y)\mu(\mathrm{d}y),$$

and we obtain the conclusion of the representer theorem. □

Appendix B.

Proofs of Theorems Related to Convergence Theorem
Proof of Theorem 4.2
Proof. $h(\mathcal{F}, X) = \sup\limits_{f\in\mathcal{F}} |Ef^2 - \frac{1}{n}\sum_{i=1}^n f^2(x_i)|$, where $X = \{x_j\}_{j=1}^n$, $\mathcal{F} = \{f | f = \sum_{t=1}^T f_t, f_t \in \mathcal{H}_t, \|f_t\|_{\kappa_t} \leqslant R^*\}$. Consider $X' = \{x_j\}_{j=1}^{i-1} \cup \{x_j\}_{j=i+1}^n \cup \{x_i'\}$, since $\forall \epsilon > 0$, there is a function $f \in \mathcal{F}$ such that $h(\mathcal{F}, X) \leqslant |Ef^2 - \frac{1}{n}\sum_{i=1}^n f^2(x_i)| + \epsilon$, we have

$$h(\mathcal{F}, X) - h(\mathcal{F}, X') \leqslant |Ef^2 - \frac{1}{n}\sum_{j=1}^{i-1} f^2(x_j) - \frac{1}{n}\sum_{j=i+1}^n f^2(x_j) - \frac{1}{n}f^2(x_i)|$$

$$- |Ef^2 - \frac{1}{n}\sum_{j=1}^{i-1} f^2(x_j) - \frac{1}{n}\sum_{j=i+1}^n f^2(x_j) - \frac{1}{n}f^2(x_i')| + \epsilon$$

$$\leqslant \frac{1}{n}[f^2(x_i) + f^2(x_i')] + \epsilon$$

Since

$$|f(x)| \leqslant \sum_{t=1}^{T} |f_t(x)| = \sum_{t=1}^{T} |\langle f_t, \kappa_t(\cdot, x)\rangle_{\kappa_t}| \leqslant \sum_{t=1}^{T} \|f_t\|_{\kappa_t} \sqrt{B} \leqslant TR^* \sqrt{B},$$

we have

$$h(\mathcal{F}, X) - h(\mathcal{F}, X') \leqslant \frac{2}{n} T^2 R^{*2} B + \epsilon.$$

Let ϵ be a small number close to zero, we have

$$h(\mathcal{F}, X) - h(\mathcal{F}, X') \leqslant \frac{2}{n} T^2 R^{*2} B.$$

Using the same technique, we know

$$h(\mathcal{F}, X') - h(\mathcal{F}, X) \leqslant \frac{2}{n} T^2 R^{*2} B.$$

Thus, $|h(\mathcal{F}, X') - h(\mathcal{F}, X)| \leqslant \frac{2}{n} T^2 R^{*2} B$. According to the McDiarmid inequality [5], for any $\epsilon > 0$,

$$P(|h(\mathcal{F}, X) - Eh(\mathcal{F}, X)| > \epsilon) \leqslant 2exp(-\frac{\epsilon n}{2T^4 R^{*4} B^2}).$$

Given $0 < \delta < 1$, let $2exp(-\frac{\epsilon n}{2T^4 R^{*4} B^2}) = \delta$, we have

$$\epsilon = T^2 R^{*2} B \sqrt{\frac{2 \log \frac{2}{\delta}}{n}}.$$

We get the conclusion given by Theorem 4.2. □

Proof of Theorem 4.3

To prove this theorem, we need a lemma given by Theorem 12 in [5]:

Lemma .3. *For $1 \leq q < \infty$, let $\mathcal{L}_{\mathcal{F},h,q} = \{|f - h|^q : f \in \mathcal{F}\}$, where h and $|f - h|$ are uniformly bounded. We have*

$$R_n(\mathcal{L}_{\mathcal{F},h,q}) \leq 2q|f - h|_\infty (R_n(\mathcal{F}) + \frac{|h|_\infty}{\sqrt{n}}),$$

where the Rademacher complexity $R_n(\mathcal{F})$ is defined as

$$R_n(\mathcal{F}) = E_{X,\sigma}(\sup_{f \in \mathcal{F}} |\frac{2}{n} \sum_{i=1}^{n} \sigma_i f(x_i)|).$$

Here $\sigma_i, i = 1, \cdots, n$ are independent ± 1-valued uniform random variables.

With the lemma above, we can prove Theorem 4.3:

Proof.

$$
\begin{aligned}
E_X h(\mathcal{F}, X) &= E_X\left(\frac{1}{n}\sup_{f\in\mathcal{F}}|E_{\tilde{X}}\left(\sum_{i=1}^{n} f^2(\tilde{x}_i)\right) - \sum_{i=1}^{n} f^2(x_i)|\right) \\
&\leqslant E_{X,\tilde{X}}\left(\frac{1}{n}\sup_{f\in\mathcal{F}}\sum_{i=1}^{n}|f^2(x_i) - f^2(\tilde{x}_i)|\right) \\
&= E_{X,\tilde{X}}\left(\frac{1}{n}\sup_{f\in\mathcal{F}}\sum_{i=1}^{n}\sigma_i(f^2(x_i) - f^2(\tilde{x}_i))\right) \\
&\leqslant E_{X,\{\sigma_i\}}\left(\frac{2}{n}\sup_{f\in\mathcal{F}}\sum_{i=1}^{n}\sigma_i f^2(x_i)\right),
\end{aligned}
$$

where $\tilde{X} = \{\tilde{x}_i\}_{i=1}^n$ is i.i.d with X, and $\sigma_i, i = 1, \cdots, n$ are independent ±1-valued uniform random variables.

Using the conclusion given by Lemma .3, let $q = 2, h = 0$, we have

$$
E_X h(\mathcal{F}, X) \leqslant 8|f|_\infty\left(E_{X,\{\sigma_i\}}(\frac{1}{n}\sup_{f\in\mathcal{F}}|\sum_{i=1}^{n} f(x_i)\sigma_i|)\right).
$$

Since $f = \sum_{t=1}^{T} f_t$ and $f_t \in \mathcal{H}_t$, $f_t(x) = \langle w_t, \varphi_t(x)\rangle$, where $\varphi_t(\cdot)$ is the feature mapping with respect to kernel κ_t and w_t is the weight vector. Thus, we have

$$
\begin{aligned}
E_X h(\mathcal{F}, X) &\leqslant 8|f|_\infty\left(E_{X,\{\sigma_i\}}(\frac{1}{n}\sup_{f\in\mathcal{F}}|\sum_{i=1}^{n}\sum_{t=1}^{T}\langle w_t, \sigma_i\varphi_t(x_i)\rangle|)\right) \\
&= 8|f|_\infty\left[E_{X,\{\sigma_i\}}\left(\frac{1}{n}\sup_{f\in\mathcal{F}}|\sum_{t=1}^{T}\langle w_t, \sum_{i=1}^{n}\sigma_i\varphi_t(x_i)\rangle|\right)\right] \\
&\leqslant 8|f|_\infty\left[\sum_{t=1}^{T} E_{X,\{\sigma_i\}}\left(\frac{1}{n}\sup_{f\in\mathcal{F}}|\langle w_t, \sum_{i=1}^{n}\sigma_i\varphi_t(x_i)\rangle|\right)\right] \\
&\leqslant 8|f|_\infty\left[\sum_{t=1}^{T} E_{X,\{\sigma_i\}}\left(\frac{1}{n}\sup_{f\in\mathcal{F}}\|w_t\|\sqrt{\sum_{i,j=1}^{n}\sigma_i\sigma_j\kappa_t(x_i, x_j)}\right)\right].
\end{aligned}
$$

Since $\|w_t\| = \|f_t\|_{\kappa_t} \leqslant R^*$ and $\max_{1\leqslant t\leqslant T}\sup_{x\in\mathcal{X}}\kappa_t(x, x) \leqslant B$, we have

$$
\begin{aligned}
E_X h(\mathcal{F}, X) &\leqslant \frac{8R^*|f|_\infty}{n}\sum_{t=1}^{T} E_{X,\{\sigma_i\}}\sqrt{\sum_{i,j=1}^{n}\sigma_i\sigma_j\kappa_t(x_i, x_j)} \\
&\leqslant \frac{8R^*|f|_\infty}{n}\sum_{t=1}^{T}\sqrt{E_{X,\{\sigma_i\}}\sum_{i,j=1}^{n}\sigma_i\sigma_j\kappa_t(x_i, x_j)} \\
&\leqslant \frac{8TR^*\sqrt{B}|f|_\infty}{\sqrt{n}}.
\end{aligned}
$$

From the proof of Theorem 4.2, we know $|f|_\infty \leqslant TR^* \sqrt{B}$, thus, we finally have

$$E_X h(\mathcal{F}, X) \leqslant \frac{8T^2 R^{*2} B}{\sqrt{n}}.$$

□

Proof of Theorem 4.5 (Convergence Theorem)
Proof. For ease of explanation, we define

$$H(\vec{f}) = \frac{1}{n}\sum_{t=1}^{T}\sum_{i=1}^{n} L(x_i, y_{ti}, f_t) + \gamma_1 \sum_{t=1}^{T} \|f_t\|_{\kappa_t}^2 + \frac{\gamma_2}{2}\sum_{s,t=1}^{T} \delta(s,t) \int (f_s(x) - f_t(x))^2 P(\mathrm{d}x)$$

$$\hat{H}(\vec{f}) = \frac{1}{n}\sum_{t=1}^{T}\sum_{i=1}^{n} L(x_i, y_{ti}, f_t) + \gamma_1 \sum_{t=1}^{T} \|f_t\|_{\kappa_t}^2 + \frac{\gamma_2}{2n}\sum_{s,t=1}^{T} \delta(s,t) \sum_{i=1}^{n}(f_s(x_i) - f_t(x_i))^2.$$

Suppose $\Delta f_t = -f_t^\star + \hat{f}_t$. Since $\vec{\hat{f}} = (\hat{f}_1, \hat{f}_2, \cdots, \hat{f}_T)$ is the minimizer of $\hat{H}(\vec{f})$, by using Theorem 4.4, with probability more than $1 - T(T-1)\delta/2$, we have

$$\hat{H}(\vec{\hat{f}}) \leqslant \hat{H}(\vec{f^\star}) \leqslant H(\vec{f^\star}) + \gamma_2 \frac{T(T-1)}{2} g(n). \tag{6}$$

On the other hand, by Theorem 3.1, we know $\hat{f}_t(x) = \sum_{i=1}^{N_t} \hat{a}_i^t e_i^t(x)$ and $f_t^\star(x) = \sum_{i=1}^{N_t} a_i^{t\star} e_i^t(x)$, where $\{e_i^t(x)\}_{i=1}^{N_t}$ is the orthornormal basis of square integrable space associated with kernel κ_t, and N_t is finite or infinite. Since $\vec{f^\star}$ is the minimizer of $H(\vec{f})$, and $H(\vec{f})$ can also be viewed as a function of $\{a_i^t\}$, by taking Taylor expansion of $H(\vec{f})$ on $\vec{f^\star}$, we have:

$$H(\vec{\hat{f}}) = H(\vec{f^\star}) + \frac{1}{2}\sum_{s,t=1}^{T}\sum_{i=1}^{N_t}\sum_{j=1}^{N_s} \partial^2_{a_i^t, a_j^s} H(\{a_i^{t\prime}\})(\hat{a}_i^t - a_i^{t\star})(\hat{a}_j^s - a_j^{s\star}),$$

where $a_i^{t\prime} = \lambda \hat{a}_i^t + (1-\lambda) a_i^{t\star}$, $\lambda \in [0,1]$.(Note that the gradient of $H(\vec{f})$ vanishes on $\vec{f^\star}$ since it is the minimizer.) Since L is convex, by property of convex function, it is easy to show that

$$\frac{1}{2}\sum_{s,t=1}^{T}\sum_{i=1}^{N_t}\sum_{j=1}^{N_s} \partial^2_{a_i^t, a_j^s} H(\{a_i^{t\prime}\})(\hat{a}_i^t - a_i^{t\star})(\hat{a}_j^s - a_j^{s\star}) \geqslant \gamma_1 \sum_{t=1}^{T}\sum_{i=1}^{N_t} \frac{(\hat{a}_i^t - a_i^{t\star})^2}{\lambda_i^t} = \gamma_1 \sum_{t=1}^{T} \|\Delta f_t\|_{\kappa_t}^2.$$

By using Theorem 4.4 and the inequality above, with probability more than $1 - T(T-1)\delta/2$, we have

$$\hat{H}(\vec{\hat{f}}) \geqslant H(\vec{\hat{f}}) - \gamma_2 \frac{T(T-1)}{2} g(n) \geqslant H(\vec{f^\star}) + \gamma_1 \sum_{t=1}^{T} \|\Delta f_t\|_{\kappa_t}^2 - \gamma_2 \frac{T(T-1)}{2} g(n). \tag{7}$$

Combining inequalities (6) and (7), we finally have

$$D_t = \sup_{x \in \mathcal{X}} |f_t^\star(x) - \hat{f}_t(x)| \leqslant \|\Delta f_t\|_{\kappa_t} \sqrt{B} \leqslant \sqrt{\frac{T(T-1)\gamma_2 g(n)}{\gamma_1}} \sqrt{B} = O(1/n^{\frac{1}{4}}).$$

□

Appendix C.

Proof of Theorem 4.6

Proof. According to Theorem 3.1, suppose that the eigenvalues and the orthonormal basis of $\mathcal{L}^2(\mathcal{X},\mu)$ associated with kernel κ are given by $\{\lambda_i\}_{i=1}^N$ and $\{e_i(x)\}_{i=1}^N$, respectively. Given s,t, $f_s(x)=\sum_{i=1}^N a_i^s e_i(x)$ and $f_t(x)=\sum_{i=1}^N a_i^t e_i(x)$.

$$\|f_t - f_s\|_\kappa^2 = \sum_{i=1}^N \frac{(a_i^t - a_i^s)^2}{\lambda_i}.$$

Since $\lambda_1 \geqslant \lambda_2 \geqslant \cdots$,

$$\|f_t - f_s\|_\kappa^2 \geqslant \sum_{i=1}^N \frac{(a_i^t - a_i^s)^2}{\lambda_1}.$$

On the other hand,

$$\int (f_t(x) - f_s(x))^2 \mu(\mathrm{d}x) = \sum_{i=1}^N (a_i^t - a_i^s)^2.$$

Thus, we know

$$\|f_t - f_s\|_\kappa^2 \geqslant \frac{1}{\lambda_1} \int (f_t(x) - f_s(x))^2 \mu(\mathrm{d}x), \quad \forall s,t.$$

We can take $C(\kappa,\mu)$ as $\frac{1}{\lambda_1}$. Moreover, if $\kappa(x,y) \geqslant 0$ and $\int \kappa(x,y)\mu(\mathrm{d}x) = 1$, we assert that $\lambda_1 = 1$.

Since $\int \kappa(x,y)\mu(\mathrm{d}x) = 1$, 1 is an eigenvalue and the corresponding eigenfunction is 1. We only have to prove that $\lambda_1 \leqslant 1$. Since $\mathcal{X}$ is compact and $e_1(x)$ is continuous, we can define

$$x_0 = \arg\max_{x\in\mathcal{X}} |e_1(x)|.$$

$|e_1(x_0)| > 0$. Since

$$\int \kappa(x,x_0) e_1(x)\mu(\mathrm{d}x) = \lambda_1 e_1(x_0),$$

we have

$$\lambda_1 |e_1(x_0)| = |\int \kappa(x,x_0)e_1(x)\mu(\mathrm{d}x)| \leqslant \int \kappa(x,x_0)|e_1(x)|\mu(\mathrm{d}x) \leqslant \int \kappa(x,x_0)\mu(\mathrm{d}x)|e_1(x_0)| = |e_1(x_0)|.$$

This means

$$\lambda_1 \leqslant 1.$$

Thus, we know that we can take $C(\mathcal{X},\mu) \leqslant 1$. □