Absent Multiple Kernel Learning Algorithms

Xinwang Liu, Lei Wang, Xinzhong Zhu, Miaomiao Li, En Zhu, Tongliang Liu, Li Liu, Yong Dou and Jianping Yin

Abstract—Multiple kernel learning (MKL) has been intensively studied during the past few years. It optimally combines multiple channels of each sample to improve classification performance. However, existing MKL algorithms cannot effectively handle the situation where some channels of samples are missing, which is common in practical applications. This paper proposes three absent MKL (AMKL) algorithms to address this issue. Different from existing approaches where missing channels are firstly imputed and then a standard MKL algorithm is deployed on the imputed data, our algorithms directly classify each sample based on its observed channels, without performing imputation. In specific, we define a margin for each sample in its own relevant space, a space corresponding to the observed channels of that sample. The proposed AMKL algorithms then maximize the minimum of all sample-based margins, and this leads to a difficult optimization problem. We first provide two two-step iterative algorithms to approximately solve this problem. After that, we show that this problem can be reformulated as a convex one by applying the representer theorem. This makes it readily be solved via existing convex optimization packages. In addition, we provide a generalization error bound to theoretically back up the proposed AMKL algorithms. Extensive experiments are conducted on nine UCI and six MKL benchmark datasets to compare the proposed algorithms with existing imputation-based methods. As demonstrated, our algorithms achieve superior performance and the improvement is more significant with the increase of missing ratio.

Index Terms—absent data learning, multiple kernel learning, max-margin classification

1 Introduction

Multiple kernel learning (MKL) has been an active topic in machine learning community during the last decade [1]–[11]. By assuming that the optimal kernel can be expressed as a linear combination of a group of pre-specified base kernels, MKL learns the optimal combination coefficient and the structural parameters of support vector machines (SVMs) jointly [3], [12]–[14]. Through this way, MKL not only learns an optimal data-dependent kernel for a specific application, but also provides an elegant framework to integrate heterogeneous data sources for learning. Such merits make MKL widely used in practical applications such as object detection [15], [16], bioinformatics data fusion [17] and signal processing [18], to name just a few.

Research work in the literature has made important progress on improving the efficiency of MKL algorithms [2], [3], [12], [19]–[21], designing non-sparse and non-linear MKL algorithms [12], [22]–[24], developing two-stage MKL algorithms [5], [13], [25] and integrating radius information into traditional margin-based MKL algorithms [26]–[28]. Besides, many novel extensions, including online MKL algorithms [7], MKL algorithms for clustering [17], [29], [30], domain transfer MKL algorithms [11], [31] and sample-adaptive MKL [32]–[34], have been proposed recently, which further expands the application of MKL algorithms.

Existing research work on MKL usually takes the following implicit assumption: the data of all channels are available for every sample. However, this assumption will not hold anymore when some channels of a sample are absent, which is common in a number of practical applications including neuroimaging [35], [36], computational biology [37], medical analysis [38], to name just a few. For example, in predicting Alzheimer's Disease with multiple imaging modalities, subjects may only participate in part of the medical examinations, resulting the information of different modalities missed [35]. Formally speaking, this case is called the missing value problem [38], [39] or absent data learning [37], which has attracted research attention in the literature [35], [37], [39]–[41]. Nevertheless, to the best of our knowledge, designing efficient MKL algorithms to directly handle absent channels has not been sufficiently researched in the literature and remains an open issue.

The violation to the above assumption makes existing MKL algorithms unable to work as usual. Traditionally, the samples with absent channels are discarded, resulting in a severe loss of available information. A straightforward remedy may firstly impute these absent channels with zero (known as zero-filling in the

- *X. Liu, M. Li, E. Zhu and Y. Dou are with College of Computer, National University of Defense Technology, Changsha, China, 410073 (e-mail: xinwangliu@nudt.edu.cn, miaomiaolinudt@gmail.com, enzhu@nudt.edu.cn and yongdou@nudt.edu.cn).*
- *L. Wang is with School of Computing and Information Technology, University of Wollongong, NSW, Australia, 2522. (e-mail: leiw@uow.edu.au).*
- *X. Zhu is with College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua, China, 321004 (e-mail: zxz@zjnu.edu.cn).*
- *T. Liu is with the UBTECH Sydney Artificial Intelligence Centre and the School of Information Technologies in the Faculty of Engineering and Information Technologies at The University of Sydney, J12 Cleveland St, Darlington NSW 2008, Australia (e-mail: tongliang.liu@sydney.edu.au).*
- *L. Liu is with College of System Engineering, National University of Defense Technology, Changsha, China, 410073 and University of Oulu, Finland (e-mail: li.liu@oulu.fi).*
- *J. Yin is with Dongguan University of Technology, Guangdong, China (e-mail: jpyin@dgut.edu.cn).*

literature), the mean value (mean-filling), or by more advanced approaches such as expectation-maximization (EM-filling) [40]. After that, a standard MKL algorithm is then deployed on the imputed data. Such imputation methods could work when missing ratio is small. Nevertheless, they could produce inaccurate imputation when the absence becomes more significant, and this will deteriorate the performance of the subsequent MKL.

Different from the afore-mentioned imputation approaches, this paper proposes to directly classify samples with absent channels without imputation. It is inspired by the concept of sample-based margin developed in [37]. In this paper, we first immigrate the concept of sample-based margin into multiple kernel-induced feature spaces, and propose to maximize the minimum of all the sample-based margins. On one hand, our approach is able to effectively handle the issue of absent channels. On the other hand, it yields a more difficult optimization problem than the one encountered in traditional MKL algorithms. We then develop three optimization algorithms from different perspectives to solve this optimization problem. The main contributions of this paper are highlighted as follows:

- Our work extends existing MKL framework by enabling it to directly handle samples with absent channels, which widens the application scope of existing MKL algorithms.
- A two-step iterative MKL (I-AMKL) algorithm is developed to iteratively solve the corresponding optimization problem in the dual space. After that, a new variant, termed I-AMKL-λ, is proposed to improve its theoretical convergence.
- By proving the validity of the representer theorem [42] for our optimization problem, we reformulate it as a convex absent MKL (C-AMKL) in its primal space. It can be readily solved by off-the-shelf convex optimization packages such as CVX [43].
- We provide a generalization error bound to theoretically back up the proposed absent MKL algorithms.
- We conduct extensive experiments to compare the proposed algorithms with existing imputation-based methods on nine UCI and six MKL benchmark datasets, with various missing ratios. The results clearly verify the superiority of the proposed algorithms, especially in the presence of intensive absence of channels.

We end up this section by discussing the differences between our work and the work in [37]. Both papers classify samples with missing observations by maximizing the sample-based margin. However, they have the following important differences: (1) Our work deals with the case in which *channels* (i.e., a group of features) of samples are absent, while the work in [37] studies the problem that *features* of samples are missing. Form this perspective, the work in [37] is a special case of our work, when each individual feature is viewed as a channel and a linear kernel is applied to all channels; (2) Our algorithms are able to work on the absence of both input features and entries of kernel matrices. However, the algorithm in [37] does not study the absence of kernel matrices at all, while only considering the absence of input features; and (3) In addition to develop a two-step iterative algorithm to solve the resultant optimization problem as what has been done in [37], we design a new variant with proved convergence and another new convex algorithm that directly solves the problem in the primal space by employing the representer theorem [42]. More importantly, the newly proposed algorithms consistently achieve significant improvements over the former, as validated by our experimental results.

2 Related Work

2.1 The Sample-based Margin

The sample-based margin is firstly proposed in the seminal work [37] and applied to absent data learning where some features of a sample are missing. An important assumption for the sample-based margin is that the learned classifier should have consistent parameters across samples, even if those samples do not reside in the same space, i.e., having different sets of observed features. Based on this assumption, the margin $\rho_i(\boldsymbol{\omega}, b)$ for the i-th $(1 \leq i \leq n)$ sample is defined as

$$\rho_i(\boldsymbol{\omega}, b) = \frac{y_i({\boldsymbol{\omega}^{(i)}}^\top \mathbf{x}_i + b)}{\|\boldsymbol{\omega}^{(i)}\|}, \tag{1}$$

where $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is a training data set, $\mathbf{x}_i$ is characterized by a subset of features from a full set $\mathcal{F}$, and $y_i \in \{+1, -1\}$ is the label of $\mathbf{x}_i$. $\boldsymbol{\omega}$ is the normal vector of SVMs on the full feature set $\mathcal{F}$, $\boldsymbol{\omega}^{(i)}$ is a vector obtained by taking the entries of $\boldsymbol{\omega}$ that are relevant to $\mathbf{x}_i$, namely those for which the sample $\mathbf{x}_i$ has observed features, and b is the bias term. $\mathbf{f}(\mathbf{x}_i) = {\boldsymbol{\omega}^{(i)}}^\top \mathbf{x}_i + b$ denotes the decision score of the learned classifier on $\mathbf{x}_i$.

Eq. (1) defines the margin for each sample in its own relevant space which the sample has observed features. As can be seen, it will reduce to a traditional margin as in SVMs when all samples have a full set $\mathcal{F}$. From this perspective, the work in [37] provides an elegant approach to handling samples with absent features. Though bearing this advantage, the maximization over the above sample-based margin makes the corresponding optimization problem more difficult to solve than the one in traditional SVMs. In [37], a two-step iterative optimization procedure is proposed to solve the problem. However, the optimization in [37] is non-convex and the global optimum cannot be guaranteed to obtain, which affects the performance of the learned classifier.

2.2 Multiple Kernel Learning

In MKL, each sample $\mathbf{x} = [{\mathbf{x}^{(1)}}^\top, \cdots, {\mathbf{x}^{(m)}}^\top]^\top$ can be treated as a concatenation of multiple base kernel mappings. Specifically, it takes the form of

$$\phi(\mathbf{x}) = [\phi_1^\top(\mathbf{x}^{(1)}), \cdots, \phi_m^\top(\mathbf{x}^{(m)})]^\top, \tag{2}$$

where $\{\mathbf{x}^{(p)}\}_{p=1}^m$ represents features from m views, $\{\phi_p(\mathbf{x}^{(p)})\}_{p=1}^m$ are m feature mappings corresponding to m pre-defined base kernels $\{\kappa_p(\cdot,\cdot)\}_{p=1}^m$, respectively. Based on this definition, the seminal work in MKL [1] proposes to optimize the following problem in Eq. (3),

$$\begin{aligned} &\min_{\boldsymbol{\omega},b,\boldsymbol{\xi}} \frac{1}{2}\left(\sum_{p=1}^m \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}\right)^2 + C\sum_{i=1}^n \xi_i, \\ &s.t.\ y_i\left(\sum_{p=1}^m \boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) \geq 1-\xi_i,\ \xi_i \geq 0,\ \forall i, \end{aligned} \tag{3}$$

where $\boldsymbol{\omega}_p$ is the normal vector corresponding to the p-th base kernel, b is the bias, $\boldsymbol{\xi}$ consists of the slack variables and $\mathcal{H}_p$ is a Hilbert space corresponding to base kernel κ_p.

Note that the first term of the objective function in Eq. (3) is not smooth since $\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}$ is not differentiable at $\boldsymbol{\omega}_p = 0$. This non-smoothness makes the problem hard to optimize. Fortunately, according to [3], [44], such a non-smooth term can be turned into a smooth objective one, as stated in Eq. (4),

$$\begin{aligned} \frac{1}{2}\left(\sum_{p=1}^m \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}\right)^2 = \Big\{ &\min_{\boldsymbol{\gamma}} \frac{1}{2}\sum_{p=1}^m \frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p} \\ &s.t.\ \sum_{p=1}^m \gamma_p = 1,\ \gamma_p \geq 0,\ \forall p\Big\}. \end{aligned} \tag{4}$$

Therefore, the MKL formulation in Eq. (3) can be equivalently rewritten as the one commonly used in the MKL literature [3]

$$\begin{aligned} &\min_{\boldsymbol{\omega},b,\boldsymbol{\xi},\boldsymbol{\gamma}} \frac{1}{2}\sum_{p=1}^m \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2 + C\sum_{i=1}^n \xi_i, \\ &s.t.\ y_i\left(\sum_{p=1}^m \sqrt{\gamma_p}\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) \geq 1-\xi_i,\ \xi_i \geq 0,\ \forall i, \\ &\quad \sum_{p=1}^m \gamma_p = 1,\ \gamma_p \geq 0,\ \forall p. \end{aligned} \tag{5}$$

In this paper, we adopt the formulation in Eq. (3) for the convenience of introducing the sample-based margin in a multiple-kernel-induced space.

To integrate multiple modalities information, current MKL algorithms assume that each sample $\mathbf{x}_i$ can be represented as $\phi(\mathbf{x}_i) = [\phi_1^\top(\mathbf{x}_i^{(1)}), \cdots, \phi_m^\top(\mathbf{x}_i^{(m)})]^\top$, where $\phi_p(\cdot)$ is applied to $\mathbf{x}_i^{(p)}$ $(1 \leq p \leq m)$ corresponding to one of its channels, as shown in Eq. (2).

A question naturally arises is that how to effectively perform MKL when some channels, i.e., part of $\phi_p(\mathbf{x}_i)$ in Eq. (2), are absent for a sample? In the following parts, we address this channel absence issue by defining the sample-based margin in multiple-kernel-induced space and develop two absent MKL (AMKL) algorithms based on this concept.

3 Absent MKL Algorithms

3.1 The Sample-based Margin in AMKL

We are given n training samples $\{(\phi(\mathbf{x}_i), y_i)\}_{i=1}^n$ and a missing matrix $\mathbf{s} \in \{0,1\}^{n\times m}$, where $\phi(\mathbf{x}_i)$ is defined as in Eq. (2) and the (i,p)-th entry of $\mathbf{s}$, $s(i,p) \in \{0,1\}$ $(1 \leq i \leq n, 1 \leq p \leq m)$, indicates whether the p-th channel of the i-th sample is absent or not. Specifically, $s(i,p) = 0$ implies absent and $s(i,p) = 1$ otherwise. Note that the missing matrices for both training and test sets are random and known in advance.

Similar to [37], we assume that a normal vector should be consistently shared across samples, no matter whether they have the same observed channels or not. Under this assumption, we define the margin for the i-th $(1 \leq i \leq n)$ sample as,

$$\rho_i(\boldsymbol{\omega}) = \frac{y_i\left(\sum_{p=1}^m s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right)}{\sum_{p=1}^m s(i,p)\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}, \tag{6}$$

where $\boldsymbol{\omega} = [\boldsymbol{\omega}_1^\top, \cdots, \boldsymbol{\omega}_m^\top]^\top$ and $\boldsymbol{\omega}_p$ $(1 \leq p \leq m)$ are the normal vectors corresponding to the whole channels and the p-th channel, respectively. $\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}$ is the norm in a Hilbert space induced by the p-th base kernel.

As can be seen, Eq. (6) defines the margin in a multiple-kernel-induced feature space for the samples with absent channels, i.e., some of $\{\phi_p(\mathbf{x}_i^{(p)})\}_{p=1}^m$ $(1 \leq i \leq n)$ are absent, as indicated by $s(i,p)$. At this point, we have extended the sample-based margin in [37], where some individual *features* of samples are missing, to MKL where some *channels* of samples are missing. As well known in the literature [1], the generalization performance of MKL is theoretically related to the margin between classes and a large margin is preferred. In AMKL, we propose to maximize the minimum of all sample-based margins so that the resultant classifier can separate the two classes as far as possible. This objective is fulfilled as in Eq. (7),

$$\max_{\boldsymbol{\omega}} \left(\min_{1\leq i\leq n} \frac{y_i\left(\sum_{p=1}^m s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right)}{\sum_{p=1}^m s(i,p)\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}\right). \tag{7}$$

Different from the traditional MKL optimization problem where its denominator is shared by all samples, the denominator in Eq. (7) varies across samples. This prevents Eq. (7) being equivalently rewritten as a readily handled constrained optimization problem, making it much more intractable than a traditional MKL problem.

In the following parts, we optimize this problem by proposing three algorithms, namely two iterative procedure solving its dual problem, and another easy-to-implement variant solving its primal problem which is reformulated to be convex. We term these algorithms iterative AMKL (I-AMKL), I-AMKL-λ and convex AMKL (C-AMKL), respectively.

3.2 The proposed I-AMKL

The difficulty of optimizing the problem in Eq. (7) lies in the variation of both the numerator and denominator with samples. To overcome this difficulty, we maintain the denominator as a variable shared across all samples by defining n auxiliary variables τ_i $(1 \leq i \leq n)$ as in Eq. (8),

$$\tau_i = \frac{\sum_{p=1}^m s(i,p)\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}{\sum_{p=1}^m \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}. \tag{8}$$

Based on $\boldsymbol{\tau} = [\tau_1, \cdots, \tau_n]^\top$, Eq. (7) can be reformulated as

$$\begin{aligned} &\max_{\boldsymbol{\omega}, b} \frac{1}{\sum_{p=1}^{m} \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}} \left(\min_{1 \le i \le n} \left(\frac{y_i}{\tau_i} \right) \left(\sum_{p=1}^{m} s(i,p) \boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b \right) \right) \\ &s.t.\ \tau_i = \frac{\sum_{p=1}^{m} s(i,p) \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}{\sum_{p=1}^{m} \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}. \end{aligned} \tag{9}$$

Due to the scale invariance of the fraction optimization, Eq. (9) is equivalently rewritten as a constrained optimization problem over $\boldsymbol{\omega}$ and $\boldsymbol{\tau}$, as in Eq. (10),

$$\begin{aligned} &\max_{\boldsymbol{\omega}, \boldsymbol{\tau}, b} \frac{1}{\sum_{p=1}^{m} \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}, \\ &s.t.\ \left(\frac{y_i}{\tau_i} \right) \left(\sum_{p=1}^{m} s(i,p) \boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b \right) \ge 1,\ \forall i, \\ &\tau_i = \frac{\sum_{p=1}^{m} s(i,p) \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}{\sum_{p=1}^{m} \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}. \end{aligned} \tag{10}$$

Turning the maximization optimization to a minimization one and adding slack variables $\boldsymbol{\xi}$ to handle the non-separable cases, Eq. (10) is conceptually rewritten as,

$$\begin{aligned} &\min_{\boldsymbol{\omega}, b, \boldsymbol{\xi}, \boldsymbol{\tau}, \boldsymbol{\gamma}} \frac{1}{2} \sum_{p=1}^{m} \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2 + C \sum_{i=1}^{n} \xi_i, \\ &s.t.\ \left(\frac{y_i}{\tau_i} \right) \left(\sum_{p=1}^{m} s(i,p) \sqrt{\gamma_p} \boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b \right) \ge 1 - \xi_i,\ \xi_i \ge 0, \forall i \\ &\sum_{p=1}^{m} \gamma_p = 1,\ 0 \le \gamma_p \le 1,\ \forall p \\ &\tau_i = \frac{\sum_{p=1}^{m} s(i,p) \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}{\sum_{p=1}^{m} \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}. \end{aligned} \tag{11}$$

where constraints on the base kernel weights $\boldsymbol{\gamma}$ are derived according to Eq. (4).

As observed, Eq. (11) has an additional variable $\boldsymbol{\tau}$ to be optimized when compared with the traditional MKL optimization problem, as presented in Eq. (5). To solve Eq. (11), we propose a two-step alternate algorithm in which an MKL-like optimization problem and the update of $\boldsymbol{\tau}$ are alternatively performed. Specifically, we optimize the parameters $\boldsymbol{\omega}$, b, $\boldsymbol{\xi}$ in the first step by solving an MKL-like problem with fixed $\boldsymbol{\tau}$. After that, $\boldsymbol{\tau}$ is updated with the optimized $\boldsymbol{\omega}$ in the last iteration via Eq. (8) in the second step. To optimize $\boldsymbol{\omega}$, b, $\boldsymbol{\xi}$, we present the Lagrange function of Eq. (11) with a fixed $\boldsymbol{\tau}$ as

$$\begin{aligned} &\mathcal{L}(\boldsymbol{\omega}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \sum_{p=1}^{m} \frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p} + C \sum_{i=1}^{n} \xi_i \\ &- \sum_{i=1}^{n} \alpha_i \Big(\big(\frac{y_i}{\tau_i} \big) \big(\sum_{p=1}^{m} s(i,p) \boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b \big) - 1 + \xi_i \Big) - \sum_{i=1}^{n} \beta_i \xi_i, \end{aligned} \tag{12}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are Lagrange multipliers. By taking the derivative of $\mathcal{L}(\boldsymbol{\omega}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\beta})$ with respect to $\boldsymbol{\omega}, b, \boldsymbol{\xi}$ and let it vanish, we obtain

$$\begin{cases} \boldsymbol{\omega}_p \ = \ \gamma_p \sum_{i=1}^{n} \alpha_i (\frac{y_i}{\tau_i}) s(i,p) \phi_p(\mathbf{x}_i^{(p)}),\ \forall p \\ \alpha_i + \beta_i = C,\ \forall i = 1, \cdots, n \\ \sum_{i=1}^{n} \alpha_i (\frac{y_i}{\tau_i}) = 0 \end{cases} \tag{13}$$

After combining Eq. (13) into Eq. (12), we derive the dual problem of Eq. (11) with a fixed $\boldsymbol{\tau}$ as follows

$$\begin{aligned} &\min_{\boldsymbol{\gamma}} \max_{\boldsymbol{\alpha}} \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2} \left(\boldsymbol{\alpha} \circ (\mathbf{y} \odot \boldsymbol{\tau}) \right)^\top \left(\sum_{p=1}^{m} \gamma_p \hat{\mathbf{K}}_p \right) \left(\boldsymbol{\alpha} \circ (\mathbf{y} \odot \boldsymbol{\tau}) \right) \\ &s.t.\ \boldsymbol{\alpha}^\top (\mathbf{y} \odot \boldsymbol{\tau}) = 0,\ 0 \le \alpha_i \le C,\ \forall i, \\ &\mathbf{1}^\top \boldsymbol{\gamma} = 1,\ 0 \le \gamma_p \le 1,\ \forall p, \end{aligned} \tag{14}$$

where $\mathbf{y} = [y_1, y_2, \cdots, y_n]^\top$ and $\boldsymbol{\tau} = [\tau_1, \tau_2, \cdots, \tau_n]^\top$. We use $\circ$ and $\odot$ to denote the element-wise product and division, respectively. $\mathbf{1}$ is a vector of all ones. $\hat{\mathbf{K}}_p$ denotes m base kernel matrices with $\hat{K}_p(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(p)}) = s(i,p)s(j,p)K_p(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(p)})\,(1 \le i, j \le n)$. It is worth pointing out that $\hat{\mathbf{K}}_p$ is still positive semi-definite (PSD) since it can be represented as $\left(\mathbf{s}(:,p)\mathbf{s}(:,p)^\top\right) \circ \mathbf{K}_p$ and the element-wise product of two PSD matrices are still PSD [45]. As a result, the inner maximization problem of Eq. (14) is kept convex as in traditional SVMs and the global optimum with respect to $\boldsymbol{\alpha}$ is guaranteed.

After obtaining $\boldsymbol{\alpha}$ by solving the Eq. (14), the norm of $\boldsymbol{\omega}_p\,(p = 1, \cdots, m)$ can be calculated via Eq. (15)

$$\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p} = \gamma_p \sqrt{\left(\boldsymbol{\alpha} \circ (\mathbf{y} \odot \boldsymbol{\tau}) \right)^\top \hat{\mathbf{K}}_p \left(\boldsymbol{\alpha} \circ (\mathbf{y} \odot \boldsymbol{\tau}) \right)}, \tag{15}$$

and then $\boldsymbol{\tau}$ is updated via Eq. (8).

This two-step procedure continues until the convergence criterion "$\max\{|\boldsymbol{\gamma}^{(t+1)} - \boldsymbol{\gamma}^{(t)}|\} \le \eta_0$" is satisfied, where $\boldsymbol{\gamma}^{(t+1)}$ and $\boldsymbol{\gamma}^{(t)}$ are the learned base kernel weights at the $(t+1)$-th and t-th iterations, respectively. The outline of the algorithm for solving I-AMKL is presented in Algorithm 1.

Algorithm 1 I-AMKL

1: **Input**: $\{\mathbf{K}_p\}_{p=1}^m$, $\mathbf{y}$, $\mathbf{s}$, C and η_0.
2: **Output**: $\boldsymbol{\alpha}$, b, $\boldsymbol{\gamma}$ and $\boldsymbol{\tau}$.
3: Initialize $\boldsymbol{\tau}^{(0)} = \mathbf{1}$ and $t = 0$.
4: **repeat**
5: Update $(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)})$ by solving Eq. (14) with $\boldsymbol{\tau}^{(t)}$.
6: Update $\boldsymbol{\tau}^{(t+1)}$ with $(\boldsymbol{\alpha}^{t+1}, \boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\tau}^{(t)})$ via Eq. (8) and (15).
7: $t = t + 1$.
8: **until** $\max\{|\boldsymbol{\gamma}^{(t+1)} - \boldsymbol{\gamma}^{(t)}|\} \le \eta_0$

As observed, Algorithm 1 iteratively performs a traditional MKL algorithm with the given $\boldsymbol{\tau}^{(t)}$ to obtain $\boldsymbol{\omega}_p^{(t+1)}$ and then updates $\boldsymbol{\tau}$ with $\boldsymbol{\omega}_p^{(t+1)}$. Consequently, the computational complexity of the proposed I-AMKL is $\mathcal{O}(N \cdot T_{\mathrm{MKL}})$, where N is the number of iteration and T_{MKL} is the computational complexity of running a traditional MKL at each iteration. According to our experimental results, it takes I-AMKL several iterations, usually less than ten ($N \le 10$), to reach the convergence criterion for all the data sets used in our experiments.

After obtaining $\boldsymbol{\alpha}$, b, $\boldsymbol{\gamma}$ and $\boldsymbol{\tau}$ by Algorithm 1, the decision score of the resultant classifier on a test sample

$\mathbf{x}_t$ is calculated as

$$\hat{y}(\mathbf{x}_t) = \sum_{p=1}^{m} \gamma_p \sum_{i=1}^{n} \alpha_i \left(\frac{y_i}{\tau_i}\right) s(i,p)t(p)\mathbf{K}_p(\mathbf{x}_i^{(p)}, \mathbf{x}_t^{(p)}) + b, \quad (16)$$

where $\mathbf{t} \in \{0,1\}^m$ is a pre-specified vector indicating the absence of the channels for $\mathbf{x}_t$.

3.3 The proposed I-AMKL-λ

Although the aforementioned I-AMKL in Algorithm 1 can be effectively solved, it is observed that the optimization w.r.t $\boldsymbol{\tau}$ is *unconstrained* during the whole course. Specifically, $\boldsymbol{\tau}$ is updated with the optimized $\boldsymbol{\omega}$ in the last iteration via Eq. (8). This makes I-AMKL only approximately solve the original optimization and its convergence hard to be theoretically guaranteed. In the following, we now propose another new algorithm that incorporates the equality constraints on τ_i into the objective to address this issue, leading to the following optimization,

$$\begin{aligned}
&\min_{\boldsymbol{\tau}} \min_{\boldsymbol{\omega},b,\boldsymbol{\xi},\boldsymbol{\gamma}} \ \frac{1}{2}\sum_{p=1}^{m} \|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2 + C\sum_{i=1}^{n}\xi_i + \frac{\lambda}{2n}\sum_{i=1}^{n}\left|\mathbf{t}_i^\top \mathbf{w}\right|^2, \\
&s.t.\ \left(\frac{y_i}{\tau_i}\right)\left(\sum_{p=1}^{m} s(i,p)\sqrt{\gamma_p}\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) \geq 1-\xi_i,\ \xi_i \geq 0, \forall i \\
&\quad \sum\nolimits_{p=1}^{m} \gamma_p = 1,\ 0 \leq \gamma_p \leq 1,\ \forall p
\end{aligned} \quad (17)$$

where the equality constraints are incorporated as penalty terms appended to the original objective, λ is a hyper-parameter to enforce these equality constraints, and $\mathbf{t}_i = [\tau_i - s(i,1), \cdots, \tau_i - s(i,m)]^\top, \mathbf{w} = [\|\boldsymbol{\omega}_1\|, \cdots, \|\boldsymbol{\omega}_m\|]^\top$.

The optimization in Eq. (17) is convex but non-smooth. We therefore turn to optimize its upper bound by observing that $|\mathbf{t}_i^\top \mathbf{w}| \leq \|\mathbf{t}_i\|^2\|\mathbf{w}\|^2, \forall i$. This leads to the optimization as follows,

$$\begin{aligned}
&\min_{\boldsymbol{\omega},\boldsymbol{\xi},b,\boldsymbol{\gamma},\boldsymbol{\tau}} \frac{1}{2}\Big(1+\frac{\lambda}{n}\sum\nolimits_{i=1}^{n}\|\mathbf{t}_i\|^2\Big)\sum\nolimits_{p=1}^{m}\|\boldsymbol{\omega}_p\|^2 + C\sum\nolimits_{i=1}^{n}\xi_i \\
&\quad s.t. \Big(\frac{y_i}{\tau_i}\Big)\Big(\sum_{p=1}^{m} s(i,p)\sqrt{\gamma_p}\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\Big) \geq 1-\xi_i,\ \xi_i \geq 0, \forall i \\
&\quad \sum\nolimits_{p=1}^{m}\gamma_p = 1, \gamma_p \geq 0,
\end{aligned} \quad (18)$$

which is equivalent to

$$\begin{aligned}
&\min_{\boldsymbol{\omega},\boldsymbol{\xi},b,\boldsymbol{\gamma},\boldsymbol{\tau}} \frac{1}{2}\left(1+\frac{\lambda}{n}\sum\nolimits_{i=1}^{n}\|\mathbf{t}_i\|^2\right)\sum\nolimits_{p=1}^{m}\frac{\|\boldsymbol{\omega}_p\|^2}{\gamma_p} + C\sum\nolimits_{i=1}^{n}\xi_i \\
&\quad s.t.\ \left(\frac{y_i}{\tau_i}\right)\left(\sum_{p=1}^{m} s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) \geq 1-\xi_i, \xi_i \geq 0, \forall i \\
&\quad \sum\nolimits_{p=1}^{m}\gamma_p = 1,\ \gamma_p \geq 0.
\end{aligned} \quad (19)$$

One can derive the Lagrangian dual or Fenchel-dual [46] of Eq. (19), and alternatively solve the $\boldsymbol{\omega}, \boldsymbol{\gamma}, \boldsymbol{\xi}$ and $\boldsymbol{\tau}$, as done in [3]. However, it would lead to a complicated fractional optimization on $\boldsymbol{\tau}$, which is difficult to solve. Instead, we take another approach to solve this obstacle. To do so, we first show what the form of $\boldsymbol{\omega}_p$ should take in the Eq. (19), as presented in the following Theorem 1.

Theorem 1. *The solution of $\boldsymbol{\omega}_p$ in Eq. (19) (and Eq. (28)) should take the form of*

$$\boldsymbol{\omega}_p = \sum\nolimits_{i=1}^{n} \alpha_i \kappa_p(\mathbf{x}_i^{(p)}, \cdot),\ \forall p \quad (20)$$

where $\kappa_p(\cdot,\cdot)$ is the p-th base kernel.

The proof of Theorem 1 are provided in the appendix due to space limit.

Based on Theorem 1, the optimization problem in Eq. (19) can be equivalently written as

$$\begin{aligned}
&\min_{\boldsymbol{\alpha},\boldsymbol{\xi},b,\boldsymbol{\gamma},\boldsymbol{\tau}} \frac{1}{2}\left(1+\frac{\lambda}{n}\sum\nolimits_{i=1}^{n}\|\mathbf{t}_i\|^2\right)\sum\nolimits_{p=1}^{m}\frac{\boldsymbol{\alpha}^\top\mathbf{K}_p\boldsymbol{\alpha}}{\gamma_p} + C\sum\nolimits_{i=1}^{n}\xi_i \\
&\quad s.t.\ y_i\left(\sum_{p=1}^{m} s(i,p)\boldsymbol{\alpha}^\top\mathbf{K}_p(:,\mathbf{x}_i^{(p)}) + b\right) \geq \tau_i(1-\xi_i), \xi_i \geq 0, \forall i \\
&\quad \sum\nolimits_{p=1}^{m}\gamma_p = 1,\ \gamma_p \geq 0,
\end{aligned} \quad (21)$$

We design a two-step alternative algorithm to solve the optimization problem in Eq. (21). In the first step, $\boldsymbol{\alpha}, \boldsymbol{\gamma}$ and $\boldsymbol{\xi}$ are optimized with given $\boldsymbol{\tau}$. With given $\boldsymbol{\tau}$, the optimization in Eq. (21) is jointly convex w.r.t $\boldsymbol{\alpha}, \boldsymbol{\gamma}$ and $\boldsymbol{\xi}$, which could be easily implemented via existing convex optimization packages such as CVX [43]. In the second step, $\boldsymbol{\tau}$ is optimized with given $\boldsymbol{\alpha}, \boldsymbol{\gamma}$ and $\boldsymbol{\xi}$. With fixed $\boldsymbol{\alpha}, \boldsymbol{\gamma}$ and $\boldsymbol{\xi}$, the optimization in Eq. (21) w.r.t $\boldsymbol{\tau}$ is as follows,

$$\begin{aligned}
&\min_{\boldsymbol{\tau}} \frac{1}{2}\sum\nolimits_{i=1}^{n}\tau_i^2 - \sum\nolimits_{i=1}^{n}\frac{\sum_{p=1}^{m} s(i,p)}{m}\tau_i \\
&s.t.\ \ v_i\tau_i \leq u_i,\ \forall i
\end{aligned} \quad (22)$$

where $v_i = 1-\xi_i$, $u_i = y_i(\sum_{p=1}^{m} s(i,p)\boldsymbol{\alpha}^\top\mathbf{K}_p(:,\mathbf{x}_i^{(p)}) + b)$.

Directly solving the optimization problem in Eq. (22) appears to be computationally intractable because it is a quadratic programming problem with n variables. Looking into this optimization problem, we can find that the constraints are separately defined on each τ_i and that the objective function is a sum over each τ_i. Therefore, we can equivalently rewrite the problem in Eq. (22) as n independent sub-problems, as stated in Eq. (23),

$$\min_{\tau_i} \ \frac{1}{2}\tau_i^2 - o_i\tau_i,\ s.t.\ v_i\tau_i \leq u_i, \quad (23)$$

where $o_i = \frac{\sum_{p=1}^{m} s(i,p)}{m}$. The optimization in Eq. (23) can be analytically obtained.

It is worth pointing out that the objective of I-AMKL-λ is guaranteed to be monotonically decreased when optimizing one variable with others fixed at each iteration. At the same time, the objective in Eq. (21) is lower-bounded by zero. As a result, the proposed I-AMKL-λ is guaranteed to converge. In addition, it is well recognized that the representer theorem in Theorem 1 usually gives a less theoretically elegant dual than the Lagrangian or Fenchel dual. However, we maintain to adopt the representer theorem dual in the paper since it: 1) well simplifies the resultant optimization problem; and 2) demonstrates comparable or even better classification performance in our experimental study.

3.4 The proposed C-AMKL

As can be seen, the procedure for I-AMKL and I-AMKL-λ alternately solve the optimization problems by defining an extra variable $\boldsymbol{\tau}$ and its global optimum is not guaranteed. This sometimes could deteriorate the classification performance of the learned classifier. In the following part, we show how to reformulate the optimization problem in Eq. (7) as a convex one such that it could be easily implemented via existing convex optimization packages such as CVX [43]. To this end, we first express the optimization problem in Eq. (7) as a constrained one as in Eq. (24),

$$\begin{aligned}\max_{\boldsymbol{\omega}} \min_{1\le i\le n} & \frac{1}{\sum_{p=1}^{m} s(i,p)\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}}, \\ s.t.\ & y_i\left(\sum_{p=1}^{m} s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) \ge 1,\ \forall i,\end{aligned} \tag{24}$$

which can be rewritten as Eq. (25),

$$\begin{aligned}\min_{\boldsymbol{\omega}} \max_{1\le i\le n} & \frac{1}{2}\left(\sum_{p=1}^{m} s(i,p)\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}\right)^2, \\ s.t.\ & y_i\left(\sum_{p=1}^{m} s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) \ge 1,\ \forall i.\end{aligned} \tag{25}$$

The optimization problem in Eq. (25) is non-smooth. With the similar trick in Eq. (4) and switching the minimization over $\boldsymbol{\gamma}$ and maximization over i in Eq. (25), we reformulate it as a smooth one, as stated in Eq. (26),

$$\begin{aligned}\min_{\boldsymbol{\omega},\boldsymbol{\gamma}} \max_{i} & \frac{1}{2}\sum_{p=1}^{m} s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p}, \\ s.t.\ & y_i\left(\sum_{p=1}^{m} s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) \ge 1,\ \forall i, \\ & \sum_{p=1}^{m}\gamma_p = 1,\ 0\le\gamma_p\le 1,\ \forall p.\end{aligned} \tag{26}$$

It is not difficult to check that the objective of Eq. (26) is an upper bound of Eq. (25). In the literature, one can minimize the objective by minimizing its upper bound. In addition, we prove that the optimums of these two optimization problems are equal. The detailed proof are provided in the appendix due to space limit.

Eq. (26) can be further rewritten as

$$\begin{aligned}\min_{\boldsymbol{\omega},\gamma} & u, \\ s.t.\ & y_i\left(\sum_{p=1}^{m} s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) \ge 1,\ \forall i, \\ & \frac{1}{2}\sum_{p=1}^{m} s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p} \le u,\ \forall i, \\ & \sum_{p=1}^{m}\gamma_p = 1,\ 0\le\gamma_p\le 1,\ \forall p.\end{aligned} \tag{27}$$

After adding slack variables $\boldsymbol{\xi}$ to deal with non-separable cases, we rewrite Eq. (27) as

$$\begin{aligned}\min_{\boldsymbol{\omega},\gamma,b,\boldsymbol{\xi},u} & u + C\sum_{i=1}^{n}\xi_i \\ s.t.\ & y_i\left(\sum_{p=1}^{m} s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) \ge 1-\xi_i,\ \xi_i\ge 0,\ \forall i, \\ & \frac{1}{2}\sum_{p=1}^{m} s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p} \le u,\ \forall i, \\ & \sum_{p=1}^{m}\gamma_p = 1,\ 0\le\gamma_p\le 1,\ \forall p.\end{aligned} \tag{28}$$

As can be seen in Eq. (28), the objective function as well as the first and third constraints are all linear in variables $[\boldsymbol{\omega}^\top, \boldsymbol{\gamma}^\top, b, \boldsymbol{\xi}^\top, u]^\top$. In addition, the second constraint, which is a quadratic cone, is also jointly convex in $[\boldsymbol{\omega}^\top, \boldsymbol{\gamma}^\top]^\top$ [43]. As a consequence, the optimization problem in Eq. (28) is convex.

Solving Eq. (28), however, is still difficult since the feature mapping functions $\{\phi_p(\mathbf{x})\}_{p=1}^m$ are usually not explicitly known. A commonly used trick to handle this problem is to derive its Lagrangian (or Fenchel) dual where $\{\phi_p(\mathbf{x})\}_{p=1}^m$ appears in the form of inner product in the kernel-induced feature space. Nevertheless, its dual problem, no matter Lagrangian or Fenchel dual, would lead to a complicated fractional optimization caused by the second constraint of Eq. (28). Instead, we apply the representer theorem [42] to overcome this obstacle. We firstly prove that $\boldsymbol{\omega}_p$ in Eq. (28) should take the form in Theorem 1. The proof are provided in the appendix due to space limit.

Based on Theorem 1 and the kernel reproducing property [47], we have

$$\mathbf{f}_p(\mathbf{x}^{(p)}) = \langle\boldsymbol{\omega}_p, \kappa_p(\mathbf{x}^{(p)},\cdot)\rangle_{\mathcal{H}_p} = \sum_{i=1}^{n}\alpha_i\kappa_p(\mathbf{x}_i^{(p)}, \mathbf{x}^{(p)}), \tag{29}$$

and

$$\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2 = \langle\boldsymbol{\omega}_p, \boldsymbol{\omega}_p\rangle_{\mathcal{H}_p} = \sum_{i,j=1}^{n}\alpha_i\alpha_j\kappa_p(\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(p)}). \tag{30}$$

With Eq. (29) and (30), the optimization problem in Eq. (28) can be equivalently rewritten as

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\gamma},b,\boldsymbol{\xi},u} & u + C\sum_{i=1}^{n}\xi_i \\ s.t.\ & y_i\left(\sum_{p=1}^{m} s(i,p)\boldsymbol{\alpha}^\top\mathbf{K}_p(:,\mathbf{x}_i^{(p)}) + b\right) \ge 1-\xi_i, \xi_i\ge 0, \forall i, \\ & \frac{1}{2}\sum_{p=1}^{m} s(i,p)\frac{\boldsymbol{\alpha}^\top\mathbf{K}_p\boldsymbol{\alpha}}{\gamma_p} \le u,\ \forall i, \\ & \sum_{p=1}^{m}\gamma_p = 1,\ 0\le\gamma_p\le 1,\ \forall p,\end{aligned} \tag{31}$$

where $\mathbf{K}_p(:,\mathbf{x}_i^{(p)}) = [\kappa_p(\mathbf{x}_1^{(p)},\mathbf{x}_i^{(p)}),\cdots,\kappa_p(\mathbf{x}_n^{(p)},\mathbf{x}_i^{(p)})]^\top$ and $\boldsymbol{\alpha} = [\alpha_1,\cdots,\alpha_n]^\top$.

The optimization problem in Eq. (31) is called convex AMKL (C-AMKL) in this paper, which has some intuitive explanation. In specific, the first constraint requires that the learnt classifier classifies samples by using the observed channels, i.e., those for which $s(i,p) = 1$. This maximally utilizes the available information while avoiding the potentially inaccurate imputation. The second constraint takes the maximum of the reciprocal of each sample-based margin and the objective function minimizes this maximum, which is equivalent to maximize the minimum of all sample-based margins. By this way, the learnt classifier is expected to have good generalization performance. Finally, we apply the convex optimization package CVX [43] to implement the optimization problem in Eq. (31). After obtaining $\boldsymbol{\alpha}$ and b, the decision score of the classifier on a new test sample $\mathbf{x}_t$ is calculated as

$$\hat{y}(\mathbf{x}_t) = \sum_{p=1}^{m} t(p)\boldsymbol{\alpha}^\top\mathbf{K}_p(:,\mathbf{x}_t^{(p)}) + b, \tag{32}$$

where $\mathbf{t} \in \{0,1\}^m$ is a vector indicating the absence of the channels for $\mathbf{x}_t$.

As can be seen from Eq. (31), its optimization problem is a linear programming with convex quadratic constraints. It is therefore a Quadratically Constrained Quadratic Program (QCQP), which can be efficiently solved by reformulated as a Second Order Cone Program problem (SOCP) [48]. Some advanced optimization techniques can be used to further improve the computational efficiency of the proposed C-AMKL, which will be a piece of our future work.

4 Generalization Analysis

We provide a generalization bound for the proposed absent multiple kernel learning algorithms. According to the Lagrange function in Eq. (12) and the objective function in Eq. (28), we conclude that all the proposed algorithms encourage the term $\frac{1}{2}\sum_{p=1}^{m}\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p}$ to be small. Let $\mathbf{w}_p \triangleq \gamma_p\boldsymbol{\omega}_p$ and $\frac{1}{2}\sum_{p=1}^{m}\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p} \leq \mu$ for simplicity. We therefore analyze the generalization property of the proposed learning algorithms by considering the complexity of the following hypothesis class

$$\mathcal{W}_{\mathbf{K}} = \Big\{\mathbf{w} : \mathbf{x} \mapsto \sum\nolimits_{p=1}^{m}\gamma_p\Big\langle \mathbf{w}_p, \phi_p(\mathbf{x}^{(p)})\Big\rangle \,|\, \sum\nolimits_{p=1}^{m}\gamma_p = 1, \gamma_p \geq 0, \frac{1}{2}\sum\nolimits_{p=1}^{m}\gamma_p\|\mathbf{w}_p\|_{\mathcal{H}_p}^2 \leq \mu\Big\}, \tag{33}$$

where $\mathbf{x} = [{\mathbf{x}^{(1)}}^\top, \cdots, {\mathbf{x}^{(m)}}^\top]^\top$ represents features from m views.

Assume that data are generated independently from a fixed but unknown probability distribution $\mathcal{D}$ over the joint space of the observed feature and label, i.e., $\mathbf{s} \circ \mathbf{X}$ and $\mathbf{Y}$. Note that if the p-th view of $\mathbf{X}$ is unobserved or absent, $s(p) = 0$; otherwise, $s(p) = 1$. Exploiting the indicator property of $s(p)$, we can define the expected error of a function $f_{\mathbf{w}}(\cdot) = \sum_{p=1}^{m}\gamma_p s(p)\langle \mathbf{w}_p, \phi_p(\cdot)\rangle$ by

$$R(\mathbf{w}) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[1_{\left(y\left(\sum_{p=1}^{m}\gamma_p s(p)\langle \mathbf{w}_p, \phi_p(\mathbf{x})\rangle\right)\leq 0\right)}\right], \tag{34}$$

where $1_{(\cdot)}$ is the indicator function representing the 0-1 loss function. Accordingly, we define the empirical margin error of the function $f_{\mathbf{w}}$ as

$$\hat{R}^{\rho}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} 1_{\left(y_i\left(\sum_{p=1}^{m}\gamma_p s(i,p)\langle \mathbf{w}_p, \phi_p(\mathbf{x}_i^{(p)})\rangle\right)\right)\leq\rho}, \tag{35}$$

where ρ is the margin. We have $R^{\rho}(\mathbf{w}) \to R(\mathbf{w})$ when $\rho \to 0$.

Let $\hat{\mathbf{w}}$ be the learned classifier. The following theorem 2 upper bounds the generalization error $R(\hat{\mathbf{w}}) - \hat{R}^{\rho}(\hat{\mathbf{w}})$.

Theorem 2. *Let $\{\phi_1, \cdots, \phi_m\}$ be the kernel mappings of a family of kernels containing m base kernels with different kernel widths. Assume that all the kernels are bounded, i.e., $\|\phi_p(\mathbf{x}^{(p)})\|_{\mathcal{H}_p}^2 \leq B$ for all $\mathbf{x}^{(p)} \in \mathcal{X}^{(p)}$ and $p \in \{1, \cdots, m\}$. Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ with $\mathbf{x}_i = [{\mathbf{x}_i^{(1)}}^\top, \cdots, {\mathbf{x}_i^{(m)}}^\top]^\top$ be an i.i.d. sample. For any $\rho > 0$ and $\delta > 0$, with probability at least $1 - 3\delta$, we have*

$$R(\hat{\mathbf{w}}) - \hat{R}^{\rho}(\hat{\mathbf{w}}) \leq \max_{1\leq p\leq m}\frac{2\sqrt{2\mu B}}{n\rho}\sqrt{\sum_{i=1}^{n}s(i,p)} + 3\sqrt{\frac{\log 1/\delta}{2n}} + \frac{16}{\rho}\sqrt{\frac{\mu B\ln((m+1)/\delta)}{n}}. \tag{36}$$

Note that $\sqrt{\sum_{i=1}^{n}s(i,p)} \leq \sqrt{n}$. Theorem 2 implies that the generalization bound will converge to zero when the training sample size n is sufficiently large. The convergence rate is of order $O(\sqrt{\frac{\ln m}{n}})$, which justifies that the proposed algorithms will generalize fast.

5 Experimental Results

5.1 Experimental Settings

In this section, we conduct experiments to compare the three variants of the proposed AMKL, i.e., I-AMKL, I-AMKL-λ and C-AMKL, with several commonly used imputation methods, including ZF-MKL, MF-MKL, EM-MKL and SVT-MKL. These methods are different in the imputation. Specifically, they are:

- **ZF-MKL** assigns zero to all absent channels.
- **MF-MKL** fills the absent channels with the value averaged on the samples for which the channels are observed.
- **EM-MKL** imputes the absent channels using the expectation maximization (EM) algorithm [40].
- **SVT-MKL** has been recently proposed to recover a large matrix from a small subset of its entries via nuclear norm minimization [49], [50]. We also include this imputation approach to see how it (singular value thresholding (SVT)) performs in the MKL setting. The SVT codes are downloaded from http://svt.stanford.edu and we follow their suggestions to set its parameters.

We discuss the differences between the proposed AMKL and the aforementioned imputation methods, i.e., ZF-MKL, MF-MKL, EM-MKL and SVT-MKL. The difference between AMKL and zero-filling is the way in calculating the margin. ZF-MKL first imputes the absent channels with zeros and maximizes the margin defined on the imputed samples. Differently, in AMKL the margin of each sample is calculated in its own relevant space and the minimum of these sample-based margins is maximized. Note that calculating the margin in a relevant space does not imply that the unobserved channels are set to zero. This difference can be clearly seen from Eq. (6). For ZF-MKL, all the term $s(i,p)$ in the denominator will simply be treated as one after the absent channels are imputed with zeros, and this will result in a denominator different from the one optimized by AMKL.

Besides differing in calculating the margin, MF-MKL fills the absent channels with the value averaged on the samples for which the channels are observed. Differently, EM-MKL imputes the absent channels with expectation

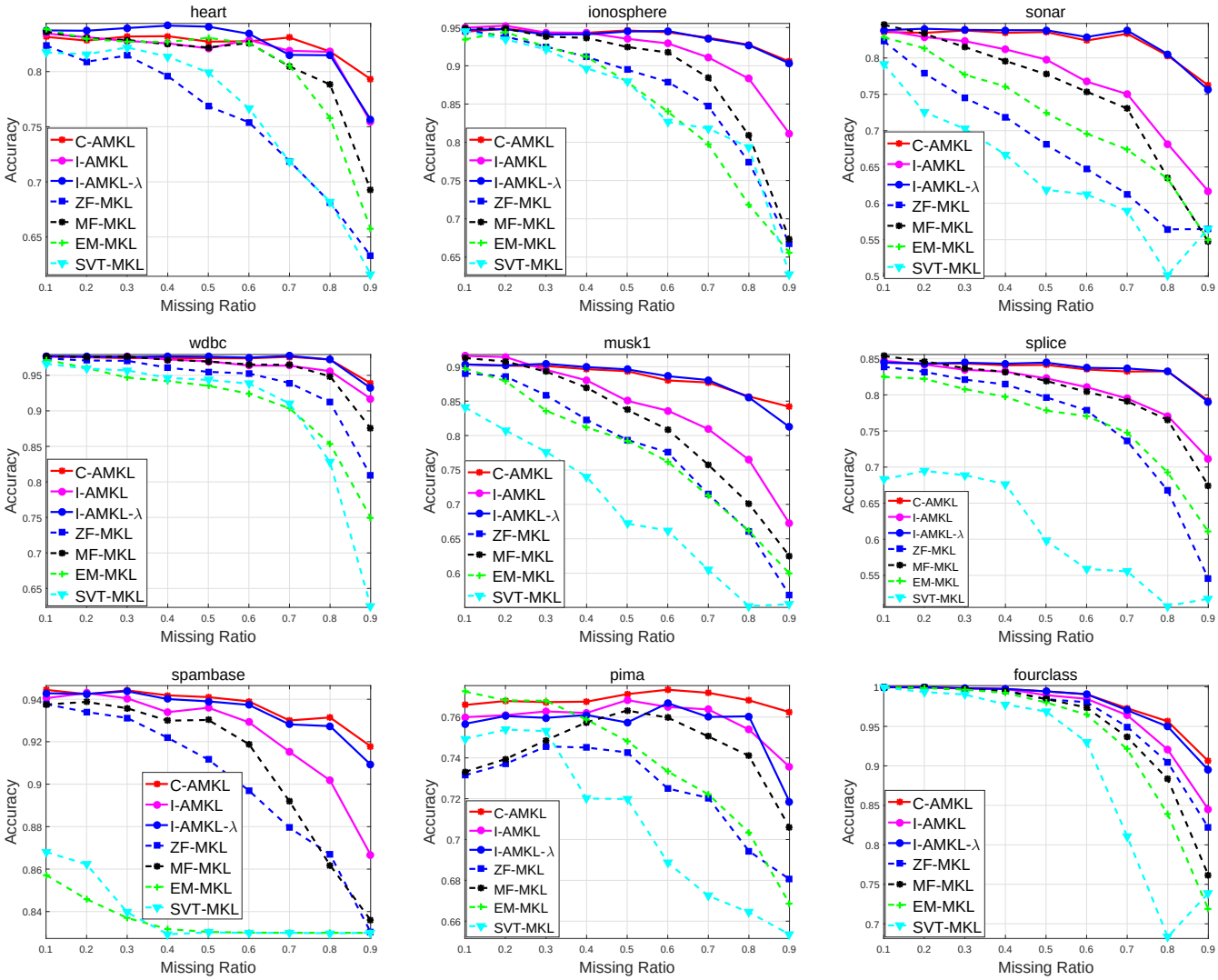

Fig. 1: Classification accuracy comparison of the above algorithms on nine UCI data sets.

maximization technique. Specifically, EM algorithm performs the following expectation (**E**) and maximization (**M**) steps iteratively at each iteration. In the E step, the mean and covariance matrix are estimated from the data matrix with missing channels filled with previous M step. Then, based on the estimated mean and the covariance, the M step fills the missing channels of each column with their conditional expectation values. These two steps are iteratively performed until convergence. The recently proposed SVT algorithm estimates the absent channels via minimizing the trace norm of data matrix. Note that all zero-filling, mean-filling, EM-filling and SVT-filling may become inaccurate when the channel absence is intensive, which would deteriorate the performance of the learnt classifier. The above drawback of ZF-MKL, MF-MKL, EM-MKL and SVT-MKL will be well verified in our experiments.

We then compare the pre-processing time of the above algorithms. For the proposed I-AMKL, I-AMKL-λ and C-AMKL, they directly classify the samples with absent channels without imputation. Therefore, the pre-processing time is zero. Comparably, the pre-processing time of ZF-MKL and MF-MKL is a little longer than that of I-AMKL, I-AMKL-λ and C-AMKL. However, this is not the case for EM-MKL and SVT-MKL. Their pre-processing time is significantly longer than the others. For example, it may take EM-MKL and SVT-MKL several days to finish the imputation even when the number of samples is less than three thousands. This is because calculating the conditional expectation values in **M** step is computationally intensive in EM-MKL. Similarly, the singular value decomposition (SVD) at each iteration makes SVT-MKL computationally inefficient. The I-AMKL, I-AMKL-λ, ZF-ML, MF-MKL, EM-MKL and SVT-MKL are implemented based on the SimpleMKL packages[1]. Meanwhile, we implement the C-AMKL via CVX [43].

We first evaluate the classification performance of the aforementioned algorithms on nine UCI data sets for binary classification[2], including *heart*, *ionosphere*, *sonar*,

1. http://asi.insa-rouen.fr/enseignants/~arakoto/code/mklindex.html
2. https://archive.ics.uci.edu/ml/datasets.html

TABLE 1: Classification accuracy comparison (mean±std) with pair-wise t-test on nine UCI data sets. Boldface means no statistical difference from the best one (p-val ≥ 0.01).

	Proposed			Imputation Methods			
	I-AMKL	I-AMKL-λ	C-AMKL	ZF-MKL	MF-MKL	EM-MKL	SVT-MKL
heart	$\mathbf{81.78 \pm 2.69}$ $\mathbf{0.03}$	$\mathbf{82.41 \pm 2.53}$ $\mathbf{0.86}$	$\mathbf{82.45 \pm 2.76}$ $\mathbf{1.00}$	75.55 ± 2.06 0.00	80.59 ± 2.30 0.00	79.99 ± 2.18 0.00	76.12 ± 2.40 0.00
ionosphere	91.76 ± 1.58 0.00	$\mathbf{93.70 \pm 1.35}$ $\mathbf{0.44}$	$\mathbf{93.76 \pm 1.33}$ $\mathbf{1.00}$	86.49 ± 1.68 0.00	88.67 ± 1.42 0.00	84.51 ± 1.35 0.00	84.92 ± 1.34 0.00
sonar	76.85 ± 3.19 0.00	$\mathbf{82.46 \pm 2.87}$ $\mathbf{1.00}$	$\mathbf{82.24 \pm 3.07}$ $\mathbf{0.11}$	68.18 ± 3.19 0.00	74.82 ± 2.75 0.00	71.74 ± 2.72 0.00	64.16 ± 2.10 0.00
wdbc	96.34 ± 0.84 0.00	$\mathbf{97.10 \pm 0.56}$ $\mathbf{1.00}$	$\mathbf{97.06 \pm 0.59}$ $\mathbf{0.49}$	93.82 ± 0.78 $\mathbf{1.00}$	95.82 ± 0.92 0.00	90.97 ± 1.02 0.00	89.72 ± 0.89 0.00
musk1	83.76 ± 2.02 0.00	$\mathbf{88.22 \pm 2.09}$ $\mathbf{0.09}$	$\mathbf{88.35 \pm 2.16}$ $\mathbf{1.00}$	77.46 ± 1.58 0.00	81.25 ± 1.82 0.00	77.25 ± 1.81 0.00	69.02 ± 1.39 0.00
splice	80.75 ± 1.22 0.00	$\mathbf{83.53 \pm 1.26}$ $\mathbf{1.00}$	$\mathbf{83.40 \pm 1.38}$ $\mathbf{0.03}$	75.90 ± 1.17 0.00	80.26 ± 1.17 0.00	76.13 ± 1.29 0.00	60.90 ± 1.13 0.00
spambase	92.30 ± 0.61 0.00	93.45 ± 0.73 0.00	$\mathbf{93.69 \pm 0.70}$ $\mathbf{1.00}$	90.11 ± 0.47 0.00	90.90 ± 0.50 0.00	83.58 ± 0.16 0.00	83.88 ± 0.27 0.00
pima	75.93 ± 0.96 0.00	75.57 ± 1.05 0.00	$\mathbf{76.84 \pm 1.09}$ $\mathbf{1.00}$	72.47 ± 1.26 0.00	74.43 ± 1.03 0.00	73.81 ± 0.88 0.00	70.84 ± 0.57 0.00
fourclass	96.65 ± 0.45 0.00	97.72 ± 0.36 0.00	$\mathbf{97.94 \pm 0.35}$ $\mathbf{1.00}$	95.91 ± 0.46 0.00	94.78 ± 0.51 0.00	93.45 ± 0.40 0.00	89.92 ± 0.46 0.00
avg.	86.24	88.24	**88.41**	81.77	84.61	81.27	76.61

spambase, *splice*, *wdbc*, *musk1*, *pima* and *fourclass*. For these data sets, we follow the approach in [5] to generate 20 Gaussian kernels as base kernels, whose width parameters are linearly equally sampled between $2^{-7}\sigma_0$ and $2^7\sigma_0$, with σ_0 the mean value of all pairwise distances. For each data set, 60% samples are randomly selected as the training set and the rest as test set. The detailed information about these data sets is summarized in the upper part of Table 2, where we treat each base kernel as a channel.

TABLE 2: UCI and MKL benchmark datasets used in our experiments.

Data	Number of			
	training	testing	classes	channels
heart	162	108	2	20
ionosphere	211	140	2	20
sonar	125	82	2	20
spambase	600	400	2	20
splice	600	400	2	20
wdbc	343	226	2	20
musk1	287	189	2	20
pima	461	307	2	20
fourclass	518	344	2	20
protein	427	267	27	12
psortPos	326	215	4	69
psortNeg	868	576	5	69
plant	566	374	4	69
flower17	816	544	17	21
caltech101	1020	2040	102	48

After that, we report the classification results of these algorithms on another six MKL benchmark data sets, including the *protein fold prediction* data set[3], *psortPos*, *psortNeg*, *plant* data sets[4], *flower17* data set[5] and the *Caltech101*[6]. All of them are multi-class classification tasks. The base kernel matrices of these data sets are pre-computed and publicly available from the above websites. For protein, psortPos, psortNeg, plant and flower17 data sets, 60% samples are randomly selected as the training set and the rest as the test set. For caltech101, ten samples are selected from each class as training set and the rest as test set. The detailed information about these data sets is presented in the lower part of Table 2.

We then show how to construct the absent matrix $\mathbf{s} \in \{0,1\}^{n\times m}$ on the training data, where n and m are the number of training samples and channels. In specific, we randomly generate a row of $\mathbf{s}$ and set its first $round(\varepsilon_0 * m)$[7] smallest values as zeros and the rest as ones, respectively. We repeat this process for n times to construct each row of $\mathbf{s}$. By this way, we construct an absent matrix $\mathbf{s}$ on training data. The absent matrix on test data is generated in the same way. The parameter ε_0, termed missing ratio in this paper, will affect the performance of the above algorithms. Intuitively, the larger the value of ε_0 is, the worse the performance that these algorithms can achieve. In order to show this point in depth, we compare the performance of these algorithms with respect to different ε_0. In specific, ε_0 on all data sets is set to be $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$, where $\varepsilon_0 = 0.1$ denotes the absence is the smallest while $\varepsilon_0 = 0.9$ means the absence is the most intensive.

The aggregated performance is used to evaluate the goodness of the above algorithms. Taking the aggregated

3. http://mkl.ucsd.edu/dataset/protein-fold-prediction/

4. http://raetschlab.org//suppl/protsubloc/

5. http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html/

6. http://files.is.tue.mpg.de/pgehler/projects/iccv09/

7. $round(\cdot)$ denotes a rounding function.

TABLE 3: Classification accuracy comparison (mean±std) with pair-wise *t*-test on the *protein fold prediction*. Boldface means no statistical difference from the best one (p-val ≥ 0.01).

Proposed			Imputation Methods				[37]
I-AMKL	I-AMKL-λ	C-AMKL	ZF-MKL	MF-MKL	EM-MKL	SVT-MKL	MMAF
52.88 ± 1.32	$\mathbf{56.96 \pm 0.79}$	55.65 ± 1.62	49.82 ± 1.44	52.92 ± 1.76	47.28 ± 1.57	34.00 ± 1.18	42.84 ± 1.40
0.00	**1.00**	0.00	0.00	0.00	0.00	0.00	0.00

TABLE 4: Classification accuracy comparison (mean±std) with pair-wise *t*-test on the *protein subcellular localization*. Boldface means no statistical difference from the best one (p-val ≥ 0.01).

	Proposed			Imputation Methods			
	I-AMKL	I-AMKL-λ	C-AMKL	ZF-MKL	MF-MKL	EM-MKL	SVT-MKL
psortPos	83.28 ± 2.04	83.92 ± 1.56	$\mathbf{85.32 \pm 2.09}$	75.37 ± 1.66	82.12 ± 1.57	76.77 ± 1.33	67.58 ± 1.35
	0.00	0.00	**1.00**	0.00	0.00	0.00	0.00
plant	84.23 ± 1.23	$\mathbf{87.52 \pm 1.70}$	86.55 ± 1.00	80.20 ± 0.80	83.31 ± 1.13	83.33 ± 0.70	66.39 ± 1.07
	0.00	**1.00**	0.00	0.00	0.00	0.00	0.00
psortNeg	82.40 ± 1.01	$\mathbf{83.21 \pm 1.46}$	$\mathbf{83.11 \pm 0.83}$	76.17 ± 0.80	82.29 ± 0.70	78.37 ± 0.97	60.96 ± 0.65
	0.00	**1.00**	**0.48**	0.00	0.00	0.00	0.00

classification accuracy for example, it is obtained by averaging the averaged accuracy achieved by an algorithm with different ε_0. We repeat this procedure 30 times on UCI data sets and ten times on MKL benchmark data sets to eliminate the randomness in generating the absent matrices, and report the averaged aggregated results and standard deviation. Furthermore, to conduct a rigorous comparison, the *paired student's t-test* is performed. The *p*-value of the pairwise *t*-test represents the probability that two sets of compared results come from distributions with an equal mean. A *p*-value of *0.01* is considered statistically significant. In our experiments, each base kernel is centralized and then scaled so that its diagonal elements are all ones. The regularization parameters C and λ for each algorithm are chosen from an appropriately large $[2^{-1}, 2^0, \cdots, 2^7]$ by 5-fold cross-validation on the training data. All experiments are conducted on a high performance cluster server, where each node has 2.3GHz CPU and 12GB memory.

5.2 Results on UCI datasets

In Figure 1, we report the classification accuracy of the above-mentioned algorithms on nine UCI data sets with the variation of missing ratios. From these figures, we have the following observations:

- The curves corresponding to the proposed C-AMKL algorithm are on the top in all sub-figures, indicating its best overall performance. At the same time, the proposed I-AMKL-λ algorithm generally demonstrates the overall second best performance. These results well demonstrate the effectiveness of the the proposed AMKL algorithms.
- The improvement of our proposed I-AMKL, I-AMKL-λ and C-AMKL algorithms is more significant with the increase of missing ratios. Taking the results on *ionosphere* data set for example, I-AMKL achieves higher accuracy than the best imputation algorithm (MF-MKL) by 1.07% when the missing ratio equals 0.5, and this improvement goes up to 13.76% when the missing ratio reaches 0.9. In addition, C-AMKL further improves these superior results to 2.17% and 23.19%, respectively.
- The variation of C-AMKL is much smaller with the increase of the missing ratio when compared with other algorithms. It implies that the performance of C-AMKL is relatively more stable, which is a desired characteristic for a good classifier.

The aggregated classification accuracy, standard deviation and the *p*-value of statistical test for each algorithm are reported in Table 1. As observed, I-AMKL and I-AMKL-λ usually achieve better performance than that of ZF-MKL, MF-MKL EM-MKL and SVT-MKL. Also, C-AMKL further significantly improves over I-AMKL and I-AMKL-λ, which is consistent with our observations in Figure 1.

We attribute the superiority of the proposed I-AMKL, I-AMKL-λ and C-AMKL algorithms to the sample-based margin maximization in each sample's own relevant space. In detail, the proposed AMKL algorithms take the channel absence of samples into consideration by maximizing the minimum of all sample-based margins. In contrast, ZF-MKL, MF-MKL, EM-MKL and SVT-MKL algorithms firstly fill the absent channels, and then maximize the margin on the imputed samples, as did in a standard MKL algorithm. As can be seen, such imputation approaches may not be reliable when the channel absence is relatively intensive, leading to poor performance in the sequential classification tasks. Also, though I-AMKL, I-AMKL-λ and C-AMKL maximize the minimum of sample-based margins, they differ in the optimization procedure. The optimization problem corresponding to I-AMKL and I-AMKL-λ is non-convex and are prone to being trapped into a local minimum. Differently, C-AMKL is free of this issue. This difference makes C-AMKL be able to achieve better performance, as validated by the experimental results.

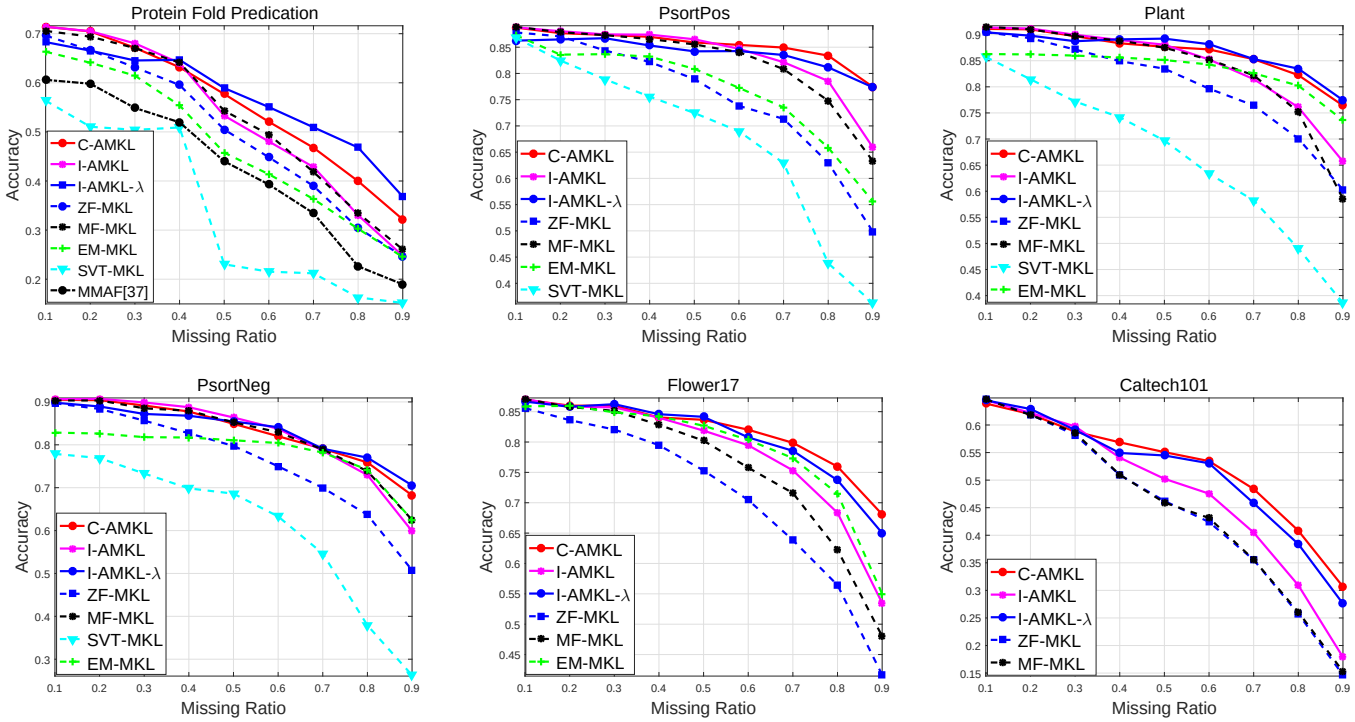

Fig. 2: Classification accuracy comparison of the above algorithms on the MKL benchmark data sets. (a) **Protein Fold Predication**. (b) **PsortPos**. (c) **Plant**. (d) **PsortNeg**. (e) **Flower17**. (f) **Caltech101**.

5.3 Results on the *protein fold prediction*

Besides the nine UCI data sets, we also compare the aforementioned algorithms on the protein fold prediction data set, which is a multi-source and multi-class data set based on a subset of the PDB-40D SCOP collection. It contains 12 different feature spaces, including composition, secondary, hydrophobicity, volume, polarity, polarizability, L1, L4, L14, L30, SWblosum62 and SWpam50. This data set has been widely adopted in the MKL community [22], [51]. For the protein fold prediction data set, the input features are available and the kernel matrices are generated as in [51], where the second order polynomial kernels are employed for feature sets one to ten and the linear kernel for the rest two feature sets.

This data set is a 27-class classification task and the one-against-rest strategy is used to solve the multi-class classification problem. As before, we vary the missing ratio from 0.1 to 0.9 with step size 0.1, and record the performance of these algorithms under different missing ratio, as plotted in Figure 2a. As can be observed, the proposed C-AMKL shows significant improvement after the missing ratio is larger than 0.4. With the missing ratio 0.5, it outperforms the second best one (MF-MKL) by 3.41%. Moreover, C-AMKL gains 6.10% improvement over the second best one (MF-MKL) when the missing ratio equals to 0.9.

The mean aggregated classification accuracy, standard deviation and the statistical test results are reported in Table 3. Again, we observe that: (1) The classification accuracy of I-AMKL is on par with the best imputation algorithms (MF-MKL), (2) I-AMKL-λ further improves I-AMKL, and demonstrates even better performance than C-AMKL when the missing ratio is larger than 0.4. In detail, I-AMKL-λ outperforms the best imputation algorithms (MF-MKL) by 4.04% in terms of the mean aggregated classification accuracy.

5.4 Results on the *protein subcellular localization*

In this subsection, We compare the performance of above-mentioned MKL algorithms on psortPos, psortNeg, plant data sets, which are from the protein subcellular localization and have been widely used in MKL community [52]–[55]. The base kernel matrices for these data sets have been pre-computed and can be publicly downloaded from the websites. Specifically, there are 69 base kernel matrices, including two kernels on phytogenetic trees, three kernels from BLAST E-values, and 64 sequence motif kernels. The class number of psortPos, psortNeg, plant data sets is four, five and four, respectively.

The accuracy achieved by the above algorithms on these three data sets with different missing ratios is plotted in Figures 2b, 2c and 2d, respectively. As can be seen, the proposed I-AMKL demonstrates overall comparable or better classification performance when compared with the imputation based algorithms on all data sets. In addition, the curves of C-AMKL and I-AMKL-λ are on the top in most of the cases, indicating the best performance. The improvement of both algorithms over the others becomes more significant with the increase of missing ratio. For example, C-AMKL outperforms the

TABLE 5: Classification accuracy comparison (mean±std) with pair-wise t-test on *Flower17*. Boldface means no statistical difference from the best one (p-val ≥ 0.01).

Proposed			Imputation Methods		
I-AMKL	I-AMKL-λ	C-AMKL	ZF-MKL	MF-MKL	EM-MKL
77.90 ± 1.04	80.61 ± 0.23	$\mathbf{81.40 \pm 1.17}$	70.94 ± 1.08	75.40 ± 1.17	78.65 ± 1.13
0.00	0.00	**1.00**	0.00	0.00	0.00

second best one by 8.97% (Figure 2b), 2.81% (Figure 2c) and 5.73% (Figure 2d) on psortPos, psortNeg, plant data sets when the missing ratio is 0.9. The mean aggregated accuracy, standard deviation and the p-value of statistical test of each algorithm are reported in Table 4. Again, the proposed C-AMKL and I-AMKL-λ demonstrate statistically significant improvement over the others.

5.5 Results on the *Flower17* dataset

We compare the above MKL algorithms on Oxford Flower17, which has been widely used as a MKL benchmark data set [56]. There are seven heterogeneous data channels available for this data set. For each data channel, we apply a Gaussian kernel with three different width parameters, i.e., $2^{-2}\sigma_0$, $2^0\sigma_0$ and $2^2\sigma_0$ to generate three kernel matrices, where σ_0 denotes the averaged pairwise distance. In this way, we obtain $21\,(7 \times 3)$ base kernels, and use them for all the MKL algorithms compared in our experiment.

Figure 2e plots the accuracy of the above algorithms with different missing ratios on the Flower17 data set. From this figure, we observe that the C-AMKL and I-AMKL-λ obtain superior performance over the others. Also, they demonstrate more improvement with the increase of missing ratio. When the missing ratio reaches 0.9, C-AMKL and I-AMKL-λ are superior to the second best one (MF-MKL) by nearly 13% and 10%, respectively. The corresponding mean aggregated accuracy, standard deviation and the statistical results are reported in Table 5. We can see that C-AMKL and I-AMKL-λ are consistently superior to other ones.

TABLE 6: Classification accuracy comparison on *Caltech101*.

Proposed			Imputation Methods	
I-AMKL	I-AMKL-λ	C-AMKL	ZF-MKL	MF-MKL
47.51	51.20	**52.22**	44.48	44.67

5.6 Results on the *Caltech101*

We conduct another experiment on the Caltech101 dataset to evaluate the performance of the proposed algorithms. This data set is a group of kernels derived from various visual features computed on the Caltech-101 object recognition task with 102 categories. It has 48 base kernels which are publicly available on websites.

The classification accuracy of the above algorithms is plotted in Figure 2f. As can be seen, the proposed I-AMKL is significantly better than the others after the missing ratio is larger than 0.3. I-AMKL-λ further significantly improves the classification accuracy of I-AMKL, and demonstrates comparable performance with C-AMKL. We also report the mean aggregated classification accuracy, standard deviation and the statistical test results in Table 6. Again, we observe that the proposed C-AMKL, I-AMKL-λ and I-AMKL achieve higher classification accuracy than the rest ones.

From the above experiments on nine UCI data sets and six MKL benchmark data sets, we conclude that: (1) The proposed AMKL effectively addresses the issue of channel absence in MKL. (2) The proposed C-AMKL, I-AMKL-λ and I-AMKL achieve superior performance ove ZF-MKL, MF-MKL, EM-MKL and SVT-MKL, especially in the presence of intensive absence.

6 Conclusion

While MKL algorithms have been used in various applications, they are not able to effectively handle the scenario where there are some absent channels in samples. To address this issue, this paper proposes to maximize the minimum of all sample-based margins in a multiple-kernel-induced feature space. After that, we propose three algorithms, namely I-AMKL, I-AMKL-λ and C-AMKL, to solve the optimization problem. Comprehensive experiments have demonstrated the effectiveness of our proposed algorithms, especially when the missing ratio is relatively high.

Many works are worth exploring in the future. For example, we plan to improve the computational efficiency of C-AMKL by solving it via more advanced optimization techniques such as the cutting plane method [19]. Moreover, considering the radius of minimum enclosing ball (MEB) [27], [28] may vary due to the channel absence of samples, it is worth trying to integrate the sample-based radius information to further improve the performance of our proposed AMKL.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (project no. 61773392 and 61672528).

References

[1] G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[2] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large Scale Multiple Kernel Learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, July 2006.

[3] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[4] J. Ye, S. Ji, and J. Chen, "Multi-class discriminant kernel learning via convex programming," *Journal of Machine Learning Research*, vol. 9, pp. 719–758, 2008.

[5] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research*, vol. 13, pp. 795–828, 2012.

[6] M. Gönen and S. Kaski, "Kernelized bayesian matrix factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 2047–2060, 2014.

[7] H. Xia, S. C. H. Hoi, R. Jin, and P. Zhao, "Online multiple kernel similarity learning for visual search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 536–549, 2014.

[8] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, "Multiple kernel learning for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1147–1160, 2011.

[9] N. A. Subrahmanya and Y. C. Shin, "Sparse multiple kernel learning for signal processing applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 788–798, 2010.

[10] S. Bucak, R. Jin, and A. Jain, "Multiple kernel learning for visual object recognition : A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1354–1369, 2014.

[11] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 465–479, 2012.

[12] Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu, "Simple and efficient multiple kernel learning by group lasso," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010, pp. 1175–1182.

[13] C. Cortes, M. Mohri, and A. Rostamizadeh, "Two-stage learning kernel algorithms," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010, pp. 239–246.

[14] M. Gönen, "Bayesian efficient multiple kernel learning," in *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

[15] A. Kembhavi, B. Siddiquie, R. Miezianko, S. McCloskey, and L. S. Davis, "Incremental multiple kernel learning for object recognition," in *ICCV*, 2009, pp. 638–645.

[16] C. H. Lampert, "Kernel methods in computer vision," *Foundations and Trends in Computer Graphics and Vision*, vol. 4, no. 3, pp. 193–285, 2009.

[17] S. Yu, L.-C. Tranchevent, X. Liu, W. Glänzel, J. A. K. Suykens, B. D. Moor, and Y. Moreau, "Optimized data fusion for kernel k-means clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 1031–1039, 2012.

[18] N. Subrahmanya and Y. Shin, "Sparse multiple kernel learning for signal processing applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 788–798, may 2010.

[19] Z. Xu, R. Jin, I. King, and M. R. Lyu, "An extended level method for efficient multiple kernel learning," in *Advances in Neural Information Processing Systems 21*, 2008, pp. 1825–1832.

[20] F. Orabona and J. Luo, "Ultra-fast optimization algorithm for sparse multi kernel learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011, pp. 249–256.

[21] X. Xu, W. Li, and D. Xu, "Distance metric learning using privileged information for face verification and person re-identification," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 26, no. 12, pp. 3150–3162, 2015.

[22] F. Yan, J. Kittler, K. Mikolajczyk, and M. A. Tahir, "Non-sparse multiple kernel fisher discriminant analysis," *Journal of Machine Learning Research*, vol. 13, pp. 607–642, 2012.

[23] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "l_p-norm multiple kernel learning," *Journal of Machine Learning Research*, vol. 12, pp. 953–997, 2011.

[24] C. Cortes, M. Mohri, and A. Rostamizadeh, "Learning non-linear combinations of kernels," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 396–404.

[25] A. Kumar, A. Niculescu-Mizil, K. Kavukcuoglu, and H. D. III, "A binary classification framework for two-stage multiple kernel learning," in *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

[26] H. Do, A. Kalousis, A. Woznica, and M. Hilario, "Margin and radius based multiple kernel learning," in *ECML/PKDD (1)*, 2009, pp. 330–343.

[27] K. Gai, G. Chen, and C. Zhang, "Learning kernels with radiuses of minimum enclosing balls," in *Advances in Neural Information Processing Systems 23*, 2010, pp. 649–657.

[28] X. Liu, L. Wang, J. Yin, E. Zhu, and J. Zhang, "An efficient approach to integrating radius information into multiple kernel learning," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 557–569, 2013.

[29] M. Gönen and A. A. Margolin, "Localized data fusion for kernel k-means clustering with application to cancer biology," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 1305–1313.

[30] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel k-means clustering with matrix-induced regularization," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2016.

[31] B. Gong, K. Grauman, and F. Sha, "Learning kernels for unsupervised domain adaptation with applications to visual object recognition," *International Journal of Computer Vision*, vol. 109, no. 1-2, pp. 3–27, 2014.

[32] M. Gönen and E. Alpaydin, "Localized multiple kernel learning," in *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML)*, 2008, pp. 352–359.

[33] X. Liu, L. Wang, J. Zhang, and J. Yin, "Sample-adaptive multiple kernel learning," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, pp. 1975–1981.

[34] X. Liu, L. Wang, J. Yin, Y. Dou, and J. Zhang, "Absent multiple kernel learning," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 2807–2813.

[35] L. Yuan, Y. Wang, P. M. Thompson, V. A. Narayan, and J. Ye, "Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data," *NeuroImage*, vol. 61, no. 3, pp. 622–632, 2012.

[36] ——, "Multi-source learning for joint analysis of incomplete multi-modality neuroimaging data," in *KDD*, 2012, pp. 1149–1157.

[37] G. Chechik, G. Heitz, G. Elidan, P. Abbeel, and D. Koller, "Max-margin classification of data with absent features," *Journal of Machine Learning Research*, vol. 9, pp. 1–21, 2008.

[38] B. M. Marlin, "Missing data problems in machine learning," Ph.D. dissertation, Department of Computer Science, University of Toronto, 2008.

[39] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann, "Kernel methods for missing variables," in *AISTATS05*, R. G. Cowell and Z. Ghahramani, Eds., 2005, pp. 325–332.

[40] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an em approach," in *Advances in Neural Information Processing Systems 6*, 1993, pp. 120–127.

[41] O. Dekel and O. Shamir, "Learning to classify with missing and corrupted features," in *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 2008, pp. 216–223.

[42] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *COLT/EuroCOLT*, 2001, pp. 416–426.

[43] I. CVX Research, "CVX: Matlab software for disciplined convex programming, version 2.0," http://cvxr.com/cvx, Aug. 2012.

[44] C. A. Micchelli and M. Pontil, "Learning the kernel function via regularization," *Journal of Machine Learning Research*, vol. 6, pp. 1099–1125, 2005.

[45] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[46] M. Kloft, U. Rückert, and P. L. Bartlett, "A unifying view of multiple kernel learning," in *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part II*, 2010, pp. 66–81.

[47] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[48] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *ICML*, 2004.

[49] J. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[50] E. J. Candès and T. Tao, "The power of convex relaxation: near-optimal matrix completion," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[51] T. Damoulas and M. A. Girolami, "Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection," *Bioinformatics*, vol. 24, no. 10, pp. 1264–1270, 2008.
[52] A. Zien and C. S. Ong, "Multiclass multiple kernel learning," in *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007, pp. 1191–1198.
[53] H. Yang, Z. Xu, J. Ye, I. King, and M. R. Lyu, "Efficient sparse generalized multiple kernel learning," *IEEE Trans. on neural networks*, vol. 22, 2011.
[54] C. Cortes, M. Mohri, and A. Rostamizadeh, "Multi-class classification with maximum margin multiple kernel," in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013, pp. 46–54.
[55] C. Cortes, M. Kloft, and M. Mohri, "Learning kernels using local rademacher complexity," in *Advances in Neural Information Processing Systems 26*, 2013, pp. 2760–2768.
[56] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1447–1454.

Xinwang Liu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Assistant Researcher of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 40+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-IP, IEEE T-NNLS, IEEE T-IFS, ICCV, AAAI, IJCAI, etc.

Lei Wang received his PhD degree from Nanyang Technological University, Singapore. He is now Associate Professor at School of Computing and Information Technology of University of Wollongong, Australia. His research interests include machine learning, pattern recognition, and computer vision. Dr. Wang has published 120+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IJCV, CVPR, ICCV and ECCV, etc. He was awarded the Early Career Researcher Award by Australian Academy of Science and Australian Research Council. He served as the General Co-Chair of DICTA 2014 and on the Technical Program Committees of 20+ international conferences and workshops. Lei Wang is senior member of IEEE.

Xinzhong Zhu is a professor at College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, PR China. He received his Ph.D. degree at XIDIAN University, China. His research interests include machine learning, computer vision, manufacturing informatization, robotics and system integration, and intelligent manufacturing. He is currently focusing on kernel learning and feature selection, multi-view clustering algorithms, real-time object detection (e.g., pedestrian detection, vehicle detection, general object detection, etc.) and deep learning, and their applications. He is a member of the ACM.

Miaomiao Li is in pursuit of her PhD degree at National University of Defense Technology, China. She is now Lecture of Changsha College, Changsha, China. Her current research interests include kernel learning and multi-view clustering. Miaomiao Li has published several peer-reviewed papers such as AAAI, IJCAI, Neurocomputing, etc. She serves on the Technical Program Committees of IJCAI 2017-2018.

En Zhu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor at School of Computer Science, NUDT, China. His main research interests are pattern recognition, image processing, machine vision and machine learning. Dr. Zhu has published 60+ peer-reviewed papers, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation.

Tongliang Liu received the PhD degree from the University of Technology Sydney. He is currently a Lecturer with the School of Information Technologies and the Faculty of Engineering and Information Technologies, and a core member in the UBTECH Sydney AI Centre, at The University of Sydney. His research interests include statistical learning theory, computer vision, and optimisation. He has authored and co-authored 40+ research papers including IEEE T-PAMI, T-NNLS, T-IP, ICML, CVPR, and KDD.

Li Liu received the Ph.D. degree in information and communication engineering from the National University of Defense Technology, China. She is now an Associate Professor with the College of System Engineering. Dr. Liu was a cochair of International Workshops at ACCV2014, CVPR2016, ICCV2017 and ECCV2018. She was a guest editor of the special issue on Compact and Efficient Feature Representation and Learning in Computer Vision for IEEE TPAMI. Her current research interests include texture analysis, image classification, object detection and scene understanding. Her papers have currently over 1400 citations in Google Scholar. She currently serves as Associate Editor of the Visual Computer Journal.

Yong Dou received his B.S., M.S., and Ph.D. degrees in computer science and technology from National University of Defense Technology (NUDT) in 1989, 1992 and 1995. He is now Professor at School of Computer Science, NUDT. His research interests include high performance computer architecture, high performance embedded microprocessor, reconfigurable computing, machine learning, and bioinformatics. He is a member of the IEEE and the ACM.

Jianping Yin received his PhD degree from National University of Defense Technology (NUDT), China. He is now the distinguished Professor at Dongguan University of Technology. His research interests include pattern recognition and machine learning. Dr. Yin has published 100+ peer-reviewed papers, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation' Supervisor and National Excellence Teacher. He served on the Technical Program Committees of 30+ international conferences and workshops.

Appendix of "Absent Multiple Kernel Learning Algorithms"

Xinwang Liu, Lei Wang, Xinzhong Zhu, Miaomiao Li, En Zhu, Tongliang Liu, Li Liu, Yong Dou and Jianping Yin

1 Summary

The appendix is organized as follows. Section 2 gives the detailed proof of Theorem 1 in the revised paper. After that, Section 3 conduct discussion on switching the order of minimization and maximization in Eq. (25) and Eq. (26) in the revised paper. We theoretically analyze the generalization error bound of the proposed AMKL algorithms in Section 4. Finally, we derive the partial duality of the optimization problem in Eq. (6) in the last response letter by following the suggestion of Reviewer_2 in Section 5.

2 Proof of Theorem 1

Theorem 1. *The solution of $\boldsymbol{\omega}_p$ in Eq. (19) (and Eq. (28) in the revised paper) should take the form of*

$$\boldsymbol{\omega}_p = \sum\nolimits_{i=1}^{n} \alpha_i \kappa_p(\mathbf{x}_i^{(p)}, \cdot), \forall p \tag{1}$$

where $\kappa_p(\cdot,\cdot)$ is the p-th base kernel.

Proof: We firstly prove that $\boldsymbol{\omega}_p$ in Eq. (19) takes the form of $\boldsymbol{\omega}_p = \sum_{i=1}^{n} \alpha_i \kappa_p(\mathbf{x}_i^{(p)}, \cdot)$, $\forall p$. The Lagrangian function of Eq. (19) is as follows,

$$\begin{aligned}\mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\xi}, b, \boldsymbol{\tau}; \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2) = &\frac{1}{2}\left(1 + \frac{\lambda}{n}\sum_{i=1}^{n}\|\mathbf{t}_i\|^2\right)\sum_{p=1}^{m}\frac{\|\boldsymbol{\omega}_p\|^2}{\gamma_p} + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_{i1}\left\{\left(\frac{y_i}{\tau_i}\right)\left(\sum_{p=1}^{m} s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) - 1 + \xi_i\right\} \\ &- \sum_{i=1}^{n}\alpha_{i2}\xi_i\end{aligned} \tag{2}$$

where $\mathbf{t}_i = [\tau_i - s(i,1), \cdots, \tau_i - s(i,m)]^\top$, and $\boldsymbol{\alpha}_1 = [\alpha_{11}, \cdots, \alpha_{1n}]^\top, \boldsymbol{\alpha}_2 = [\alpha_{21}, \cdots, \alpha_{2n}]^\top$ are Lagrange multipliers.

By taking the derivative of $\mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\xi}, b, \boldsymbol{\tau}; \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$ with respect to $\boldsymbol{\omega}_p$ and let it vanish, we obtain

$$\boldsymbol{\omega}_p = \frac{\gamma_p \sum_{i=1}^{n} \alpha_{1i}\left(\frac{y_i}{\tau_i}\right) s(i,p)\phi_p(\mathbf{x}_i^{(p)})}{1 + \frac{\lambda}{n}\sum_{i=1}^{n}\|\mathbf{t}_i\|^2}, \forall p \tag{3}$$

By redefining $\alpha_i = \frac{\gamma_p \alpha_{1i}\left(\frac{y_i}{\tau_i}\right) s(i,p)}{\left(1+\frac{\lambda}{n}\sum_{i=1}^{n}\|\mathbf{t}_i\|^2\right)}$, we have $\boldsymbol{\omega}_p = \sum_{i=1}^{n} \alpha_i \phi_p(\mathbf{x}_i^{(p)})$.

- *X. Liu, M. Li, E. Zhu and Y. Dou are with College of Computer, National University of Defense Technology, Changsha, China, 410073 (e-mail: xinwangliu@nudt.edu.cn, miaomiaolinudt@gmail.com, enzhu@nudt.edu.cn and yongdou@nudt.edu.cn).*
- *L. Wang is with School of Computing and Information Technology, University of Wollongong, NSW, Australia, 2522. (e-mail: leiw@uow.edu.au).*
- *X. Zhu is with College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua, China, 321004 (e-mail: zxz@zjnu.edu.cn).*
- *T. Liu is with the UBTECH Sydney Artificial Intelligence Centre and the School of Information Technologies in the Faculty of Engineering and Information Technologies at The University of Sydney, J12 Cleveland St, Darlington NSW 2008, Australia (e-mail: tongliang.liu@sydney.edu.au).*
- *L. Liu is with College of System Engineering, National University of Defense Technology, Changsha, China, 410073 and University of Oulu, Finland (e-mail: li.liu@oulu.fi).*
- *J. Yin is with Dongguan University of Technology, Guangdong, China (e-mail: jpyin@dgut.edu.cn).*

We then prove that $\boldsymbol{\omega}_p$ in Eq. (28) takes the form of $\boldsymbol{\omega}_p = \sum_{i=1}^n \alpha_i \kappa_p(\mathbf{x}_i^{(p)}, \cdot)$, $\forall p$. The Lagrange function of Eq. (28) is

$$\begin{aligned}\mathcal{L}(\boldsymbol{\omega}, b, \boldsymbol{\xi}, u; \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\beta}) =& u + C\sum_{i=1}^n \xi_i - \sum_{i=1}^n \mu_{1i}\left(y_i \sum_{p=1}^m s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + y_i b - 1 + \xi_i\right) - \sum_{i=1}^n \mu_{2i}\xi_i \\ &+ \sum_{i=1}^n \beta_i\left(\frac{1}{2}\sum_{p=1}^m s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p} - u\right)\end{aligned} \tag{4}$$

where $\boldsymbol{\mu}_1 = [\mu_{11}, \cdots, \mu_{1n}]^\top, \boldsymbol{\mu}_2 = [\mu_{21}, \cdots, \mu_{2n}]^\top, \boldsymbol{\beta} = [\beta_1, \cdots, \beta_n]^\top$ are Lagrange multipliers.

By taking the derivative of $\mathcal{L}(\boldsymbol{\omega}, b, \boldsymbol{\xi}, u; \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\beta}, \lambda_1, \boldsymbol{\lambda}_2)$ with respect to $\boldsymbol{\omega}_p$ and let it vanish, we obtain

$$\boldsymbol{\omega}_p = \frac{\gamma_p \sum_{i=1}^n \mu_{1i} y_i s(i,p)\phi_p(\mathbf{x}_i^{(p)})}{\sum_{i=1}^n \beta_i s(i,p)}, \forall p \tag{5}$$

By redefine $\alpha_i = \frac{\gamma_p \mu_{1i} y_i s(i,p)}{\sum_{i=1}^n \beta_i s(i,p)}$, we obtain $\boldsymbol{\omega}_p = \sum_{i=1}^n \alpha_i \phi_p(\mathbf{x}_i^{(p)})$. This completes the proof. □

3 Discussion on Switching the Order of Minimization and Maximization

In the following, we firstly show that the optimization $\min_{\boldsymbol{\omega}} \min_{\boldsymbol{\gamma}} \max_i$ is an upper bound of $\min_{\boldsymbol{\omega}} \max_i \min_{\boldsymbol{\gamma}}$. After that, we prove that the optimums of these two optimization problems are equal.

Let's define

$$\begin{aligned}(\textbf{OPT1}) \qquad & \min_{\boldsymbol{\omega}} \max_i \min_{\boldsymbol{\gamma}} \ \frac{1}{2}\sum\nolimits_{p=1}^m s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p}, \\ & s.t.\ y_i\left(\sum\nolimits_{p=1}^m s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) \geq 1, \forall i, \\ & \sum\nolimits_{p=1}^m \gamma_p = 1,\ 0 \leq \gamma_p \leq 1, \forall p,\end{aligned} \tag{6}$$

and

$$\begin{aligned}(\textbf{OPT2}) \qquad & \min_{\boldsymbol{\omega}} \min_{\boldsymbol{\gamma}} \max_i \ \frac{1}{2}\sum\nolimits_{p=1}^m s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p}, \\ & s.t.\ y_i\left(\sum\nolimits_{p=1}^m s(i,p)\boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i^{(p)}) + b\right) \geq 1, \forall i, \\ & \sum\nolimits_{p=1}^m \gamma_p = 1,\ 0 \leq \gamma_p \leq 1, \forall p,\end{aligned} \tag{7}$$

as in Eq. (6) and Eq. (7). Note that Eq. (**OPT1**) and (**OPT2**) are labelled as Eq. (19) and Eq. (20) (in the last submission), respectively.

The inequality $\min_{\boldsymbol{\gamma}} \frac{1}{2}\sum_{p=1}^m s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p} \leq \frac{1}{2}\sum_{p=1}^m s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p} \leq \max_i \frac{1}{2}\sum_{p=1}^m s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p}$ holds for all $1 \leq i \leq n$ and $\boldsymbol{\gamma}$. Therefore, we have $\max_i \min_{\boldsymbol{\gamma}} \frac{1}{2}\sum_{p=1}^m s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p} \leq \min_{\boldsymbol{\gamma}} \max_i \frac{1}{2}\sum_{p=1}^m s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p}$. This leads to $\min_{\boldsymbol{\omega}} \max_i \min_{\boldsymbol{\gamma}} \frac{1}{2}\sum_{p=1}^m s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p} \leq \min_{\boldsymbol{\omega}} \min_{\boldsymbol{\gamma}} \max_i \frac{1}{2}\sum_{p=1}^m s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p}$, indicating that the objective of (**OPT2**) is an upper bound of (**OPT1**). In the literature, one can minimize the objective by minimizing one of its upper bound.

We then prove that the optimums of (**OPT1**) and (**OPT2**) are equal. Let $(\boldsymbol{\gamma}^{(1)}, i^{(1)})$ and $(\boldsymbol{\gamma}^{(2)}, i^{(2)})$ be the optimal solutions of **OPT1** and **OPT2**, respectively. We have $\frac{1}{2}\sum_{p=1}^m s(i^{(2)},p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p^{(2)}} = \min_{\boldsymbol{\gamma}} \frac{1}{2}\sum_{p=1}^m s(i^{(2)},p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p} \leq \frac{1}{2}\sum_{p=1}^m s(i^{(2)},p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p^{(1)}} \leq \max_i \frac{1}{2}\sum_{p=1}^m s(i,p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p^{(1)}} = \frac{1}{2}\sum_{p=1}^m s(i^{(1)},p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p^{(1)}}$. This indicates that the objective of (**OPT2**) at $(\boldsymbol{\gamma}^{(2)}, i^{(2)})$ is no greater than that of (**OPT1**) at $(\boldsymbol{\gamma}^{(1)}, i^{(1)})$. On the other hand, the inequality $\frac{1}{2}\sum_{p=1}^m s(i^{(1)},p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p^{(1)}} \leq \frac{1}{2}\sum_{p=1}^m s(i^{(2)},p)\frac{\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2}{\gamma_p^{(2)}}$ holds according to the analysis in the last paragraph. Therefore, we conclude that the objectives of **OPT1** and **OPT2** are the same at their own optimums.

4 Proof of Generalization Error Bound

Let $\hat{\mathbf{w}}$ denote the learned classifier. The following theorem 2 upper bounds the generalization error $R(\hat{\mathbf{w}}) - \hat{R}^\rho(\hat{\mathbf{w}})$.

Theorem 2. *Let $\{\phi_1, \cdots, \phi_m\}$ be the kernel mappings of a family of kernels containing m base kernels with different kernel widths. Assume that all the kernels are bounded, i.e., $\|\phi_p(\mathbf{x}^{(p)})\|_{\mathcal{H}_p}^2 \leq B$ for all $\mathbf{x}^{(p)} \in \mathcal{X}^{(p)}$ and $p \in \{1, \cdots, m\}$. Let*

$\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with $\mathbf{x}_i = [{\mathbf{x}_i^{(1)}}^\top, \cdots, {\mathbf{x}_i^{(m)}}^\top]^\top$ be an i.i.d. sample. For any $\rho > 0$ and $\delta > 0$, with probability at least $1 - 3\delta$, we have

$$R(\hat{\mathbf{w}}) - \hat{R}^\rho(\hat{\mathbf{w}}) \leq \max_{1 \leq p \leq m} \frac{2\sqrt{2\mu B}}{n\rho} \sqrt{\sum_{i=1}^n s(i,p)} + \frac{16}{\rho}\sqrt{\frac{\mu B \ln((m+1)/\delta)}{n}} + 3\sqrt{\frac{\log 1/\delta}{2n}}. \tag{8}$$

Note that $\sqrt{\sum_{i=1}^n s(i,p)} \leq \sqrt{n}$. Theorem 2 implies that the generalization bound will converge to zero when the training sample size n is sufficiently large. The convergence rate is of order $O(\sqrt{\frac{\ln m}{n}})$, which justifies that the proposed algorithms will generalize fast.

Before proving Theorem 2, we first introduce the following function:

$$\ell^\rho(x) = \begin{cases} 0 & \text{if } x \geq \rho, \\ 1 - x/\rho & \text{if } 0 \leq x \leq \rho, \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

It can be easily verified that $R(\mathbf{w}) \leq \mathbb{E}\left[\ell^\rho(y\mathbf{w}(\mathbf{x}))\right]$ and that $\ell^\rho(x)$ is $1/\rho$-Lipschitz, i.e., $\forall x, x', |\ell^\rho(x) - \ell^\rho(x')| \leq (1/\rho)|x - x'|$.

Let

$$\hat{R}^\rho(\ell^\rho \cdot \mathbf{w}) = \frac{1}{n}\sum_{i=1}^n \ell^\rho\left(y_i\left\{\sum_{p=1}^m \gamma_p s(i,p)\langle \mathbf{w}_p, \phi_p(\mathbf{x}_i^{(p)})\rangle\right\}\right), \tag{10}$$

where $\ell^\rho \cdot \mathbf{w}$ stands for the composed function. It can also be easily verified that $\hat{R}^\rho(\mathbf{w}) \geq \hat{R}^\rho(\ell^\rho \cdot \mathbf{w})$.

By employing the framework of Rademacher complexity based generalization bound (Theorem 3.1 in [1]), we can obtain the following result.

Theorem 3. *Let $(\mathbf{s}_i \circ \mathbf{x}_i, y_i)_{i=1}^n$ be a sample independently generated from the distribution $\mathcal{D}$, where $\mathbf{s}_i = [s(i,1), \cdots, s(i,m)]^\top$ and $s(i,p) = 1$ indicates the p-th view of $\mathbf{x}_i$ is observed, and $s(i,p) = 0$ otherwise. For any $\delta > 0$, with probability at least $1 - \delta$, for all $\mathbf{w} \in \mathcal{W}_\mathbf{K}$, we have*

$$\mathbb{E}\left[\ell^\rho(y\mathbf{w}(\mathbf{x}))\right] - \hat{R}^\rho(\ell^\rho \cdot \mathbf{w}) \leq 2\mathfrak{R}_n(\ell^\rho \cdot \mathcal{W}_\mathbf{K}) + \sqrt{\frac{\log 1/\delta}{2n}}, \tag{11}$$

where

$$\mathfrak{R}_n(\ell^\rho \cdot \mathcal{W}_\mathbf{K}) = \mathbb{E}\left[\sup_{\mathbf{w} \in \mathcal{W}_\mathbf{K}} \frac{1}{n}\sum_{i=1}^n \sigma_i \ell^\rho\left(y_i\left\{\sum_{p=1}^m \gamma_p s(i,p)\left\langle \mathbf{w}_p, \phi_p(\mathbf{x}_i^{(p)})\right\rangle\right\}\right)\right], \tag{12}$$

$\mathfrak{R}_n(\ell^\rho \cdot \mathcal{W}_\mathbf{K})$ is the Rademacher complexity, and $\{\sigma_1, \cdots, \sigma_n\}$ are independent Rademacher variables uniformly distributed from $\{-1, 1\}$.

Rearranging the result in Theorem 3, with probability at least $1 - \delta$, we have

$$R(\hat{\mathbf{w}}) - \hat{R}^\rho(\hat{\mathbf{w}}) \leq 2\mathfrak{R}_n(\ell^\rho \cdot \mathcal{W}_\mathbf{K}) + \sqrt{\frac{\log 1/\delta}{2n}}. \tag{13}$$

Now, we are going to upper bound the Rademacher complexity $\mathfrak{R}_n(\ell^\rho \cdot \mathcal{W}_\mathbf{K})$. By further exploiting the McDiarmid's concentration inequality [2], the expected Rademacher complexity $\mathfrak{R}_n(\ell^\rho \cdot \mathcal{W}_\mathbf{K})$ can be upper bounded by using its empirical counterpart

$$\hat{\mathfrak{R}}_n(\ell^\rho \cdot \mathcal{W}_\mathbf{K}) = \mathbb{E}\left[\sup_{\mathbf{w} \in \mathcal{W}_\mathbf{K}} \frac{1}{n}\sum_{i=1}^n \sigma_i \ell^\rho\left(y_i\left\{\sum_{p=1}^m \gamma_p s(i,p)\langle \mathbf{w}_p, \phi_p(\mathbf{x}_i^{(p)})\rangle\right\}\right) \Big| (\mathbf{x}_i, y_i)_{i=1}^n\right]. \tag{14}$$

Theorem 4. *For any integer n and any $\delta > 0$, with probability at least $1 - \delta$, we have*

$$\mathfrak{R}_n(\ell^\rho \cdot \mathcal{W}_\mathbf{K}) \leq \hat{\mathfrak{R}}_n(\ell^\rho \cdot \mathcal{W}_\mathbf{K}) + \sqrt{\frac{\log 1/\delta}{2n}}. \tag{15}$$

Since ℓ^ρ is $1/\rho$-Lipschitz, using the Talagrand's contraction Lemma (page 78, Lemma 4.2 in [1]), we have

$$\hat{\mathfrak{R}}_n(\ell^\rho \cdot \mathcal{W}_\mathbf{K}) \leq 1/\rho \hat{\mathfrak{R}}_n(\mathcal{W}_\mathbf{K}). \tag{16}$$

Similar to the proof of [3], we will upper bound $\hat{\mathfrak{R}}_n(\mathcal{W}_{\mathbf{K}})$ by rewriting $\mathcal{W}_{\mathbf{K}}$ to be a convex hull of a set of function classes, i.e.,

$$\mathcal{W}_{\mathbf{K}} \subseteq \left\{ \sum_{p=1}^{m} \gamma_p \mathbf{w}_p \Big| \mathbf{w}_p \in \mathcal{W}_{\mathbf{K}_p}; \sum_{p=1}^{m} \gamma_p = 1, \gamma_p \geq 0 \right\} = \text{con}\left(\cup_{p=1}^{m} \mathcal{W}_{\mathbf{K}_p} \right), \tag{17}$$

where $\mathcal{W}_{\mathbf{K}_p} = \{\mathbf{w}_p : \mathbf{x} \mapsto s(p) \left\langle \mathbf{w}_p, \phi_p(\mathbf{x}^{(p)}) \right\rangle | \frac{1}{2}\|\mathbf{w}_p\|^2_{\mathcal{H}_p} \leq \mu\}$.

It can be easily verified that Rademacher complexity has the following properties: If $F \subseteq H$, then $\hat{\mathfrak{R}}_n(F) \leq \hat{\mathfrak{R}}_n(H)$; $\hat{\mathfrak{R}}_n(F) = \hat{\mathfrak{R}}_n(\text{con}F)$. Then, we have

$$\hat{\mathfrak{R}}_n(\mathcal{W}_{\mathbf{K}}) \leq \hat{\mathfrak{R}}_n(\cup_{p=1}^{m} \mathcal{W}_{\mathbf{K}_p}). \tag{18}$$

We employ the following theorem to upper bound $\hat{\mathfrak{R}}_n(\cup_{p=1}^{m} \mathcal{W}_{\mathbf{K}_p})$.

Theorem 5 ([3]). *If all the functions in $\cup_{p=1}^{m} \mathcal{W}_{\mathbf{K}_p}$ have the range of $[0, r]$, with probability at least $1 - \delta$, we have*

$$\hat{\mathfrak{R}}_n(\cup_{p=1}^{m} \mathcal{W}_{\mathbf{K}_p}) \leq \max_{1 \leq p \leq m} \hat{\mathfrak{R}}_n(\mathcal{W}_{\mathbf{K}_p}) + 8r\sqrt{\frac{\ln((m+1)/\delta)}{2n}}. \tag{19}$$

Using Cauchy-Schwarz inequality, we have that $\mathbf{w}_p(\mathbf{x}^{(p)}) = s(p) \left\langle \mathbf{w}_p, \phi_p(\mathbf{x}^{(p)}) \right\rangle \leq \|\mathbf{w}_p\|_{\mathcal{H}_p} \|\phi_p(\mathbf{x}^{(p)})\|_{\mathcal{H}_p} \leq \sqrt{2\mu B}$. Then, with probability at least $1 - \delta$, we have

$$\hat{\mathfrak{R}}_n(\cup_{p=1}^{m} \mathcal{W}_{\mathbf{K}_p}) \leq \max_{1 \leq p \leq m} \hat{\mathfrak{R}}_n(\mathcal{W}_{\mathbf{K}_p}) + 8\sqrt{\frac{\mu B \ln((m+1)/\delta)}{n}}, \tag{20}$$

where $\hat{\mathfrak{R}}_n(W_{\mathbf{K}_p}) = \mathbb{E}\big[\sup_{\mathbf{w}_p \in \mathcal{W}_{\mathbf{K}_p}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i y_i s(i,p) \langle \mathbf{w}_p, \phi_p(\mathbf{x}_i^{(p)}) \rangle \big| (\mathbf{x}_i, y_i)_{i=1}^n \big]$

By combining Eqs. (13), (15), (16), (18), and (20), with probability at least $1 - 3\delta$, we have

$$R(\hat{\mathbf{w}}) - \hat{R}^{\rho}(\hat{\mathbf{w}}) \leq \frac{2}{\rho} \max_{1 \leq p \leq m} \hat{\mathfrak{R}}_n(\mathcal{W}_{\mathbf{K}_p}) + \frac{16}{\rho}\sqrt{\frac{\mu B \ln((m+1)/\delta)}{n}} + 3\sqrt{\frac{\log 1/\delta}{2n}}. \tag{21}$$

Now we are going to upper bound $\hat{\mathfrak{R}}_n(\mathcal{W}_{\mathbf{K}_p})$. We have

$$\begin{aligned}
\hat{\mathfrak{R}}_n(\mathcal{W}_{\mathbf{K}_p}) &= \mathbb{E}\left[\sup_{\mathbf{w}_p \in \mathcal{W}_{\mathbf{K}_p}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i y_i s(i,p) \left\langle \mathbf{w}_p, \phi_p(\boldsymbol{x}_i^{(p)}) \right\rangle \big| (\mathbf{x}_i, y_i)_{i=1}^n \right] \\
&= \mathbb{E}\left[\sup_{\mathbf{w}_p \in \mathcal{W}_{\mathbf{K}_p}} \left\langle \mathbf{w}_p, \frac{1}{n} \sum_{i=1}^{n} \sigma_i s(i,p) \phi_p(\mathbf{x}_i^{(p)}) \right\rangle \big| (\mathbf{x}_i, y_i)_{i=1}^n \right] \\
&\leq \mathbb{E}\left[\sup_{\mathbf{w}_p \in \mathcal{W}_{\mathbf{K}_p}} \|\mathbf{w}_p\|_{\mathcal{H}_p} \left\| \frac{1}{n} \sum_{i=1}^{n} \sigma_i s(i,p) \phi_p(\mathbf{x}_i^{(p)}) \right\|_{\mathcal{H}_p} \big| (\mathbf{x}_i, y_i)_{i=1}^n \right] \\
&\leq \frac{\sqrt{2\mu}}{n} \mathbb{E}\left[\sqrt{\sum_{i=1}^{n} s(i,p) K_p(\mathbf{x}_i^{(p)}, \mathbf{x}_i^{(p)})} \big| (\mathbf{x}_i, y_i)_{i=1}^n \right] \\
&\leq \frac{\sqrt{2\mu}}{n} \sqrt{\mathbb{E}\left[\sum_{i=1}^{n} s(i,p) K_p(\mathbf{x}_i^{(p)}, \mathbf{x}_i^{(p)}) | (\mathbf{x}_i, x_i)_{i=1}^n \right]} \\
&\leq \frac{\sqrt{2\mu B}}{n} \sqrt{\sum_{i=1}^{n} s(i,p)},
\end{aligned} \tag{22}$$

where the first inequality holds because of Cauchy-Schwarz inequality; the second inequality holds because $\mathbb{E}[\sigma_i \sigma_j] = 0$ when $i \neq j$; the third inequality holds because of Jansen's inequality; and the last inequality holds because of Hölder's inequality. The proof of Theorem 2 ends by combining Eqs. (21) and (22). ■

5 Derivation of Partial Duality in Eq. (6)

We derive the partial duality of the following optimization problem (i.e., Eq. (6) in the last response letter),

$$\begin{aligned}\min_{\boldsymbol{\tau}} \min_{\boldsymbol{\omega},b,\boldsymbol{\xi},\boldsymbol{\gamma}} \ & \frac{1}{2}\left(1+\frac{\lambda}{n}\sum\nolimits_{i=1}^{n}\sum\nolimits_{q=1}^{m}(\tau_i - s(i,q))^2\right)\sum\nolimits_{p=1}^{m}\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2 + C\sum\nolimits_{i=1}^{n}\xi_i \\ s.t. \ & \left(\frac{y_i}{\tau_i}\right)\left(\sum_{p=1}^{m} s(i,p)\sqrt{\gamma_p}\boldsymbol{\omega}_p^\top\phi_p(\mathbf{x}_i^{(p)}) + b\right) \geq 1-\xi_i,\ \xi_i \geq 0, \forall i,\ \sum\nolimits_{p=1}^{m}\gamma_p = 1,\ 0\leq\gamma_p\leq 1,\ \forall p\end{aligned} \tag{23}$$

The Lagrange function of Eq. (23) with fixed $\boldsymbol{\tau}$ is

$$\begin{aligned}\sup_{\boldsymbol{\alpha},\boldsymbol{\beta}} \inf_{\boldsymbol{\gamma}} \inf_{\boldsymbol{\omega},b,\boldsymbol{\xi}} \ & \frac{1}{2}\left(1+\frac{\lambda}{n}\sum\nolimits_{i=1}^{n}\sum\nolimits_{q=1}^{m}(\tau_i - s(i,q))^2\right)\sum\nolimits_{p=1}^{m}\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2 + C\sum\nolimits_{i=1}^{n}\xi_i \\ & - \sum_{i=1}^{n}\alpha_i\left(\left(\frac{y_i}{\tau_i}\right)\left(\sum_{p=1}^{m} s(i,p)\sqrt{\gamma_p}\boldsymbol{\omega}_p^\top\phi_p(\mathbf{x}_i^{(p)}) + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{n}\beta_i\xi_i\end{aligned} \tag{24}$$

where $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_n]^\top$ and $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_n]^\top$ are Lagrange multipliers. Eq. (24) can be equivalently rearranged as

$$\begin{aligned}&\sup_{\boldsymbol{\alpha}} \inf_{\boldsymbol{\gamma}} \sum_{p=1}^{m} -\sup_{\boldsymbol{\omega}_p}\left\{\left\langle\boldsymbol{\omega}_p, \sqrt{\gamma_p}\sum_{i=1}^{n}\alpha_i\left(\frac{y_i}{\tau_i}\right)s(i,p)\phi_p(\mathbf{x}_i^{(p)})\right\rangle - \frac{1}{2}\left(1+\frac{\lambda}{n}\sum\nolimits_{i=1}^{n}\sum\nolimits_{q=1}^{m}(\tau_i - s(i,q))^2\right)\|\boldsymbol{\omega}_p\|_{\mathcal{H}_p}^2\right\} \\ &+\sup_{\boldsymbol{\alpha},\boldsymbol{\beta}} -C\sum\nolimits_{i=1}^{n}\sup_{\xi_i}\left\langle\xi_i, \frac{\alpha_i+\beta_i}{C} - 1\right\rangle + \sup_{\boldsymbol{\alpha}} -\sup_{b}\left\langle b, \sum\nolimits_{i=1}^{n}\alpha_i\left(\frac{y_i}{\tau_i}\right)\right\rangle + \sup_{\boldsymbol{\alpha}}\sum\nolimits_{i=1}^{n}\alpha_i.\end{aligned} \tag{25}$$

By following [4], we can express the above Lagrangian in terms of Fenchel-Legendre conjugate functions $h^*(\mathbf{x}) = \sup_{\mathbf{u}}\ \mathbf{x}^\top\mathbf{u} - h(\mathbf{u})$ as follows,

$$\begin{aligned}\sup_{\boldsymbol{\alpha}} \inf_{\boldsymbol{\gamma}} \ & -\frac{1}{2}\sum\nolimits_{p=1}^{m}\frac{\gamma_p}{\left(1+\frac{\lambda}{n}\sum_{i=1}^{n}\sum_{q=1}^{m}(\tau_i - s(i,q))^2\right)}\left\|\sum\nolimits_{i=1}^{n}\alpha_i\left(\frac{y_i}{\tau_i}\right)s(i,p)\phi_p(\mathbf{x}_i^{(p)})\right\|_{\mathcal{H}_p}^2 + \sum\nolimits_{i=1}^{n}\alpha_i \\ s.t. \ & \sum\nolimits_{i=1}^{n}\alpha_i\left(\frac{y_i}{\tau_i}\right) = 0,\ 0\leq\alpha_i\leq C,\ \forall i,\ \sum\nolimits_{p=1}^{m}\gamma_p = 1,\ 0\leq\gamma_p\leq 1,\ \forall p,\end{aligned} \tag{26}$$

which is equivalent to

$$\begin{aligned}\max_{\boldsymbol{\alpha}} \min_{\boldsymbol{\gamma}} \ & -\frac{1}{2}\sum_{p=1}^{m}\frac{\gamma_p}{\left(1+\frac{\lambda}{n}\sum_{i=1}^{n}\sum_{q=1}^{m}(\tau_i - s(i,q))^2\right)}\sum_{i,j=1}^{n}\alpha_i\alpha_j\left(\frac{y_i}{\tau_i}\right)\left(\frac{y_j}{\tau_j}\right)s(i,p)s(j,p)K_p(\mathbf{x}_i^{(p)},\mathbf{x}_j^{(p)}) + \sum_{i=1}^{n}\alpha_i \\ s.t. \ & \sum\nolimits_{i=1}^{n}\alpha_i\left(\frac{y_i}{\tau_i}\right) = 0,\ 0\leq\alpha_i\leq C,\ \forall i,\ \sum\nolimits_{p=1}^{m}\gamma_p = 1,\ 0\leq\gamma_p\leq 1,\ \forall p,\end{aligned} \tag{27}$$

Since Eq. (27) is convex in $\boldsymbol{\alpha}$ and concave in $\boldsymbol{\gamma}$, we can then switch the minimization of $\boldsymbol{\gamma}$ with the maximization of $\boldsymbol{\alpha}$, i.e.,

$$\begin{aligned}\min_{\boldsymbol{\gamma}} \max_{\boldsymbol{\alpha}} \ & -\frac{1}{2}\sum_{p=1}^{m}\frac{\gamma_p}{\left(1+\frac{\lambda}{n}\sum_{i=1}^{n}\sum_{q=1}^{m}(\tau_i - s(i,q))^2\right)}\sum_{i,j=1}^{n}\alpha_i\alpha_j\left(\frac{y_i}{\tau_i}\right)\left(\frac{y_j}{\tau_j}\right)s(i,p)s(j,p)K_p(\mathbf{x}_i^{(p)},\mathbf{x}_j^{(p)}) + \sum_{i=1}^{n}\alpha_i \\ s.t. \ & \sum\nolimits_{i=1}^{n}\alpha_i\left(\frac{y_i}{\tau_i}\right) = 0,\ 0\leq\alpha_i\leq C,\ \forall i,\ \sum\nolimits_{p=1}^{m}\gamma_p = 1,\ 0\leq\gamma_p\leq 1,\ \forall p,\end{aligned} \tag{28}$$

Eq. (28) is the partial dual of Eq. (23) with fixed $\boldsymbol{\tau}$.

References

[1] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2012.
[2] S. Boucheron, G. Lugosi, and P. Massart, *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.
[3] Z. Hussain and J. Shawe-Taylor, "A note on improved loss bounds for multiple kernel learning," *CoRR*, vol. abs/1106.6258, 2011. [Online]. Available: http://arxiv.org/abs/1106.6258
[4] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "l_p-norm multiple kernel learning," *Journal of Machine Learning Research*, vol. 12, pp. 953–997, 2011.