Multi-Task Learning in Square Integrable Space

Wei Wu
MOE-Microsoft Key Laboratory of Statistics
and Information Technology, Peking University
v-wew@microsoft.com

Hang Li, Yunhua Hu
Microsoft Research Asia
hangli@microsoft.com
yuhu@microsoft.com

Rong Jin
Computer Science and Engineering
Michigan State University
rongjin@cse.msu.edu

Abstract

Several kernel based methods for multi-task learning have been proposed, which leverage relations among tasks as regularization to enhance the overall learning accuracies. These methods assume that the tasks share the same kernel, which could limit their applications because in practice different tasks may need different kernels. The main challenge of introducing multiple kernels into multiple tasks is that models from different Reproducing Kernel Hilbert Spaces (RKHSs) are not comparable, making it difficult to exploit relations among tasks. This paper addresses the challenge by formalizing 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 an existing method. We also give an SVM based implementation of our method for multi-label classification. Experiments on two real-world data sets show that the proposed method performs better than the existing method.

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 to enhance the performance of learning (cf., (Caruana 1997)). Evigeniou et al. (2006), as well as Kato et al. (2008), proposed methods which incorporate task relations into 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.

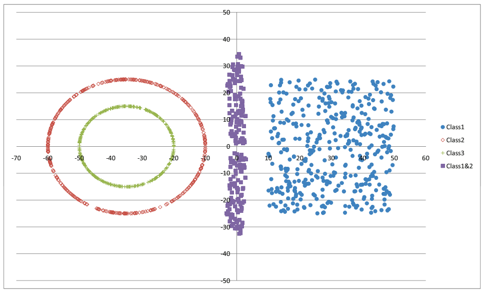

Figure 1: Artificial Data

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 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).

In this paper, we consider exploiting task relations for multi-task learning by using different kernels for different tasks. The main challenge is that models for different tasks may be defined in different RKHSs, which are not comparable. We formulate multi-kernel multi-task learning in the Square Integrable Space (SIS). Since SIS includes RKHSs for different tasks as subspaces, the task relations can be naturally incorporated into the regularization term defined in the SIS. We then present a 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.

Our approach is a natural extension of the approach proposed by Evgeniou et al. (2006) for single kernel multi-task learning using graph regularization. We specifically show the relationship between our method and Evgeniou etl al.'s method, when only a single kernel is employed for multiple tasks in our method. We give a specific algorithm of our method based on SVM technique. Experiments of multi-label classification on two real-world data sets show that our approach performs better than the existing method for 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.

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.,(Caruana 1997)). One group of methods attempt to use task relations. For example, Evgeniou et al. (2006), and Kato et al.(2008) 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, Ando & Zhang (2005), as well as Argyriou et al. (2007) 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 (Schölkopf and Smola 2002). 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 (c.f., (Lanckriet et al. 2002; Bach, Lanckriet, and Jordan 2004)). 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. Tang et al. (2009) proposed a method of simultaneously learning multiple kernels for multiple tasks, but they did not utilize task relations. Ji et al. (2009) 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. Duan et al. (2009) 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

In this section, we define notations used in this paper and introduce some background knowledge.

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 consider multi-label classification. We assume that $\mathcal{X}_t$=$\mathcal{X}$ for all tasks and x_t is independent from t. $\mathcal{X}$ is a compact subset in $\mathbb{R}^d$ and $\mathcal{Y}_t = \{+1, -1\}$. Moreover, we have a training data set $S_t = \{(x_i, y_{ti})\}_{i=1}^n$ for each task t and our goal is to learn a classifier $f_t(\cdot)$: $\mathcal{X} \to \mathbb{R}$ that can assign a label to a new instance x. Different tasks share a common marginal distribution $P(x)$ but have different conditional distributions $P_t(y|x)$. In MKMT, the function spaces $\mathcal{F}_t$ ($f_t(\cdot) \in \mathcal{F}_t$) of different tasks are assumed to be different.

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 for multi-label classification in Section 6. We use Δ in learning of the classifiers $\{f_t(\cdot)\}_{t=1}^T$.

RKHS and Square Integrable Space

We briefly review the theory on Reproducing Kernel Hilbert Space (RKHS) (Aronszajn 1950) and show the relationship between RKHS and Square Integrable Space (SIS).

A Hilbert space $\mathcal{H}$ of functions $f(\cdot) : \mathcal{X} \to \mathbb{R}$ is an RKHS if and only if there is a function $K(\cdot, \cdot) : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ that satisfies 1) $K(x, y) = K(y, x)\ \forall x, y \in \mathcal{X}$; 2) $\forall \{\alpha_i\}_{i=1}^n \subset \mathbb{R}, \{x_i\}_{i=1}^n \subset \mathcal{X}$, $\sum_{i,j=1}^n \alpha_i\alpha_j K(x_i, x_j) \geqslant 0$ such that $f(x) = \langle f(\cdot), K(x, \cdot)\rangle_{\mathcal{H}}$ $\forall x$. $K(\cdot, \cdot)$ is called 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 $f(\cdot)$ satisfies $\int f^2(x)\mu(\mathrm{d}x) < \infty$. According to Mercer Theorem (Cucker and Smale 2002), there is an orthonormal basis $\{e_i(\cdot)\}$ of $\mathcal{L}^2(\mathcal{X}, \mu)$ associated with K and $K(x, y) = \sum_i \lambda_i e_i(x)e_i(y)\ \forall x, y \in \mathcal{X}$. $\forall i$, $\lambda_i \geqslant 0$ is the i-th eigenvalue and $e_i(\cdot)$ is continuous.

The following theorem (Cucker and Smale 2002) unveils that for any Mercer kernel K, the RKHS $\mathcal{H}_K$ is a subspace of Square Integrable Space.

Theorem 3.1. $\mathcal{H}_K = \{f \in \mathcal{L}^2(\mathcal{X}, \mu) | f(\cdot) = \sum_{i=1}^\infty a_i e_i(\cdot)$, $\sum_{i=1}^\infty \frac{a_i^2}{\lambda_i} < \infty\}$. $\forall\ f \in \mathcal{H}_K$, $||f||_{\mathcal{H}_K}^2 = \sum_{i=1}^\infty \frac{a_i^2}{\lambda_i}$, *and* f *is continuous on* $\mathcal{X}$.

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, $f_1, f_2, \ldots, f_T$ 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 $\mathcal{L}^2(\mathcal{X}, P(x))$, where $P(x)$ is the marginal distribution on $\mathcal{X}$ (We specifically let μ in Theorem 3.1 be $P(x)$ in this paper).

One advantage of the approach is that we can naturally use task relations in $\mathcal{L}^2(\mathcal{X}, P(x))$, 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.

Ideal Solution

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

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

where the second term is a normal regularization term which controls the complexity of models in their own RKHSs, and the third term is a regularization term which measures difference of models in the common space (i.e.,$\mathcal{L}^2(\mathcal{X}, P(x))$). 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 $L(\cdot,\cdot)$ is non-negative, convex and continuous, the solution to the optimization problem* (1) *exists and has the following form:*

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

where $\alpha_{ti} \in \mathbb{R}$ and $\theta_t(\cdot) \in \mathcal{L}^2(\mathcal{X}, P(x))$.

Formula (2) offers an ideal solution. The proof sketch of Theorem 4.1 is given in Appendix A.

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

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

We need to answer the question whether the practical solution converges to the ideal solution when the training data size goes to infinity. 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.

Convergence of Practical Solution

In this section, we further assume that L is differentiable and prove the convergence of practical solution under the condition [2]. Theorem 4.5 gives a bound on the difference between the two solutions. The result indicates that the practical solution (3) converges to the ideal solution (2) in probability. We briefly explain how to obtain the bound and present the proof of Theorem 4.5 in Appendix B.

Define $D_t = \sup_{x \in \mathcal{X}} |f_t^\star(x) - \hat{f}_t(x)|$. Suppose that $\max_{1 \leqslant t \leqslant T} \sup_{x \in \mathcal{X}} \quad \kappa_t(x,x) \leqslant B$. 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 $\{f_t^\star(\cdot)\}_{t=1}^{T}$ minimize (1), $\sum_{t=1}^{T} \|f_t^\star\|_{\kappa_t}^2 \leqslant \frac{1}{n\gamma_1} \sum_{t=1}^{T} \sum_{i=1}^{n} L(0, y_{ti}) \leqslant \frac{TU}{\gamma_1}$, where $L(0,y) \leqslant U$. 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 (Bartlett and Mendelson 2002), we obtain the following lemma:

Lemma 4.2. *Given an arbitrary small positive number δ, with probability at least $1-\delta$, 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 lemma, we only need to bound $Eh(\mathcal{F}, X)$. Using the conclusion (Theorem 12.6) and some proof techniques (proof of Theorem 8 and Lemma 22) in (Bartlett and Mendelson 2002), we obtain

Lemma 4.3.

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

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

Theorem 4.4. *With probability at least $1 - \delta$, 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:

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

[2] 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 (e.g., Weierstrass Approximation Theorem).

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

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

Relation with Existing Approach

Evgeniou et al. (2006) exploit task relations (i.e. Δ) to conduct multi-task learning through the following optimization problem :

$$\begin{aligned} \arg\min_{\{f_t\} \subset \mathcal{H}} \ & \frac{1}{n} \sum_{t=1}^{T} \sum_{i=1}^{n} L(f_t(x_i), y_{ti}) + \gamma_1 \sum_{t=1}^{T} \|f_t\|_\kappa^2 \\ & + \frac{\gamma_2}{2} \sum_{s,t=1}^{T} \delta(s,t) \|f_s - f_t\|_\kappa^2. \end{aligned} \quad (4)$$

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. Our approach (1) extends their approach and can handle both SKMT and MKMT cases. When all tasks share the same kernel, the following theorem shows the relationship between the two regularization terms of our approach and Evgeniou et al.'s approach:

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

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

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

The theorem 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 as hinge loss. Moreover, we also add a bias b_t into each classifier $f_t(\cdot)$ following the convention. By introducing slack variables, we can obtain the primal problem:

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

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 $(\mathcal{K} + \frac{\gamma_2}{\gamma_1'}\mathcal{K}(\mathcal{L} \otimes I)\mathcal{K})^{-1}$
5: Compute $\beta^*$ by solving the dual problem
6: Compute $\alpha'^*$ by using equation (5).
7: Output: $f_t^*(\cdot) = \sum_{t=1}^n \alpha_{ti}'^* \kappa_t(x_i, \cdot) + b_t^*$, $1 \leqslant t \leqslant T$
```

Combining α_{ti} and $\frac{1}{n}\theta_{ti}$ in Equation (3) as α_{ti}' and substituting the solution given by Equation (3) into the primal problem, we obtain the dual problem:

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

Here, $\beta = (\beta_1^\top, \beta_2^\top, \ldots, \beta_T^\top)^\top$ is the dual variable where $\beta_t = (\beta_{t1}, \beta_{t2}, \ldots, \beta_{tn})^\top$ $t = 1, 2 \ldots T$. Y is a diagonal matrix whose $t \times i$-th element is y_{ti}. $\mathcal{L}$ is the 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. I is an identity matrix. $\mathcal{K}$ is a block diagonal matrix with each block $\mathcal{K}_t = (\kappa_t(x_i, x_j))_{n \times n}$.

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

$$\alpha'^* = \frac{1}{2\gamma_1'} (\mathcal{K} + \frac{\gamma_2}{\gamma_1'} \mathcal{K}(\mathcal{L} \otimes I)\mathcal{K})^{-1} \mathcal{K} Y \beta^*. \quad (5)$$

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 (Elisseeff and Weston 2002)), which is a gradient descent based method. At step 4, we need to compute the inverse of a matrix, which is of order $O((Tn)^3)$. 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 on multi-label classification to verify the effectiveness of our approach. We used two real-world classification data sets: protein data and music data. 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 (4). We denote our method Multiple (Kernel). As evaluation measure, we utilized ROC score (Lanckriet et al. 2004).

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 co-occurrence of tasks (classes) as similarity between them.

Both of the two real-world data sets are on multi-label classification. The first data set [3] is on classification of protein functions, which contains 3,588 proteins with 13 function classes. The average number of functions per protein is 1.53. For more details about the data set, see (Lanckriet et al. 2004). The second data set [4] is on music emotion classification, which contains 593 pieces of music and 6 types of emotion. The average number of emotion types per piece of music is 1.87.

we randomly chose 500 instances for protein data and 100 instances for music data as training data, respectively, and evenly divided the rest into two subsets as validation and testing data for both data sets. We used the heuristic method to calculate task similarities from the training data.

For protein data, no information on features is available, and the kernels are provided as kernel matrices. There are in total 8 kernel matrices (8 kernels). For music data, we created Gaussian kernels ($e^{-\sigma\|x-y\|^2}$) 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 $\{0.1, 0.5, 1, 5\}$, and γ_2 from $\{0.01, 0.05, 0.1, 0.5, 1, 5\}$.

Kernel selection and parameter tuning were conducted using the validation data. First, kernels were selected and parameter γ_1' was tuned for Individual. Next, the kernels and parameter γ_1' were fixed, and parameter γ_2 was tuned for Multiple. Finally, the kernels of Individual were used, and the two parameters of γ_1' and γ_2 were simultaneously tuned for Single.

We repeated the above process ten times, and the final results are averaged over ten trials. Table 1 gives the results of three methods on two data sets. On protein data, 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}). 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. On music data, 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 stand for that the improvements of our methods are statistically significant over Individual, Single using Gaussian kernel and Single using polynomial kernel, respectively.

From the results in Table 1, we can see that kernel selection is crucial for Single. With a good kernel selection, SKMT's performance can be comparable to Multiple, 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 an overall good performance.

Table 1: Comparison of three methods on two real-world data sets

Comparison of three methods on protein data				
	Individual	Single (K_{sw})	Single (K_{pfam_E})	Multiple
class 1	0.721	0.697	0.736	**0.740**^{1,2}
class 2	**0.666**	0.652	0.638	0.657^{3}
class 3	0.654	0.655	0.653	**0.676**^{1,3}
class 4	0.728	0.745	0.741	**0.753**^{1,3}
class 5	0.779	0.790	0.779	**0.797**^{1,3}
class 6	0.682	0.650	0.686	**0.688**^{2}
class 7	0.671	0.653	0.673	**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	**0.651**^{3}
class 11	0.541	**0.557**	0.542	0.541
class 12	0.881	0.826	0.887	**0.891**^{1,2,3}
class 13	0.579	0.590	0.594	**0.602**^{1}
Average	0.676	0.668	0.674	**0.687**^{1,2,3}

Comparison of three methods on music data				
	Individual	Single (Gaussian)	Single (Polynomial)	Multiple
class 1	0.723	0.675	0.727	**0.732**^{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**^{2}
class 5	0.748	0.653	0.755	**0.760**^{1,2}
class 6	0.736	0.706	0.747	**0.752**^{1,2}
Average	0.733	0.695	0.720	**0.743**^{1,2,3}

7 Conclusion and Future Work

We have proposed a new kernel approach to conducting multi-task learning in the Square Integrable Space in order to simultaneously exploit multiple kernels and task relations. Specifically, we define a new regularization term in SIS to incorporate task relations into the objective function. We have proved a new 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 SVM based algorithm for implementing our approach for multi-label classification, and conducted experiments on two real-world data sets to verify its effectiveness.

As future work, we plan (1) to study principled ways of selecting kernels in our approach, (2) to study principled ways of learning task relations in our approach, and (3) to develop more efficient learning algorithms.

A Proof Sketch of Theorem 4.1

Proof Sketch. We first prove the existence of the minimizer of optimization problem (1) (denote the objective function as $H(\vec{f})$, where $\vec{f}(\cdot) = (f_1(\cdot), \ldots, f_T(\cdot))$). First, we consider the following problem:

$$\arg\min_{f_t \in \mathcal{H}_t} \frac{1}{n} \sum_{t=1}^{T} \sum_{i=1}^{n} L(f_t(x_i), y_{ti}) + \frac{\gamma_2}{2} \sum_{s,t=1}^{T} \delta(s,t) \mathbb{E}_{P(x)}(f_s - f_t)^2$$

$$\text{subject to } \sum_{t=1}^{T} \|f_t\|_{K_t}^2 \leqslant M,$$

where $\mathbb{E}_{P(x)}(f_s - f_t)^2 = \int (f_s(x) - f_t(x))^2 P(\mathrm{d}x)$, and $M \geq 0$ is a constant. We can prove the existence of the minimizer of the objective function (denoted as $V(\vec{f})$), since $\{\vec{f} | f_t \in \mathcal{H}_t, \sum_{t=1}^{T} \|f_t\|_{K_t}^2 \leqslant M\}$ is compact in $C(\mathcal{X})$ (continuous function space). We denote it as $\vec{f}^M$. Then, we can prove that

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

[4] http://mlkd.csd.auth.gr/multilabel.html

when $M \geq 0$, the minimizer of $V(\vec{f}^M) + \gamma_1 M$ exists. This means that the minimizer of Problem (1) also 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 (e.g., Weierstrass Approximation Theorem). By using Theorem 3.1, $f_t^\star(\cdot) = \sum_j a_j^{t\star} e_j^t(\cdot)$. Substituting this formula to $H(\vec{f})$ and differentiating it with respect to $a_j^{t\star}$ (The "convex" condition can guarantee that by letting the first order derivative be zero, we will get the minimal point). By using the conclusion given by Lemma 3 in (Belkin, Niyogi, and Sindhwani 2006), it is easy to get the conclusion of Theorem 4.1. □

B Proof of Theorem 4.5

Proof. For ease of explanation, we define

$$H(\vec{f}) = Z(\vec{f}) + \frac{\gamma_2}{2}\sum_{s,t=1}^{T}\delta(s,t)\int (f_s(x) - f_t(x))^2 P(\mathrm{d}x)$$

$$\hat{H}(\vec{f}) = Z(\vec{f}) + \frac{\gamma_2}{2n}\sum_{s,t=1}^{T}\delta(s,t)\sum_{i=1}^{n}(f_s(x_i) - f_t(x_i))^2,$$

where $Z(\vec{f}) = \frac{1}{n}\sum_{t=1}^{T}\sum_{i=1}^{n} L(f_t(x_i), y_{ti}) + \gamma_1 \sum_{t=1}^{T} \|f_t\|_{\kappa_t}^2$. Suppose $\Delta f_t = -f_t^\star + \hat{f}_t$. Since $\hat{\vec{f}}$ minimize $\hat{H}(\vec{f})$, by using Theorem 4.4, with probability at least $1 - \frac{T(T-1)\delta}{2}$, we have

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

On the other hand, by Theorem 3.1, we know $\hat{f}_t(\cdot) = \sum_{i=1}^{N_t} \hat{a}_i^t e_i^t(\cdot)$ and $f_t^\star(\cdot) = \sum_{i=1}^{N_t} a_i^{t\star} e_i^t(\cdot)$, where $e_i^t(\cdot)$ is associated with kernel κ_t, and $N_t \leqslant \infty$. 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:

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

where $a_i^{t\prime} = \tau \hat{a}_i^t + (1-\tau) a_i^{t\star}$, $\tau \in [0, 1]$.(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, we have

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

where λ_i^t is the i-th eigenvalue of the kernel κ_t. By using Theorem 4.4 and the inequality above, with probability at least $1 - \frac{T(T-1)\delta}{2}$, we have

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

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

$$D_t \leqslant \|\Delta f_t\|_{\kappa_t} \sqrt{B} \leqslant \sqrt{\frac{T(T-1)\gamma_2 g(n)}{\gamma_1}} \sqrt{B} = O(1/n^{\frac{1}{4}}).$$

□

References

Ando, R., and Zhang, T. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *JLMR'05* 6:1817–1853.

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2007. Multi-task feature learning. *NIPS'07* 19:41.

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

Bach, F.; Lanckriet, G.; and Jordan, M. 2004. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML'04*.

Bartlett, P. L., and Mendelson, S. 2002. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR* 3:463–482.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2006. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *JLMR'06* 7:2399–2434.

Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

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

Duan, L.; Tsang, I.; Xu, D.; and Chua, T. 2009. Domain adaptation from multiple sources via auxiliary classifiers. In *ICML'09*.

Elisseeff, A., and Weston, J. 2002. Kernel methods for Multi-labelled classification and Categorical regression problems. In *NIPS'02*.

Evgeniou, T.; Micchelli, C.; and Pontil, M. 2006. Learning multiple tasks with kernel methods. *JLMR'06* 6(1):615.

Ji, S.; Sun, L.; Jin, R.; and Ye, J. 2009. Multi-label multiple kernel learning. *NIPS*.

Kato, T.; Kashima, H.; Sugiyama, M.; and Asai, K. 2008. Multi-task learning via conic programming. *NIPS'08* 20:737–744.

Lanckriet, G. R. G.; Cristianini, N.; Bartlett, P.; Ghaoui, L. E.; and Jordan, M. I. 2002. Learning the kernel matrix with semi-definite programming. In *ICML'02*, 323–330.

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

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

Tang, L.; Chen, J.; and Ye, J. 2009. On Multiple Kernel Learning with Multiple Labels. In *IJCAI'09*.