Two-Stage Fuzzy Multiple Kernel Learning Based on Hilbert-Schmidt Independence Criterion

Tinghua Wang, Jie Lu, *Fellow, IEEE*, and Guangquan Zhang

***Abstract*—Multiple kernel learning (MKL) is a principled approach to kernel combination and selection for a variety of learning tasks such as classification, clustering and dimensionality reduction. In this paper, we develop a novel fuzzy multiple kernel learning model based on the Hilbert-Schmidt independence criterion (HSIC) for classification, which we call HSIC-FMKL. In this model, we first propose an HSIC Lasso-based MKL formulation, which not only has a clear statistical interpretation that minimum redundant kernels with maximum dependence on output labels are found and combined, but also enables the global optimal solution to be computed efficiently by solving a Lasso optimization problem. Since the traditional support vector machine (SVM) is sensitive to outliers or noises in the data set, fuzzy support vector machine (FSVM) is used to select the prediction hypothesis once the optimal kernel has been obtained. The main advantage of FSVM is that we can associate a fuzzy membership with each data point such that these data points can have different effects on the training of the learning machine. We propose a new fuzzy membership function using a heuristic strategy based on the HSIC. The proposed HSIC-FMKL is a two-stage kernel learning approach and the HSIC is applied in both stages. We perform extensive experiments on real-world data sets from the UCI benchmark repository and the application domain of computational biology which validate the superiority of the proposed model in terms of prediction accuracy.**

***Index Terms*—Kernel method, classification, multiple kernel learning, fuzzy support vector machine.**

I. Introduction

KERNEL methods such as support vector machines (SVM) and kernel Fisher discriminant analysis (KFDA) have been successfully applied to a wide variety of machine learning problems [1]-[3]. These methods map data points from the input space to the feature space, i.e., higher dimensional reproducing kernel Hilbert space (RKHS), where even relatively simple algorithms, such as linear methods, can deliver very impressive performance. The mapping is determined implicitly by a kernel function (or simply a kernel), which computes the inner product of data points in the feature space. Despite the popularity of kernel methods, there is not yet a mechanism that is capable of guiding kernel leaning and selection. It is well known that selecting an appropriate kernel and thus an appropriate feature space is very important to the success of any kernel method [4]-[7]. To address this issue, researchers in recent years have engaged in active research on learning effective kernels automatically from data. One popular technique for kernel learning and selection is multiple kernel learning (MKL) [8], [9], which aims to learn a linear or nonlinear combination of a set of predefined kernels (base kernels) in order to identify a good target kernel for real applications. Compared with traditional kernel methods employing a fixed kernel, MKL demonstrates flexibility in automated kernel learning and also reflects the fact that typical learning problems often involve multiple, heterogeneous data sources. The idea of MKL can be generally applied to many kinds of kernel methods, such as the commonly used SVM and KFDA, leading to SVM-based MKL and discriminant MKL, respectively. Our work in this paper will mainly focus on the SVM-based MKL formulations.

There are two active research directions in SVM-based MKL [9], [10]. One is to improve the learning efficiency of MKL by exploiting different optimization techniques. Following the seminal work in which MKL was formulated and solved as a semidefinite programming (SDP) problem [11], many more efficient optimization algorithms [12]-[17] were proposed to handle medium or large scale problems. Among these algorithms, the alternating optimization technique, which alternates between the optimization of kernel weights and the optimization of SVM classifiers, is most widely used for practical applications. In each step of this technique, given the current solution of kernel weights, a classical SVM is solved with the combined kernel and a specific procedure to update the kernel weights is subsequently used.

The other research direction is to improve the prediction/classification accuracy of MKL by exploring possible combinations of base kernels. Researchers have developed various regularizers on kernel combinations, such as L_1-norm [13], L_p-norm ($p>1$) [18], [19], entropy-based [20],

This work was supported in part by the National Natural Science Foundation of China under grant 61562003, the Australian Research Council (ARC) under discovery grant DP170101632, and the China Scholarship Council under grant 201508360144.

T. Wang is with the School of Mathematics and Computer Science, Gannan Normal University, Ganzhou 341000, P. R. China, and also with the Centre for Artificial Intelligence, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, Broadway, NSW 2007, Australia (e-mail: wthgnnu@163.com).

J. Lu and G. Zhang are with the Centre for Artificial Intelligence, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, Broadway NSW 2007, Australia (e-mail: jie.lu@uts.edu.au; guangquan.zhang@uts.edu.au).

and mixed norms [21]. Of these, the L_1-norm of the kernel weights, also known as the simplex constraint, is probably the most popular choice because it results in sparse solutions and potentially eliminates irrelevant and noisy kernels. In addition to these linear combinations of base kernels, the possibility of nonlinear combinations of kernels has also been investigated [22]-[24]. Although the solution space is enlarged, nonlinear combinations usually result in a non-convex optimization problem, leading to even higher computational cost. Moreover, the solution of nonlinear combination is difficult to interpret. In summary, the first option makes MKL applicable to large scale learning tasks, while the second helps MKL to achieve superior classification performance.

Many extended MKL techniques have also been proposed to improve the regular MKL method, e.g., localized MKL [25], which achieves local assignments of kernel weights at the group level; sample-adaptive MKL [26], which switches off kernels at the data sample level; Bayesian MKL [27], which estimates the entire posterior distribution of model weights; and two-stage MKL [28]-[31], which first learns the optimal kernel weights according to certain criteria, then applies the learned optimal kernel to train a kernel classifier. Compared with one-stage MKL algorithms [14], [19], [25], which learn both the optimal weights for kernel combination and the SVM solution by solving a joint optimization problem, two-stage MKL algorithms generally achieve comparable or even better classification performance, while incurring much less computational cost. Moreover, two-stage MKL algorithms are more flexible, since the learned kernel in the first stage can be directly applied to train different kernel classifiers such as SVM or KFDA in the second stage. In other words, the two-stage MKL is classifer-independent while the one-stage MKL is classifier-dependent.

From a more general point of view, the two-stage MKL can be considered as a model selection problem: the kernel weights are considered as the hyperparameters of the classifier and can be tuned based on certain model selection criteria [32]. This approach has its root in the notions of kernel alignment [28], [33], [34] and kernel polarization [30, 35], which measure the level of similarity between a learning task and a kernel and have been extensively applied to kernel optimization and selection [6]. Geometrically, kernel learning with these criteria seeks a desired RKHS, in which data points belonging to different classes move apart while those associated with the same class come close. On the other hand, in recent years, the Hilbert-Schmidt independence criterion (HSIC) [36], a well-known kernel statistical dependence measure, has been successfully applied to not only the statistical independence test [37], [38], but also a variety of learning problems [39]. For example, clustering can be achieved by maximizing the statistical dependence between a discrete set of labels and the observations [40]. If labels are provided, feature selection can be viewed as searching for a feature subset in the observations which maximize the statistical dependence between features and labels [41]. Similarly, in subspace learning, a low dimensional embedding is sought which retains additional side information such as class labels and distance between neighboring observations [42]. The success is based on the fact that many existing learning tasks can be cast into problems of statistical dependence maximization (or minimization). However, there is as yet no such application in MKL.

Besides, traditional MKL approaches cannot deal with uncertainties (noise) in data sets. Actually, real-world data is never perfect and can often suffer from noise that may impact interpretations of the data, models created from the data and decisions made based on the data. Noise can reduce system performance in terms of classification accuracy, time in building a classifier and the size of the classifier [43]. In the binary classification scenario, for instance, a training data point may not exactly belong to either of the two classes when the outliers or noises exist in real-world applications. For example, a data point near the margin may belong to one class or just be a noise point. Thus, treating every training data point equally may cause over-fitting and lead to the degradation of the generalization performance of the final classifier [44].

To address these two issues, this paper presents a two-stage fuzzy MKL model based on the HSIC, called HSIC-FMKL. In the first stage, we propose an HSIC Lasso-based MKL formulation [1], which not only has a clear statistical interpretation that minimum redundant kernels with maximum dependence on output labels are found and combined, but also can efficiently compute the global optimal solution by solving a Lasso optimization problem. Since the traditional SVM is sensitive to outliers or noises in the data set, fuzzy support vector machine (FSVM) [46] is used in the second stage to select the prediction hypothesis. The main advantage of FSVM is that we can associate a fuzzy membership with each data point such that different data points have different effects on the training of the learning machine. Although well-determined fuzzy memberships can improve classification performance, there are no general guidelines for their construction [47]-[52]. In this study, a new fuzzy membership function calculation method is also proposed in which a heuristic function derived from the HSIC is used to calculate the dependence between a data point and its associated label. Lastly, the FSVM is trained to induce the final decision function to show classification results with the learned optimal kernel and fuzzy membership. The main contributions of the paper are outlined as follows:

- A novel MKL formulation based on the HSIC Lasso [53]-[55] was proposed, which not only has a clear statistical interpretation, but also enables the global optimal solution to be computed efficiently by solving a Lasso optimization problem. It should be pointed out that the HSIC Lasso [54] was originally proposed for high-dimensional feature selection, which needs to predefine the kernels (for example, Gaussian kernel for inputs and delta kernel for outputs) before feature selection, whereas our work employs the HSIC Lasso for MKL, aiming to learn an optimal composite kernel

[1] In statistics and machine learning, Lasso (least absolute shrinkage and selection operator, also referred to as LASSO) [45] is a regression analysis method that performs both variable selection and regularization to enhance the prediction accuracy and interpretability of the statistical model it produces.

to train a kernel classifier.

- To cope with the sensitivity to outliers or noises in the data set, FSVM is employed rather than SVM to induce the final decision function in the proposed MKL framework. A strategy of setting the reasonable fuzzy memberships for FSVM based on the HSIC is also presented.
- Comprehensive experiments on real-world data sets from the UCI benchmark repository and the application domain of computational biology verify the effectiveness of the proposed HSIC-FMKL model and show its possible real-world applications.

The rest of this paper is organized as follows. Brief introductions to HSIC, FSVM and MKL are given in Section II. The formulation, optimization and algorithm of our HSIC-FMKL are detailed in Section III. Section IV reports the experimental results, followed by the conclusion and further study in Section V.

II. Preliminaries

In this section, we briefly review some preliminaries on HSIC, FSVM and MKL.

A. Hilbert-Schmidt Independence Criterion

Let X and Y be two domains from which a set of samples $D=\{(x_i,y_i)\}_{i=1}^n$ is jointly drawn from some probability distribution P_{xy}. In the RKHS, HSIC [36] measures the independence (or dependence) between x and y by calculating the norm of the cross-covariance operator over the domain $\mathrm{X}\times\mathrm{Y}$. Formally, given $x,\ x'\in\mathrm{X}$, $y,\ y'\in\mathrm{Y}$, and two feature maps $\phi:\mathrm{X}\to\mathrm{F}$ and $\varphi:\mathrm{Y}\to\mathrm{G}$, where F and G are the RKHSs on X and Y, respectively, the corresponding reproducing kernels are defined as $k(x,x')=\langle\phi(x),\phi(x')\rangle$ and $l(y,y')=\langle\varphi(y),\varphi(y')\rangle$, respectively. The cross-covariance operator between ϕ and φ is a linear operator $C_{xy}:\mathrm{G}\to\mathrm{F}$, such that

$$C_{xy}=\mathrm{E}_{x,y}[[\phi(x)-\mathrm{E}_x[\phi(x)]]\otimes[\varphi(y)-\mathrm{E}_y[\varphi(y)]]] \tag{1}$$

where $\otimes$ is the tensor product, and the expectations E_x, E_y and $\mathrm{E}_{x,y}$ are taken according to marginal probability distributions P_x, P_y, and probability distribution P_{xy}, respectively. The HSIC is then defined by the square of the Hilbert-Schmidt norm of C_{xy}:

$$\begin{aligned}HSIC(\mathrm{F},\mathrm{G},\mathrm{P}_{xy}) &=\|C_{xy}\|_{\mathrm{HS}}^2=\mathrm{E}_{x,x',y,y'}[k(x,x')l(y,y')]\\ &-2\mathrm{E}_{x,y}[\mathrm{E}_{x'}[k(x,x')]\mathrm{E}_{y'}[l(y,y')]]\\ &+\mathrm{E}_{x,x'}[k(x,x')]\mathrm{E}_{y,y'}[l(y,y')]\end{aligned} \tag{2}$$

where $\mathrm{E}_{x,x',y,y'}$ is the expectation over both $(x,y)\sim\mathrm{P}_{xy}$ and an additional pair of variables $(x',y')\sim\mathrm{P}_{xy}$ drawn according to the same law independently. It is easy to find that if both feature maps are linear (i.e., $\phi(x)=x$ and $\varphi(y)=y$), HSIC is the same as the square of the Frobenius norm of the cross-covariance matrix. Given a sample set D, an empirical estimator of HSIC is as follows:

$$\begin{aligned}&HSIC(\mathrm{F},\mathrm{G},D)\\ &=\frac{1}{n^2}\mathrm{tr}(\mathbf{KL})-\frac{2}{n^3}\mathbf{e}^{\mathrm{T}}\mathbf{KLe}+\frac{1}{n^4}\mathbf{e}^{\mathrm{T}}\mathbf{Kee}^{\mathrm{T}}\mathbf{Le}\\ &=\frac{1}{n^2}\left[\mathrm{tr}(\mathbf{KL})-\frac{1}{n}\mathrm{tr}(\mathbf{KLee}^{\mathrm{T}})-\frac{1}{n}\mathrm{tr}(\mathbf{LKee}^{\mathrm{T}})+\frac{1}{n^2}\mathrm{tr}(\mathbf{Lee}^{\mathrm{T}}\mathbf{Kee}^{\mathrm{T}})\right]\\ &=\frac{1}{n^2}\left\{\mathrm{tr}\left[\mathbf{KL}\left(\mathbf{I}-\frac{1}{n}\mathbf{ee}^{\mathrm{T}}\right)\right]-\frac{1}{n}\mathrm{tr}\left[\mathbf{Kee}^{\mathrm{T}}\mathbf{L}\left(\mathbf{I}-\frac{1}{n}\mathbf{ee}^{\mathrm{T}}\right)\right]\right\}\\ &=\frac{1}{n^2}\mathrm{tr}\left[\mathbf{K}\left(\mathbf{I}-\frac{1}{n}\mathbf{ee}^{\mathrm{T}}\right)\mathbf{L}\left(\mathbf{I}-\frac{1}{n}\mathbf{ee}^{\mathrm{T}}\right)\right]\\ &=\frac{1}{n^2}\mathrm{tr}(\mathbf{KHLH})\triangleq HSIC(\mathbf{K},\mathbf{L})\end{aligned} \tag{3}$$

where $\mathrm{tr}(\cdot)$ is the trace operator, $\mathbf{e}=(1,\cdots,1)^{\mathrm{T}}\in\mathrm{R}^n$, $\mathbf{K},\ \mathbf{L}\in\mathrm{R}^{n\times n}$ (R denotes the set of real numbers) are the kernel matrices defined as $\mathbf{K}_{ij}=k(x_i,x_j)$ and $\mathbf{L}_{ij}=l(y_i,y_j)$, respectively, and $\mathbf{H}=\mathbf{I}-\mathbf{ee}^{\mathrm{T}}/n\in\mathrm{R}^{n\times n}$ is a centering matrix, where $\mathbf{I}\in\mathrm{R}^{n\times n}$ is the identity matrix. For convenience, we denote $HSIC(\mathrm{F},\mathrm{G},D)$ by $HSIC(\mathbf{K},\mathbf{L})$.

It is clear that the empirical estimator of HSIC is expressed completely in terms of kernels. For the so-called universal or characteristic kernels [56] such as Gaussian kernel and Laplace kernel, HSIC is equal to zero if and only if two random variables are statistically independent. Note that non-universal kernels or non-characteristic kernels can be also employed in HSIC, although they may not guarantee that all dependence can be detected [41].

B. Fuzzy Support Vector Machines

Support vector machine is a theoretically well motivated algorithm developed from statistical learning theory which has shown impressive performance in many fields [1], [2]. In spite of its success, it still suffers from the noise sensitivity problem originating from the assumption that each training point has equal importance or weight in the training process. In many real world problems, it is well known that there are cases where some training points are noises or outliers, and treating every training point equally may lead to over-fitting.

To relax the noise sensitivity problem, Lin and Wang [46] proposed an FSVM model based on the standard SVM for classification problems with noises or outliers. In the FSVM, a fuzzy membership associated with each training point is introduced such that different training points make different contributions to the final decision function. Formally, suppose we are given a set of labeled training samples $\{(\boldsymbol{x}_i,y_i,s_i)\}_{i=1}^n$ in a binary classification problem, where $\boldsymbol{x}_i\in\mathrm{X}\subset R^d$ is the input data, $y_i\in\{+1,-1\}$ is the corresponding class label, and

$s_i \in [0,1]$ is the fuzzy membership which represents the degree of $\boldsymbol{x}_i$ belonging to y_i. As with SVM, the goal of the FSVM is also to find an optimal hyperplane $\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x})+b=0$ that separates the training points into two classes with the maximal margin, where $\boldsymbol{w}$ is the normal vector of the hyperplane, b is a bias, and ϕ is a feature map which maps $\boldsymbol{x}_i$ to a high-dimensional feature space. This hyperplane can be obtained by solving the following optimization problem:

$$\begin{aligned} &\min \ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} s_i\xi_i \\ &\text{s.t.} \ \ y_i(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b) \geq 1-\xi_i \\ &\qquad \xi_i \geq 0, \ \ i=1,\cdots,n \end{aligned} \tag{4}$$

where $\boldsymbol{\xi}=(\xi_1,\cdots,\xi_n)^{\mathrm{T}}$ is the vector of slack variables and C is the regularization parameter used to impose a trade-off between the training error and generalization. Since the ξ_i is a measure of error for classifying the point $\boldsymbol{x}_i$ and the s_i is the attitude of the point $\boldsymbol{x}_i$ toward one class, the term $s_i\xi_i$ can be considered as a measure of error with different weights. Therefore, a smaller s_i can reduce the effect of the slack variable ξ_i in the objective function in (4), such that the corresponding point $\boldsymbol{x}_i$ is treated as less important. From another viewpoint, if we consider C as the cost assigned for a misclassification, then each data point is assigned with a different misclassification cost s_iC, such that more important data points have a higher cost, while less important data points have a lower cost.

To solve the FSVM optimization problem, suppose α_i is the Lagrange multiplier corresponding to the i^{th} inequality in (4), the dual problem of (4) is shown to be

$$\begin{aligned} &\max \ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_iy_j\alpha_i\alpha_j k(\boldsymbol{x}_i,\boldsymbol{x}_j) \\ &\text{s.t.} \ \sum_{i=1}^{n}\alpha_i y_i = 0, \\ &\qquad 0 \leq \alpha_i \leq s_iC, \ \ i=1,\cdots,n. \end{aligned} \tag{5}$$

The only difference between the original SVM dual-optimization problem and the FSVM dual-optimization problem is the upper bound of the values of α_i. After the solution has been obtained, the FSVM decision function is given by

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i k(\boldsymbol{x}_i,\boldsymbol{x}) + b\right) \tag{6}$$

where the samples $\boldsymbol{x}_i$ with $\alpha_i > 0$ are called support vectors.

This FSVM model has been successfully applied to reduce the effect of noises or outliers in a variety of applications with different methods of computing fuzzy memberships [46]-[52].

C. Multiple Kernel Learning

Instead of formulating an optimization criterion with a fixed kernel k, one can leave the kernel k as a combination of a set of predefined kernels, which results in the problem of MKL [8], [9]. MKL maps each sample to a multiple-kernel-induced feature space and a linear classifier is learned in this space. The feature mapping used in MKL takes the form of $\phi(\cdot)=[\phi_1^{\mathrm{T}}(\cdot),\cdots,\phi_M^{\mathrm{T}}(\cdot)]^{\mathrm{T}}$, which is induced by M predefined base kernels $\{k_m(\cdot,\cdot)\}_{m=1}^{M}$ with alternative kernel forms or kernel parameters. The linear combination of these kernels is given by

$$k = \sum_{m=1}^{M}\mu_m k_m \tag{7}$$

where μ_m is the corresponding combination coefficient. Let $\boldsymbol{\mu}=(\mu_1,\cdots,\mu_M)^{\mathrm{T}} \in \Delta$, where Δ is the domain of $\boldsymbol{\mu}$. By varying the constraint on $\boldsymbol{\mu}$, different MKL models can be obtained. For example, when $\boldsymbol{\mu} \in \Delta$ lies in a simplex, i.e.:

$$\Delta = \left\{\boldsymbol{\mu} : \|\boldsymbol{\mu}\|_1 = \sum_{m=1}^{M}\mu_m = 1, \mu_m \geq 0\right\} \tag{8}$$

we call the L_1-norm of kernel weights, and the resulting model is the L_1-MKL. Most MKL methods fall into this category. When

$$\Delta = \left\{\boldsymbol{\mu} : \|\boldsymbol{\mu}\|_p \leq 1, p>1, \mu_m \geq 0\right\} \tag{9}$$

we call the L_p-norm of kernel weights, and the resulting model is L_p-MKL [19]. A special case is the L_2-norm of kernel weights and the resulting model L_2-MKL [18].

Like SVM, the dual problem of MKL can be represented as

$$\begin{aligned} &\max \ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_iy_j\alpha_i\alpha_j \sum_{m=1}^{M}\mu_m k_m(\boldsymbol{x}_i,\boldsymbol{x}_j) \\ &\text{s.t.} \ \sum_{i=1}^{n}\alpha_i y_i = 0, \ \boldsymbol{\mu} \in \Delta, \\ &\qquad 0 \leq \alpha_i \leq C, \ \ i=1,\cdots,n. \end{aligned} \tag{10}$$

The goal of training MKL is to learn μ_m, α_i and b with the given M base kernels, and the final decision function is given by

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i \sum_{m=1}^{M}\mu_m k_m(\boldsymbol{x}_i,\boldsymbol{x}) + b\right). \tag{11}$$

III. The Two-stage Fuzzy MKL Method (HSIC-FMKL)

In this section, we present the two-stage fuzzy MKL method (HSIC-FMKL) for learning kernels in detail. The learned kernel is in the form of a linear combination of M base kernels $\{k_m(\cdot,\cdot)\}_{m=1}^{M}$ or kernel matrices $\{\mathbf{K}_m\}_{m=1}^{M}$. The corresponding combination coefficient μ_m is selected subject to the condition $\mu_m \geq 0$. In the first stage, the algorithm determines the combination coefficient μ_m, and in the second stage, an FSVM

is trained with the learned kernel.

A. The First Stage: MKL Using Lasso

Let $\bar{\mathbf{L}} = \mathbf{HLH}$ and $\bar{\mathbf{K}} = \mathbf{HKH}$, where $\mathbf{K}$, $\mathbf{L}$ and $\mathbf{H}$ are the kernel matrix for input data, kernel matrix for output labels, and centering matrix, respectively. We propose the use of HSIC Lasso [54] to estimate the combination coefficient $\boldsymbol{\mu}$:

$$\begin{aligned} &\min \ \frac{1}{2}\left\| \bar{\mathbf{L}} - \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m \right\|_{\mathrm{F}}^2 + \lambda \|\boldsymbol{\mu}\|_1 \\ &\text{s.t. } \mu_1, \cdots, \mu_M \geq 0 \end{aligned} \tag{12}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm and $\lambda > 0$ is the regularization parameter. In (12), the first term means that we are aligning the centered output kernel matrix $\bar{\mathbf{L}}$ by a linear combination of the centered input base kernel matrices $\{\bar{\mathbf{K}}_m\}_{m=1}^{M}$, and the second term means that the combination coefficients for irrelevant base kernels become zero since the L_1-regularizer tends to produce a sparse solution. After estimating $\boldsymbol{\mu}$, we normalize each element of $\boldsymbol{\mu}$ as $\mu_m \rightarrow \mu_m \big/ \sum_{m=1}^{M} \mu_m$.

Noting $< \bar{\mathbf{K}}, \bar{\mathbf{L}} >_{\mathrm{F}} = < \bar{\mathbf{K}}, \mathbf{L} >_{\mathrm{F}} = < \mathbf{K}, \bar{\mathbf{L}} >_{\mathrm{F}} = n^2 HSIC(\mathbf{K}, \mathbf{L})$, we can rewrite the first term of (12) as

$$\begin{aligned} &\frac{1}{2}\left\| \bar{\mathbf{L}} - \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m \right\|_{\mathrm{F}}^2 \\ &= \frac{1}{2}\left\langle \bar{\mathbf{L}} - \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m, \bar{\mathbf{L}} - \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m \right\rangle_{\mathrm{F}} \\ &= \frac{1}{2}\left\langle \bar{\mathbf{L}}, \bar{\mathbf{L}} \right\rangle_{\mathrm{F}} - \left\langle \bar{\mathbf{L}}, \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m \right\rangle_{\mathrm{F}} + \frac{1}{2}\left\langle \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m, \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m \right\rangle_{\mathrm{F}} \\ &= \frac{1}{2}\left\langle \bar{\mathbf{L}}, \bar{\mathbf{L}} \right\rangle_{\mathrm{F}} - \sum_{m=1}^{M} \mu_m \left\langle \bar{\mathbf{L}}, \bar{\mathbf{K}}_m \right\rangle_{\mathrm{F}} + \frac{1}{2}\sum_{m=1}^{M}\sum_{o=1}^{M} \mu_m \mu_o \left\langle \bar{\mathbf{K}}_m, \bar{\mathbf{K}}_o \right\rangle_{\mathrm{F}} \\ &= \frac{n^2}{2} HSIC(\mathbf{L}, \mathbf{L}) - n^2 \sum_{m=1}^{M} \mu_m HSIC(\mathbf{L}, \mathbf{K}_m) \\ &\quad + \frac{n^2}{2}\sum_{m=1}^{M}\sum_{o=1}^{M} \mu_m \mu_o HSIC(\mathbf{K}_m, \mathbf{K}_o). \end{aligned} \tag{13}$$

In (13), the n^2 and $HSIC(\mathbf{L}, \mathbf{L})$ are constant and can be ignored. We have a clear statistical interpretation of MKL using HSIC Lasso. First, if the m^{th} kernel matrix $\mathbf{K}_m$ has high dependence on the output matrix $\mathbf{L}$, $HSIC(\mathbf{L}, \mathbf{K}_m)$ takes a large value and thus μ_m should also be large so that (13) is minimized. On the other hand, if $\mathbf{K}_m$ and $\mathbf{L}$ are independent, $HSIC(\mathbf{L}, \mathbf{K}_m)$ is close to zero and thus μ_m tends to be removed by the L_1-regularizer. This means that relevant kernels that have strong dependence on output $\mathbf{L}$ tend to be selected by the HSIC Lasso. Second, if $\mathbf{K}_m$ and $\mathbf{K}_o$ are strongly dependent, which means one of them is a redundant kernel, $HSIC(\mathbf{K}_m, \mathbf{K}_o)$ takes a large value and thus either μ_m or μ_o tends to be zero. This means that redundant kernels tend to be removed by the HSIC Lasso. In summary, HSIC Lasso tends to find non-redundant kernels with strong dependence on output $\mathbf{L}$, which is a preferable property in kernel learning.

In practice, many Lasso optimization techniques can be applied to solve the HSIC Lasso problem shown in (12), such as dual augmented Lagrangian (DAL) [57], [58], which has been successfully employed for high-dimensional feature selection [53], [54].

The statistical interpretation of MKL using HSIC Lasso is very similar to feature selection using the minimum redundancy maximum relevance (MRMR) criterion [59], [60], which aims to find non-redundant features with strong dependence on output labels. The idea of MRMR was recently applied to base kernel selection before the optimization of MKL solvers [61]. In [61], the kernel alignment [33] rather than the mutual information in [59], [60] was employed as the dependence measure. Mathematically, let $\mathbf{KC}$ be a set of candidate kernels, $\mathbf{KS}_m$ consisting of m kernels be a selected subset of $\mathbf{KC}$. The m^{th} kernel is selected according to

$$\max_{\mathbf{K}_i \in \mathbf{KC}\text{-}\mathbf{KS}_{m-1}} \left[KA(\mathbf{K}_i, \mathbf{L}) - \frac{1}{m-1} \sum_{\mathbf{K}_j \in \mathbf{KS}_{m-1}} KA(\mathbf{K}_i, \mathbf{K}_j) \right] \tag{14}$$

or

$$\max_{\mathbf{K}_i \in \mathbf{KC}\text{-}\mathbf{KS}_{m-1}} \left[KA(\mathbf{K}_i, \mathbf{L}) \Big/ \frac{1}{m-1} \sum_{\mathbf{K}_j \in \mathbf{KS}_{m-1}} KA(\mathbf{K}_i, \mathbf{K}_j) \right] \tag{15}$$

where $KA(\mathbf{K}_1, \mathbf{K}_2)$ is the kernel alignment between kernel matrices $\mathbf{K}_1$ and $\mathbf{K}_2$, and given by

$$KA(\mathbf{K}_1, \mathbf{K}_2) = \frac{< \mathbf{K}_1, \mathbf{K}_2 >_{\mathrm{F}}}{\sqrt{< \mathbf{K}_1, \mathbf{K}_1 >_{\mathrm{F}} < \mathbf{K}_2, \mathbf{K}_2 >_{\mathrm{F}}}}. \tag{16}$$

It is evident that the method in [61] employs a greedy search strategy, i.e., forward selection, to find the expected kernels from $\mathbf{KC}$. Although greedy approaches are more computationally efficient than the brute search method, they tend to only produce a locally optimal base kernel set.

A limitation of kernel alignment is that it does not consider the unbalanced class distribution, which may cause the sensitivity of the measure to drop drastically. Cortes et al. [28] proposed to center kernels (or kernel matrices) before computing the alignment measure to cancel the effect of unbalanced class distribution. Centered kernel alignment (CKA) is defined as

$$CKA(\mathbf{K}_1, \mathbf{K}_2) = \frac{< \bar{\mathbf{K}}_1, \bar{\mathbf{K}}_2 >_{\mathrm{F}}}{\sqrt{< \bar{\mathbf{K}}_1, \bar{\mathbf{K}}_1 >_{\mathrm{F}} < \bar{\mathbf{K}}_2, \bar{\mathbf{K}}_2 >_{\mathrm{F}}}}. \tag{17}$$

Although this improved definition of alignment may appear to be a technicality, it is actually a critical difference. Without that centering, the definition of alignment does not correlate well with the performance of learning machines [28]. It is interesting to note the relationship between the proposed MKL method

using HSIC Lasso and MKL using CKA. The MKL using CKA can be expressed as

$$\max\ CKA(\sum_{m=1}^{M}\mu_m\mathbf{K}_m,\mathbf{L})=\frac{<\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m,\bar{\mathbf{L}}>_{\mathrm{F}}}{\sqrt{<\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m,\sum_{o=1}^{M}\mu_m\bar{\mathbf{K}}_m>_{\mathrm{F}}<\bar{\mathbf{L}},\bar{\mathbf{L}}>_{\mathrm{F}}}}$$
$$\text{s.t. } \mu_1,\cdots,\mu_M\geq 0 \tag{18}$$

which is equivalent to

$$\begin{aligned}&\max\ <\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m,\bar{\mathbf{L}}>_{\mathrm{F}}\\ &\text{s.t. } <\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m,\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m>_{\mathrm{F}}<\bar{\mathbf{L}},\bar{\mathbf{L}}>_{\mathrm{F}}=C_1,\\ &\quad \mu_1,\cdots,\mu_M\geq 0\end{aligned} \tag{19}$$

where $C_1>0$ is a constant. Let $C_2=C_1/<\bar{\mathbf{L}},\bar{\mathbf{L}}>_{\mathrm{F}}$, (19) can be rewritten as

$$\begin{aligned}&\max\ <\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m,\bar{\mathbf{L}}>_{\mathrm{F}}\\ &\text{s.t. } <\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m,\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m>_{\mathrm{F}}=C_2,\\ &\quad \mu_1,\cdots,\mu_M\geq 0.\end{aligned} \tag{20}$$

Using the Lagrange multiplier method, (20) is equivalent to

$$\max\ <\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m,\bar{\mathbf{L}}>_{\mathrm{F}}-\eta\left(<\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m,\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m>_{\mathrm{F}}-C_2\right)$$
$$\text{s.t. } \mu_1,\cdots,\mu_M\geq 0 \tag{21}$$

where $\eta>0$. Since the alignment is invariant to rescaling $\boldsymbol{\mu}$, we can choose $\eta=1/2$. Therefore, the MKL using CKA can be finally represented as

$$\begin{aligned}&\max\ <\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m,\bar{\mathbf{L}}>_{\mathrm{F}}-\frac{1}{2}<\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m,\sum_{m=1}^{M}\mu_m\bar{\mathbf{K}}_m>_{\mathrm{F}}\\ &\text{s.t. } \mu_1,\cdots,\mu_M\geq 0.\end{aligned} \tag{22}$$

Comparing (12) and (22), it is clear that the difference between the proposed MKL method using HSIC Lasso and the MKL using CKA is that the latter does not have the L_1-regularization term. Without this constraint term, the kernel can over-fit its alignment to the training set, making its alignment to the test set poor [34].

B. The Second Stage: Training FSVM with HSIC-based Fuzzy Membership

A limitation of traditional SVM-based MKL approaches is that they do not consider the noise sensitivity problem, which originates from the assumption that each training point has equal importance or weight in the training process. To address this issue, we propose the employment of FSVM rather than SVM for MKL. A key step before training FSVM is to determine the fuzzy memberships of training samples. Although many techniques to define membership functions exist and well-defined membership functions can improve classification performance, there are so far no general guidelines for determining them [46]-[52].

In this study, we present a strategy in which the reasonable fuzzy memberships for FSVM are set using the HSIC. For a binary classification problem, let $\boldsymbol{y}=(y_1,\cdots,y_n)^{\mathrm{T}}$, thus the HSIC can be rewritten as

$$\begin{aligned}HSIC(\mathbf{K},\mathbf{L})&=\frac{1}{n^2}<\bar{\mathbf{K}},\mathbf{L}>_{\mathrm{F}}=\frac{1}{n^2}<\bar{\mathbf{K}},\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}>_{\mathrm{F}}\\ &=\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}y_iy_j\bar{k}(\boldsymbol{x}_i,\boldsymbol{x}_j)\\ &=\frac{1}{n^2}\left[\sum_{y_i=y_j}\bar{k}(\boldsymbol{x}_i,\boldsymbol{x}_j)-\sum_{y_i\neq y_j}\bar{k}(\boldsymbol{x}_i,\boldsymbol{x}_j)\right]\end{aligned} \tag{23}$$

where $\bar{k}(\boldsymbol{x}_i,\boldsymbol{x}_j)$ is the centered kernel function and $\bar{k}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\bar{\mathbf{K}}_{i,j}$. Given the specific input data $(\boldsymbol{x}_t,y_t)$, where $1\leq t\leq n$, we define the instance HSIC (called I-HSIC) as

$$I\text{-}HSIC(\mathbf{K},\mathbf{L},\boldsymbol{x}_t)=\frac{1}{n^2}\left[\sum_{y_t=y_i}\bar{k}(\boldsymbol{x}_t,\boldsymbol{x}_i)-\sum_{y_t\neq y_i}\bar{k}(\boldsymbol{x}_t,\boldsymbol{x}_i)\right]. \tag{24}$$

Since the kernel $\bar{k}(\boldsymbol{x}_t,\boldsymbol{x}_i)$ measures the similarity between points $\boldsymbol{x}_t$ and $\boldsymbol{x}_i$, it is clear that the I-HSIC will increase if the similarity represented by the kernel is large for input patterns of the same class and small for patterns from different classes. In other words, data points with a larger I-HSIC value should make a greater contribution to classification performance and those with smaller I-HSIC value can be considered as outliers or noises. Hence, we can use I-HSIC as the scoring function, denoted by $score_t$, which measures the degree of importance of data point $\boldsymbol{x}_t$ for classification. Similar to the method used in [52], we apply the following linear map function to map the scores into fuzzy membership values in the unit interval:

$$s_t=\frac{score_t-\min\{score_i\}_{i=1}^{n}}{\max\{score_i\}_{i=1}^{n}-\min\{score_i\}_{i=1}^{n}} \tag{25}$$

where $\min\{score_i\}_{i=1}^{n}$ and $\max\{score_i\}_{i=1}^{n}$ are the minimum and maximum of the scores of the training data points, respectively.

C. The Overall Procedure of HSIC-FMKL and its Computational Complexity

Putting the above two parts together, we sketch the overall procedure of the proposed HSIC-FMKL in Algorithm 1, where the centered kernel matrix can be calculated by

$$\bar{\mathbf{K}}_{ij}=\mathbf{K}_{ij}-\frac{1}{n}\sum_{i=1}^{n}\mathbf{K}_{ij}-\frac{1}{n}\sum_{j=1}^{n}\mathbf{K}_{ij}+\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{K}_{ij} \tag{26}$$

Algorithm 1. HSIC-FMKL

Input: Labeled data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, base kernels $\{k_m(\cdot,\cdot)\}_{m=1}^M$ or kernel matrices $\{\mathbf{K}_m\}_{m=1}^M$, and regularization parameters C and λ.

Output: Fuzzy SVM classifier $f(\boldsymbol{x})$.

1: Initialize $\boldsymbol{\mu} = \boldsymbol{e}/M$.

2: Calculate the kernel matrix $\mathbf{L} = \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}$.

3: Calculate the centered kernel matrices $\bar{\mathbf{L}}$ and $\{\bar{\mathbf{K}}_m\}_{m=1}^M$.

4: Obtain $\boldsymbol{\mu}$ by solving (12).

5: Normalize each element of $\boldsymbol{\mu}$ as $\mu_m \to \mu_m \big/ \sum_{m=1}^M \mu_m$.

6: Obtain $\{s_i\}_{i=1}^n$ using (24) and (25).

7: Combine the kernel matrices using the weight $\boldsymbol{\mu}$ and train a fuzzy SVM classifier.

We analyze the computational complexity of Algorithm 1 with the O notation. First, the computational complexity of calculating centered kernel matrices in step 3 is $O(Mn^2)$. Second, the complexity of the quadratic programming (QP) solver in step 4 is $O(TM^3)$ with T being the number of iterations in solving (12). Third, calculating the fuzzy memberships in step 6 costs $O(M+n)$. Lastly, note that empirically the SVM training complexity is $O(n^{2.3})$ [62] and given the fuzzy memberships, the training cost of fuzzy SVM is the same as the SVM, thus the computational complexity of step 7 is $O(M+n^{2.3})$. The total computational complexity of our proposed HSIC-FMKL is therefore

$$\begin{aligned} &O(Mn^2)+O(M^3)+O(M+n)+O(M+n^{2.3}) \\ &= O(Mn^2+M^3+n^{2.3}). \end{aligned} \tag{27}$$

It should be noted that we suppose that multiple base kernels (kernel matrices) can be precomputed and loaded into memory prior to the HSIC-FMKL training. The computational cost of calculating the base kernels is then ignored.

IV. Experiments

In this section, we perform extensive experiments on binary classification problems to evaluate the efficacy of the proposed HSIC-FMKL approach. To validate the fuzzy technology employed in HSIC-FMKL, we also train an SVM on the same combination of the base kernel matrices used by HSIC-FMKL, which we refer to as HSIC-MKL.

A. Comparison Approaches and Parameter Settings

We compare HSIC-FMKL and HSIC-MKL with the following state-of-the-art kernel learning algorithms:

- AvgMKL: The average combination of multiple base kernels. It has been reported that AvgMKL is competitive with many algorithms [8] [9].
- SimpleMKL [14]: An algorithm reformulates the mixed-norm regularization of MKL problem as the weighted 2-norm regularization, and the L_1-norm is imposed on kernel weights.
- LpMKL [19]: An algorithm generalizes the regular L_1-norm MKL to arbitrary L_p-norm ($p>1$) MKL. We adopt the cutting plane algorithm with second order Taylor approximation of L_p.
- SMKL [20]: An algorithm employs the entropy of the kernel weights and transforms the non-smooth function induced by the L_1-norm simplex constraint into a smooth one.
- BM^{3}KL [27]: A Bayesian max margin MKL with the Dirichlet prior and Three Parameter Beta Normal (TPBN) prior imposed on the kernel weights and sample weights, respectively.
- CKA-MKL [28]: The two-stage MKL with centered kernel alignment.

For parameter settings, the regularization parameters C and λ are determined by 5-fold cross-validation on the training set. We perform grid-search in one dimension (i.e., a line-search) to choose the regularization parameters C from the set $\{10^{-2}, 10^{-1}, \cdots, 10^{2}\}$ for all the compared methods except BM^{3}KL. We perform grid-search over two dimensions for our proposed HSIC-FMKL and HSIC-MKL approaches, i.e., $C = \{10^{-2}, 10^{0}, \cdots, 10^{2}\}$ and $\lambda = \{10^{-2}, 10^{-1}, \cdots, 10^{2}\}$. In addition, we examine $p = 2, 3, 4$ for LpMKL and report the best results. The hyper-parameter settings of BM^{3}KL are the same as those in [27]. All methods are implemented using MATLAB in the SVM-KM toolbox [2] framework. Note that SimpleMKL has been implemented in the SimpleMKL software package [3], which requires the SVM-KM toolbox.

TABLE I
STATISTICS OF THE SELECTED NINE DATA SETS FROM UCI

Data set	Number of samples	Number of features	Original data set
Australian	690	14	Australian Credit Approval
Breast	683	9	Breast Cancer Wisconsin (Original)
Diabetes	768	8	Pima Indians Diabetes
German	1000	20	German Credit Data
Heart	270	13	Heart
Ionosphere	351	34	Ionosphere
Liver	345	7	Liver Disorders
Sonar	208	60	Sonar
Spambase	4601	57	Spambase

[2] http://asi.insa-rouen.fr/enseignants/~arakoto/toolbox/
[3] http://asi.insa-rouen.fr/enseignants/~arakoto/code/mklindex.html

TABLE II
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT MKL ALGORITHMS ON UCI DATA SETS

Data set	Classification accuracy (%)							
	AvgMKL	SimpleMKL	LpMKL	SMKL	BM^{3}KL	CKA-MKL	HSIC-MKL	HSIC-FMKL
Australian	66.8±4.5	85.1±1.3	84.6±1.7	85.2±1.2	86.0±1.6	**87.2±0.4**	**86.7±1.3**	**87.0±0.9**
Breast	95.4±0.9	**96.6±0.7**	**96.1±0.6**	**96.5±0.6**	**96.2±1.2**	**96.4±0.9**	**96.6±1.0**	**97.0±1.2**
Diabetes	65.3±1.8	75.9±2.3	72.7±2.4	75.8±2.7	74.5±3.0	75.3±3.5	77.1±2.2	**78.5±0.7**
German	69.6±1.4	71.5±2.6	**74.4±1.5**	**74.5±2.2**	**74.7±1.7**	72.0±1.2	72.4±0.9	73.6±1.8
Heart	75.5±5.3	**83.1±2.8**	80.6±3.6	81.4±3.5	81.0±3.4	82.1±1.8	**83.3±2.6**	**83.7±3.2**
Ionosphere	91.2±1.8	93.5±1.2	94.8±2.1	95.1±2.6	95.3±1.9	93.7±1.0	95.5±0.8	**96.3±1.5**
Liver	57.4±2.1	62.4±4.3	69.3±2.8	69.0±1.7	69.7±2.3	68.8±1.6	70.0±2.9	**71.5±2.3**
Sonar	59.0±8.7	78.2±3.5	**84.7±3.3**	**84.9±3.5**	84.1±3.8	81.3±2.8	81.8±3.2	83.4±2.6
Spambase	88.3±1.2	88.9±0.8	89.5±1.4	90.0±0.7	89.8±1.3	90.0±0.9	90.5±1.2	**91.2±1.1**

TABLE III
SIGNIFICANCE TEST OF CLASSIFICATION RESULTS ON UCI DATA SETS

Data set	Win-tie-loss (W-T-L)						
	HSIC-MKL vs. AvgMKL	HSIC-MKL vs. SimpleMKL	HSIC-MKL vs. LpMKL	HSIC-MKL vs. SMKL	HSIC-MKL vs. BM^{3}KL	HSIC-MKL vs. CKA-MKL	HSIC-FMKL vs. HSIC-MKL
Australian	W	W	W	W	W	T	T
Breast	W	T	T	T	T	T	T
Diabetes	W	W	W	W	W	W	W
German	W	W	L	L	L	T	W
Heart	W	T	W	W	W	W	T
Ionosphere	W	W	W	T	T	W	W
Liver	W	W	W	W	T	W	W
Sonar	W	W	L	L	L	T	W
Spambase	W	W	W	W	W	T	W

B. Experiments on UCI Data Sets

We select nine popular binary classification data sets, i.e., *Australian Credit Approval*, *Breast Cancer Wisconsin (Original)*, *Pima Indians Diabetes*, *German Credit Data*, *Heart*, *Ionosphere*, *Liver Disorders*, *Sonar*, and *Spambase* from the UCI machine learning repository [63]. For *Breast Cancer Wisconsin (Original)*, we directly eliminate the samples with missing attribute values. Table I provides the statistics of these data sets. The short name of each data set is presented, as well as the number of samples, the number of features, and the original name of the data set.

We partition each data set into a training set and a test set by stratified sampling (whereby which the object generation follows class prior probabilities): 50% of the data set serves as the training set and the remaining 50% forms the test set. The training samples are normalized to be of zero mean and unit variance, and the test samples are also normalized using the same mean and variance as the training data. Following the settings of previous MKL studies [10], [14], [20], and [27] , we use the Gaussian kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 / 2\sigma^2)$ and polynomial kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^d$ as the base kernels:

- Gaussian kernels with 10 different widths $\sigma \in \{2^{-3}, 2^{-2}, \cdots, 2^6\}$ on each individual feature as well as all features.
- Polynomial kernels with three different degrees $d \in \{1, 2, 3\}$ on each individual feature as well as all features.

All kernel matrices are normalized to unit trace and precomputed prior to running the algorithms. It is clear that there are $13(n+1)$ base kernels (i.e. $M = 13(n+1)$) in total used for MKL, where n is the number of features.

To obtain stable results, we independently split each data set and then run each algorithm on it 20 times. The average classification accuracy and standard deviation of each algorithm are reported in Table II. To conduct a rigorous comparison, the paired *t*-test [64] [65] is performed. The paired *t*-test is used to analyze whether the difference between two compared algorithms on one data set is significant. The *p*-value of the paired *t*-test represents the probability that two sets of compared results come from distributions with an equal mean. A *p*-value of 0.05 is considered to be statistically significant. The win-tie-loss (W-T-L) summarizations based on the paired t -test are listed in Table III, where HSIC-MKL and AvgMKL,

SimpleMKL, LpMKL, SMKL, BM^{3}KL, and CKA-MKL are respectively compared. HSIC-FMKL and HSIC-MKL are also compared. In comparing two algorithms such as algorithm 1 vs. algorithm 2, a win or a loss means that algorithm 1 is better or worse than algorithm 2 on a data set. A tie means that both algorithms achieve the same performance. In Table II, for each data set, the boldface with underline denote the best performance of MKL methods and those that exhibit no statisitcal difference with the best method are written in bold without underline.

From Tables II and III, we find that the proposed HSIC-MKL and HSIC-FMKL achieve superior performance to other baseline approaches, and that HSIC-FMKL consistently achieves the overall best classification performance. Of the nine data sets evaluated, SMKL, BM^{3}KL, and CKA-MKL report one best result, respectively, while our HSIC-FMKL reports six best results. We make two observations concerning the significance test. First, although HSIC-MKL is outperformed by LpMKL on the *German* and *Sonar* data sets, it produces significantly better classification performance than LpMKL on the *Australian*, *Diabetes*, *Heart*, *Ionosphere*, *Liver*, and *Spambase* data sets. Similar analysis can be done when HSIC-MKL and SMKL, and HSIC-MKL and BM^{3}KL are compared. Compared with CKA-MKL, HSIC-MKL significantly outperforms CKA-MKL on the *Diabetes*, *Heart*, *Ionosphere*, and *Liver* data sets, and yields the same performance on the rest of the data sets. Overall, HSIC-MKL is better than LpMKL, SMKL, BM^{3}KL, and CKA-MKL. Second, HSIC-FMKL significantly outperforms HSIC-MKL on the *Diabetes*, *German*, *Ionosphere*, *Liver*, *Sonar*, and *Spambase* data sets, and yields the same performance on the rest of the data sets. Clearly, the performance of HSIC-MKL is further improved by HSIC-FMKL. We attribute this to the fact that the HSIC-FMKL model assigns a weight to each sample as its fuzzy membership to reflect its corresponding confidence level or importance, and thus effectively suppresses the influence of noises or outliers during the training process.

C. Experiments on Protein Fold Prediction

Protein fold recognition is an important method of structure discovery in computational biology that does not rely on sequence similarity [66]. It is crucial for drug design since the function of a protein is closely linked to its folding class. We utilize the *Protein Fold Prediction* data set [4], composed of 27 folds which have six or more proteins and represent all major structure classes: α, β, α/β, and $\alpha+\beta$ [66]. The folds and corresponding number of proteins for training and testing are shown in Table IV. Collectively, this data set consists of 27 fold classes with 311 proteins used for training and 383 proteins for testing. In addition, there are 12 feature representations from different sources in this data set, which are summarized in Table V.

We construct a binary classification problem by combining the major structure classes $\{\alpha, \beta\}$ as one class and $\{\alpha/\beta, \alpha+\beta\}$ as the other class. We employ the second-order polynomial kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^2$ for each global representation (COM, SEC, HYD, VOL, POL, PLZ, L1, L4, L14, and L30) and linear kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$ for each local representation (BLO and PAM) as they provide better embedding of the feature representations [67]. All kernel matrices are normalized to unit trace and precomputed prior to running the algorithms.

TABLE IV
PROTEIN FOLDS AND CORRESPONDING NUMBER OF TRAINING AND TEST SAMPLES

Fold	Number of training samples	Number of testing samples
α		
Globin-like	13	6
Cytochrome *c*	7	9
DNA-binding 3-helical bundle	12	20
4-helical up-and-down bundle	7	8
4-helical cytokines	9	9
Alpha; EF-hand	6	9
β		
Immunoglobulin-like β-sandwich	30	44
Cupredoxins	9	12
Viral coat and capsid proteins	16	13
ConA-like lectins/glucanases	7	6
SH3-like barrel	8	8
OB-fold	13	19
Trefoil	8	4
Trypsin-like serine proteases	9	4
Lipocalins	9	7
α/β		
(TIM)-barrel	29	48
FAD (also NAD)-binding motif	11	12
Flavodoxin-like	11	13
NAD(P)-binding Rossmann-fold	13	27
P-loop containing nucleotide	10	12
Thioredoxin-like	9	8
Ribonuclease H-like motif	10	12
Hydrolases	11	7
Periplasmic binding protein-like	11	4
$\alpha+\beta$		
β-grasp	7	8
Ferredoxin-like	13	27
Small inhibitors, toxins, lectins	13	27

Fig. 1 presents the classification performance of SVM classifiers with individual base kernels, which are constructed on the corresponding feature representations. Fig .2 shows the performance comparison of the MKL algorithms in terms of classification accuracy with different combinations of base kernels: kernels on feature representations 1 to 6 (COM, SEC,

[4] http://mldata.org/repository/data/viewslug/protein-fold-prediction-ucsd-mkl/

HYD, VOL, POL, and PLZ), kernels on feature representations 1 to 10 (COM, SEC, HYD, VOL, POL, PLZ L1, L4, L14, and L30), and kernels on all feature representations. It is evident that all the MKL algorithms outperform the single kernel method. More importantly, the proposed HSIC-MKL and HSIC-FMKL are the two best algorithms, and that HSIC-FMKL achieves the highest accuracy. In additon, it is worth noting that higher classification accuracies are obtained with a larger number of base kernels for all the comparion MKL algorithms. This observation suggests that these feature representations as a whole carry complementary (rather than conflict) discriminative information and MKL can utilize such information to improve predictive performance.

TABLE V
MULTIPLE FEATURE REPRESENTATIONS FROM DIFFERENT SOURCES

ID	Feature representation	Data source	Number of features
1	COM	Amino-acid composition	20
2	SEC	Predicted secondary structure	21
3	HYD	Hydrophobicity	21
4	VOL	Van der Waals volume	21
5	POL	Polarity	21
6	PLZ	Polarizability	21
7	L1	Pseudo amino-acid composition at interval 1	22
8	L4	Pseudo amino-acid composition at interval 4	28
9	L14	Pseudo amino-acid composition at interval 14	48
10	L30	Pseudo amino-acid composition at interval 30	80
11	BLO	Smith-Waterman scores with the BLOSUM 62 matrix	311
12	PAM	Smith-Waterman scores with the PAM 50 matrix	311

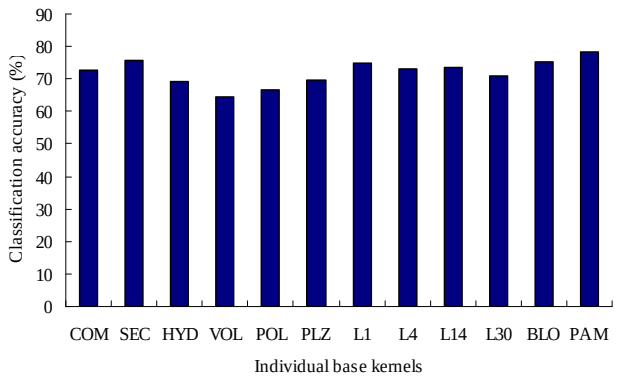

Fig. 1. Classification accuracies of SVM classifiers with individual base kernels for Protein Fold Prediction

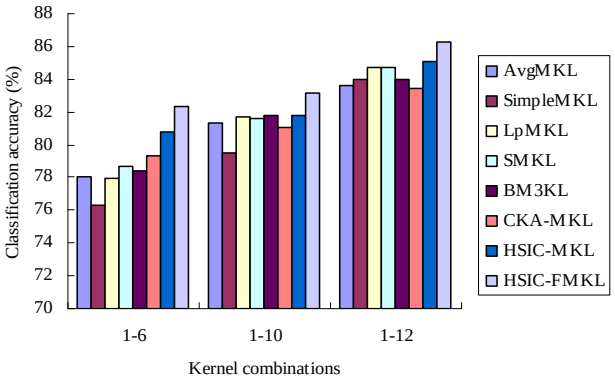

Fig. 2. Classification accuracy comparison of MKL algorithms for Protein Fold Prediction

V. CONCLUSION AND FURTHER STUDY

This paper presents an effective two-stage fuzzy MKL algorithm based on the notion of HSIC, called HSIC-FMKL. In discussing the connection between MKL and HSIC Lasso, we find that the proposed algorithm not only has a clear statistical interpretation that minimum redundant kernels with maximum dependence on output labels are found and combined, but also can efficiently compute the global optimal solution by solving a Lasso optimization problem. To further improve the accuracy of traditional MKL methods, the notion of instance HSIC (denoted by I-HSIC) is introduced in the second stage of this algorithm to calculate the fuzzy memberships for constructing a fuzzy SVM classifier, which is used to select the final prediction hypothesis. Comprehensive experiments on a number of benchmark data sets demonstrate the promising results of the proposed algorithm.

Future investigation will focus on the theoretical analysis of the proposed HSIC-FMKL. The classification ability (generalization to unseen data) of the SVM-based MKL is generally decided by two factors: the empirical error on the training data and the complexity of the classifier. In [68], the empirical Rademacher complexity is used to quantify the complexity of the classifier. Generally, a more complex classifier demonstrates a better classification performance on the training data. Hence, the optimal classification ability is a tradeoff between the empirical error and the empirical Rademacher complexity. Since the classification ability is quantified by the generalizaiton error, we will attempt to develop a convergence bound of the generalization error of HSIC-FMKL based on the established theory of Rademacher complexities. Besides, expanding the proposed model to multiple kernel clustering [69], extreme learning machine [70] and domain transfer learning [71], as well as adding more complicated complicated kernels such as the Chebyshev kernel and Hermite kernel [72] into the pool of base kernels for MKL are also important issues to be investigated.

Last but not least, it is interesting to further investigate

feature selection based on the relevance redundancy trade-off criteria [73], [74], which aims to find non-redundant features with strong dependence on output labels. For example, since the statistical interpretation of MKL using HSIC Lasso is very similar to feature selection using the minimum redundancy maximum relevance criterion [59], [60], it is worth considering a unified view on kernel learning and feature selection based on relevance redundancy trade-off criteria. By treating various learning models under a unifed framework and elucidating their relations, we also believe the research results on fuzzy multiple kernel learning can be applied in data-driven decision support systems for, e.g., business intelligence [75].

REFERENCES

[1] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," ***IEEE Transactions on Neural Networks***, vol. 12, no. 2, pp. 181-202, Mar. 2001.

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

[3] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," ***The Annals of Statistics***, vol. 36, no. 3, pp. 1171-1220, 2008.

[4] O. Chapelle, V. Vapnik, and S. Mukherjee, "Choosing multiple parameters for support vector machines," ***Machine Learning***, vol. 46, no. 2, pp. 131-159, 2002.

[5] T. Wang, S. Tian, H. Huang, and D. Deng, "Learning by local kernel polarization," ***Neurocomputing***, vol. 72, nos. 13-15, pp. 3077-3084, 2009.

[6] T. Wang, D. Zhao, and S. Tian, "An overview of kernel alignment and its applications," ***Artificial Intelligence Review***, vol. 43, no. 2, pp. 179-192, 2015.

[7] B. Pan, W.-S. Chen, C. Xu, and B. Chen, "A novel framework for learning geometry-aware kernels", ***IEEE Transactions on Neural Networks and Learning Systems***, vol. 27, no. 5, pp. 939-951, May 2016.

[8] M. Gönen and E. Alpayın, "Multiple kernel learning algorithms," ***Journal of Machine Learning Research***, vol. 12, pp. 2211-2268, 2011.

[9] S. S. Bucak, R. Jin, and A. K. 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, Jul. 2014.

[10] 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***, Haifa, Israel, 2010, pp. 1175-1182.

[11] G. R. G. Lanckriet, N. Cristianini, P. 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.

[12] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in ***Proceedings of the 21st International Conference on Machine Learning***, Banff, Canada, 2004, pp. 41-48.

[13] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," ***Journal of Machine Learning Research***, vol. 7, pp. 1531-1565, 2006.

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

[15] 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.

[16] A. Afkanpour, A. György, C. Szepesvári, and M. Bowling. A randomized mirror descent algorithm for large scale multiple kernel learning," in ***Proceedings of the 30th International Conference on Machine Learning***, Atlanta, USA, 2013, pp. 374-382.

[17] F. Aiolli and M. Donini, "EasyMKL: A scalable multiple kernel learning algorithm," ***Neurocomputing***, vol. 169, pp. 215-224, 2015.

[18] M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg, "Non-sparse multiple kernel learning," in ***Proceedings of the NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels***, 2008.

[19] 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.

[20] Z. Xu, R. Jin, S. Zhu, M. R. Lyu, and I. King, "Smooth optimization for effective multiple kernel learning," in ***Proceedings of the 24th AAAI Conference on Artificial Intelligence***, Atlanta, USA, 2010.

[21] M. Kowalski, M. Szafranski, and L. Ralaivola, "Multiple indefinite kernel learning with mixed norm regularization," in ***Proceedings of the 26th International Conference on Machine Learning***, Montreal, Canada, 2009, pp. 545-552.

[22] M. Varma and B. R. Babu, "More generality in efficient multiple kernel learning," in ***Proceedings of the 26th International Conference on Machine Learning***, Montreal, Canada, 2009, pp. 1065-1072.

[23] 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.

[24] Y. Zhou, N. Hu, and C. J. Spanos, "Veto-consensus multiple kernel learning," in ***Proceedings of the 30th AAAI Conference on Artificial Intelligence***, Phoenix, USA, 2016, pp. 2407-2414.

[25] Y. Han, K. Yang, Y. Ma, and G. Liu, "Localized multiple kernel learning via sample-wise alternating optimization," ***IEEE Transactions on Cybernetics***, vol. 44, no. 1, pp. 137-148, Jan. 2014.

[26] X. Liu, L. Wang, J. Zhang, and J. Yin, "Sample-adaptive multiple kernel learning," in ***Proceedings of the 28th AAAI Conference on Artificial Intelligence***, Québec, Canada, 2014, pp. 1975-1981.

[27] C.-Y, Du, C.-D, Du, G.. Long, X. Jin, and Y. Li, "Efficient Bayesian maximum margin multiple kernel learning," in ***Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases***, Riva del Garda, Italy, Part I, Lecture Notes in Computer Science 9852, 2016, pp. 165-181.

[28] 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.

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

[30] T. Wang, D. Zhao, and Y. Feng, "Two-stage multiple kernel learning with multiclass kernel polarization," ***Knowledge-Based System***s, vol. 48, pp. 10-16, 2013.

[31] A. Nazarpour and P. Adibi, "Two-stage multiple kernel learning for supervised dimensionality reduction," ***Pattern Recognition***, vol. 48, no. 5, pp. 1854-1862, 2015.

[32] O. Chapelle and A. Rakotomamonjy, "Second order optimization of kernel parameters," in ***NIPS 2008 Workshop on Kernel Learning: Automatic Selection of Optimal Kernel***s, Whistler, Canada, 2008.

[33] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S. Kandola, "On kernel-target alignment," in ***Advances in Neural Information Processing Systems*** 14, 2001, pp. 367-373.

[34] J. S. Kandola, J. Shawe-Taylor J, and N. Cristianini, "Optimizing kernel alignment over combinations of kernels," ***Technical Report 121***, Department of Computer Science, University of London, UK, 2002.

[35] Y. Baram, "Learning by kernel polarization," ***Neural Computation***, vol. 17, no. 6, pp. 1264-1275, 2005.

[36] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in ***Proceedings of the 16th International Conference on Algorithmic Learning Theory***, Singapore, 2005, pp. 63-77.

[37] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. Smola, "A kernel statistical test of independence," in ***Advances in Neural Information Processing System***s 20, 2007, pp. 585-592.

[38] K. Chwialkowski and A. Gretton, "A kernel independence test of random process," in ***Proceedings of the 31st International Conference on Machine Learning***, Beijing, China, 2014, pp. 1422-1430.

[39] T. Wang and W. Li, "Kernel learning and optimization with Hilbert-Schmidt independence criterion," ***International Journal of Machine Learning and Cybernetics***, DOI: 10.1007/s13042-017-0675-7, pp. 1-14, 2017.

[40] L. Song, A. Smola, A. Gretton, and K. Borgwardt, "A dependence maximization view of clustering," in ***Proceedings of the 24th International Conference on Machine Learning***, Corvallis, USA, 2007, pp. 823-830.

[41] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt, "Feature selection via dependence maximization," ***Journal of Machine Learning Research***, vol. 13, pp. 1393-1434, 2012.

[42] X. Shu, D. Lai, H. Xu, and L. Tao, "Learning shared subspace for multi-label dimensionality reduction via dependence maximization," ***Neurocomputing***, vol. 168, pp. 356-364, 2015.

[43] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," ***Artificial Intelligence Review***, vol. 22, no. 3, pp. 177-210, 2004.

[44] X. Zhang, "Using class-center vectors to build support vector machines," in ***Proceedings of the 1999 IEEE Signal Processing Society Workshop: Neural Networks for Signal Processing IX***, Madison, USA, 1999, pp. 3-11.

[45] R. Tibshirani, "Regression shrinkage and selection via the lasso," ***Journal of the Royal Statistical Society***, ***Series B***, vol. 58, no. 1. pp. 267-288, 1996.

[46] C.-F. Lin and S.-D. Wang, "Fuzzy support vector machines," ***IEEE Transactions on Neural Networks***, vol. 13, no. 2, pp. 464-471, Mar. 2002.

[47] C.-F. Lin and S.-D. Wang, "Training algorithms for fuzzy support vector machines with noisy data," ***Pattern Recognition Letters***, vol. 25, no. 14, pp. 1647-1656, 2004.

[48] Y. Wang, S. Wang, and K. K. Lai, "A new fuzzy support vector machine to evaluate credit risk," ***IEEE Transactions on Fuzzy Systems***, vol. 13, no. 6, pp. 820-831, Dec. 2005.

[49] R. Batuwita and V. Palade, "FSVM-CIL: Fuzzy support vector machines for class imbalance learning," ***IEEE Transactions on Fuzzy Systems***, vol. 18, no. 3, pp. 558-571, Jun. 2010.

[50] W. M. Tang, "Fuzzy SVM with a new fuzzy membership function to solve the two-class problems," ***Neural Processing Letters***, vol. 34, pp. 209-219, 2011.

[51] X. Yang, G. Zhang, J. Lu, and J. Ma, "A kernel fuzzy c-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises," ***IEEE Transactions on Fuzzy Systems***, vol. 19, no. 1, pp. 105-115, Feb. 2011.

[52] S.-T. Lin and C.-C. Chen, "A regularized monotonic fuzzy support vector machine model for data mining with prior knowledge," ***IEEE Transactions on Fuzzy Systems***, vol. 23, no. 5, pp. 1713-1727, Oct. 2015.

[53] M. Yamada, A. Kimura, F. Naya, and H. Sawada, "Change-point detection with feature selection in high-dimensional time-series data," in ***Proceedings of the 23rd International Joint Conference on Artificial Intelligence***, Beijing, China, 2013, pp. 1827-1833.

[54] M. Yamada, W. Jitkrittum, L. Sigal, E.P. Xing, and M. Sugiyama, "High-dimensional feature selection by feature-wise kernelized Lasso," ***Neural Computation***, vol. 26, no. 1, pp. 185-207, 2014.

[55] D. He, I. Rish, and L. Parida, "Transductive HSIC Lasso," in ***Proceedings of the SIAM International Conference on Data Mining***, Philadelphia, USA, 2014, pp. 154-162.

[56] I. Steinwart, "On the influence of the kernels on the consistency of support vector machines," ***Journal of Machine Learning Research***, vol. 2, pp. 67-93, 2001.

[57] R. Tomioka and M. Sugiyama, "Dual-augmented Lagrangian method for efficient sparse reconstruction," ***IEEE Signal Processing Letters***, vol. 16, no. 12, pp. 1067-1070, Dec. 2009.

[58] R. Tomioka and M. Sugiyama, "Super-linear convergence of dual augmented Lagrangian algorithm for sparsity regularized estimation," ***Journal of Machine Learning Research***, vol. 12, pp. 1537-1586, 2011.

[59] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," ***IEEE Transactions on Pattern Analysis and Machine Intelligence***, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[60] A. Unler, A. Murat, and R. B. Chinnam, "Mr2PSO: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification," ***Information Sciences***, vol. 181, no. 20, pp. 4625-4641, 2011.

[61] P. Wu, F. Duan, and P. Guo, "A pre-selecting base kernel method in multiple kernel learning," ***Neurocomputing***, vol. 165, pp. 46-53, 2015.

[62] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," ***Advances in Kernel Methods: Support Vector Learning***, pp. 185–208, MIT Press, 1999.

[63] M. Lichman, "UCI machine learning repository," Irvine, CA: University of California, School of Information and Computer Science, 2013. [http://archive.ics.uci.edu/ml/].

[64] V. Vapnik, **Statistical Learning Theory**. Hoboken, New Jersey, USA: John Wiley & Sons, 1998.

[65] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," ***Journal of Machine Learning Research***, vol. 7, pp. 1-30, 2006.

[66] C. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural network," ***Bioinformatics***, vol. 17, no. 4, pp. 349-358, 2001.

[67] 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.

[68] P. Bartlett and S. Mendelson, "Rademacher and Gaussian complexities: Risk bounds and structural results," ***Journal of Machine Learning Research***, vol. 3, pp. 463-482, 2002.

[69] Y. Lu, L. Wang, J. Lu, J. Yang , and C. Shen, "Multiple kernel clustering based on centered kernel alignment," ***Pattern Recognition***, vol. 47, no. 5, pp. 3656-3664, 2014.

[70] P. Liu, Y. Huang, L. Meng, S. Gong, and G. Zhang, "Two-stage extreme learning machine for high-dimensional data," ***International Journal of Machine Learning and Cybernetics***, vol. 7, no. 5, pp. 765-772, 2016.

[71] 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, Mar. 2012.

[72] V. H. Moghaddam and J. Hamidzadeh, "New Hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier," ***Pattern Recognition***, vol. 60, pp. 921-935, 2016.

[73] M. García-Torres, F. Gómez-Vela, B. Melián-Batista, and J. M. Moreno-Vega, "High-dimensional feature selection via feature grouping: A variable neighborhood search approach," ***Information Sciences***, vol. 326, pp. 102-118, 2016.

[74] J. Che, Y. Yang, L. Li, X. Bai, S. Zhang, and C. Deng, "Maximum relevance minimum common redundancy feature selection for nonlinear data," ***Information Sciences***, vol. 409-410, pp. 68-86, 2017.

[75] M. Pratama, J. Lu, E. Lughofer, G. Zhang, and M. J. Er, "An incremental learning of concept drifts using evolving type-2 recurrent fuzzy neural networks," ***IEEE Transactions on Fuzzy Systems***, vol. 25, no. 5, pp. 1175-1192, Oct. 2017.

Tinghua Wang received the M.S. degree in computer science from Nanchang University, P.R.China, in 2006 and the Ph.D. degree in computer science from Beijing Jiaotong University, P.R.China, in 2010.

From Oct. 2011 to Oct. 2013, He was a Postdoctoral Research Fellow with the Institute of Computer Science and Technology, Peking University, P.R.China. From Mar. 2016 to Mar. 2017, he was a Visting Scholar with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. He is currently an Associate Professor with the School of Mathematics and Computer Science, Gannan Normal University, P.R.China. His research interests include machine learning and data mining.

Jie Lu (F'18) received the Ph.D. degree in information systems from Curtin University of Technology, Australia, in 2000.

She is currently a Distinguished Professor and the Associate Dean in Research Excellence with the Faculty of Engineering and Information Technology, and the

Director of Centre for Artificial Intelligence, University of Technology Sydney, Australia. She has published six research books and 400 papers in *Artificial Intelligence*, *IEEE transactions on Fuzzy Systems*, other refereed journals, and conference proceedings. She has won more than 20 Australian Research Council (ARC) discovery and other research grants for over $4 million in the last 15 years. Her research interests include fuzzy transfer learning, decision support systems, concept drift, and recommender systems.

Dr. Lu serves as Editor-In-Chief for *Knowledge-Based Systems* (Elsevier) and Editor-In-Chief for *International Journal on Computational Intelligence Systems* (Atlantis), Associate Editor for *IEEE Transactions on Fuzzy Systems*, Editor for books series on Intelligent Information Systems (World Scientific), and has served as a guest editor of 12 special issues for IEEE transactions and other international journals. She has delivered 20 keynote speeches at international conferences, and has chaired 10 IEEE and other international conferences. She is a Fellow of IEEE and a Fellow of IFSA.

Guangquan Zhang received the Ph.D. degree in applied mathematics from Curtin University of Technology, Australia, in 2001.

He is currently an Associate Professor and the Director of the Decision Systems and e-Service Intelligence (DeSI) Laboratory, Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. He has authored four monographs, five reference books, and over 400 papers, including over 200 refereed journal papers. His research interests include multiobjective and group decision making, decision support systems, fuzzy measure and optimization, fuzzy machine learning, and uncertain information processing.

Dr. Zhang serves as a member of the editorial boards for several international journals, and has served as a guest editor of eight special issues for IEEE transactions and other international journals, and has co-chaired several international conferences and workshops in the area of fuzzy decision-making and knowledge engineering.