Multiple Kernel Fusion with HSIC Lasso

Tinghua Wang(✉) and Fulai Liu

School of Mathematics and Computer Science, Gannan Normal University, Ganzhou 341000, People's Republic of China
wthgnnu@163.com

Abstract. Multiple kernel learning (MKL) is a principled way for kernel fusion for various learning tasks, such as classification, clustering and dimensionality reduction. In this paper, we develop a novel multiple kernel learning model based on the Hilbert-Schmidt independence criterion (HSIC) for classification (called HSIC-MKL). In the proposed HSIC-MKL model, we first propose a 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 the global optimal solution can be computed efficiently by solving a Lasso optimization problem. After the optimal kernel is obtained, the support vector machine (SVM) is used to select the prediction hypothesis. It is evident that the proposed HSIC-MKL is a two-stage kernel learning approach. Extensive experiments on real-world data sets from UCI benchmark repository validate the superiority of the proposed model in terms of prediction accuracy.

Keywords: Kernel method · Kernel fusion · Multiple kernel learning (MKL) Support vector machine (SVM) · Hilbert-Schmidt independence criterion (HSIC) Lasso

1 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]. These methods map data points from the input space to some 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 in place that can serve to guide the kernel leaning and selection. It is well known that selecting an appropriate kernel, thereby, an appropriate feature space is of great importance to the success of kernel methods [2]. To address this issue, recent years have witnessed the active research on learning effective kernels automatically from data. One popular technique for kernel learning and selection is multiple kernel learning (MKL) [3–5], which aims at learning a linear or nonlinear combination of a set of predefined kernels (base kernels) in order to identify a good target kernel for the applications. Compared with traditional kernel methods employing a fixed kernel, MKL exhibits its flexibility of

X. Geng and B.-H. Kang (Eds.): PRICAI 2018, LNAI 11012, pp. 246–255, 2018.
https://doi.org/10.1007/978-3-319-97304-3_19

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 all 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 only focus on the SVM-based MKL formulation. Specifically, we present a two-stage multiple kernel learning model based on the Hilbert-Schmidt independence criterion (HSIC), called HSIC-MKL. HSIC, which was initially introduced for measuring the statistical dependence between random variables or random processes [6], has been successfully applied in various machine learning problems [7], such as feature selection, clustering and subspace learning. The success is based on the fact that many existing learning tasks can be cast into problems of dependence maximization (or minimization). Motivated by this, in the first stage, we propose a 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 the global optimal solution can be computed efficiently by solving a Lasso optimization problem[1]. In the second stage, the SVM is used to select the prediction hypothesis, i.e., the SVM is trained to induce the final decision function to show classification results. It should be pointed out that the HSIC Lasso [8, 9] 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 to train a kernel classifier.

2 Multiple Kernel Learning

In this section, we briefly review the MKL. Suppose we are given a set of labeled training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ in a binary classification problem, where $\boldsymbol{x}_i \in \mathrm{X} \subset R^d$ is the input data and $y_i \in \{+1, -1\}$ is the corresponding class label. The goal of the SVM is 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

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

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

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.

To solve the SVM optimization problem, suppose α_i be the Lagrange multiplier corresponding to the ith inequality in (1), the dual problem of (1) is shown to

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

where $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}_j)$ is the kernel function which implicitly defines the feature map ϕ.

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 issue of MKL [3–5]. 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 pre-defined base kernels $\{k_m(\cdot, \cdot)\}_{m=1}^{M}$ with different kernel forms or different kernel parameters. The linear combination of the base kernels is given by $k = \sum_{m-1}^{M} \mu_m k_m$, 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 $\boldsymbol{\mu}$. With different constraints on $\boldsymbol{\mu}$, different MKL models can be obtained. For example, when $\boldsymbol{\mu} \in \Delta$ lies in a simplex, i.e.:

$$\Delta = \left\{ \boldsymbol{\mu} : \|\boldsymbol{\mu}\|_1 = \sum_{m=1}^{M} \mu_m = 1, \mu_m \ge 0 \right\} \tag{3}$$

we call it L_1-norm of kernel weights and the resulting model L_1-MKL [10]. Most MKL methods fall in this category. When

$$\Delta = \left\{ \boldsymbol{\mu} : \|\boldsymbol{\mu}\|_p \le 1, p > 1, \mu_m \ge 0 \right\} \tag{4}$$

we call it L_p-norm of kernel weights and the resulting model L_p-MKL [11].

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

$$\begin{aligned} \max \ & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \sum_{m=1}^{M} \mu_m k_m(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{s.t.} \ & \sum_{i=1}^{n} \alpha_i y_i = 0, \ \boldsymbol{\mu} \in \Delta, \\ & 0 \le \alpha_i \le C, \quad i = 1, \cdots, n. \end{aligned} \tag{5}$$

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

$$f(\boldsymbol{x}) = \mathrm{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \sum_{m=1}^{M} \mu_m k_m(\boldsymbol{x}_i, \boldsymbol{x}) + b\right) \tag{6}$$

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

3 MKL-HSIC

In this section, we detailedly discuss the two-stage MKL method (HSIC-MKL) for learning kernels in the form of 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 SVM is trained with the learned kernel.

We first introduce the notion of the HSIC. Let $\mathbf{e} = (1, \cdots, 1)^{\mathrm{T}} \in \mathrm{R}^n$ and $\mathbf{I} \in \mathrm{R}^{n \times n}$ be the identity matrix. Given the centering matrix $\mathbf{H} = \mathbf{I} - \mathbf{e}\mathbf{e}^{\mathrm{T}}/n \in \mathrm{R}^{n \times n}$, the centered kernel matrix associated with $\mathbf{K}$ is given by $\bar{\mathbf{K}} = \mathbf{HKH}$. Given two kernels k_1 and k_2, the HSIC between these two kernels is defined as

$$HSIC(\mathbf{K}_1, \mathbf{K}_2) = \frac{1}{n^2}\mathrm{tr}(\mathbf{K}_1\mathbf{H}\mathbf{K}_2\mathbf{H}) \tag{7}$$

Let $\bar{\mathbf{L}} = \mathbf{HLH}$ and $\bar{\mathbf{K}} = \mathbf{HKH}$, where $\mathbf{K}$ and $\mathbf{L}$ are the kernel matrix for input data an kernel matrix for output labels, respectively. We here propose using HSIC Lasso [8, 9] for estimating the combination coefficient $\boldsymbol{\mu}$:

$$\begin{aligned} &\min \ \frac{1}{2}\left\|\bar{\mathbf{L}} - \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m\right\|_{\mathrm{F}}^2 + \lambda\|\boldsymbol{\mu}\|_1 \\ &\text{s.t.} \quad \mu_1, \cdots, \mu_M \geq 0 \end{aligned} \tag{8}$$

where $||\cdot||_{\mathrm{F}}$ is the Frobenius norm and $\lambda > 0$ is the regularization parameter. In (8), 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 $\mu_m \rightarrow \mu_m / \sum_{m=1}^{M} \mu_m$.

Noting that $<\bar{\mathbf{K}}, \bar{\mathbf{L}}>_{\mathrm{F}} = <\bar{\mathbf{K}}, \mathbf{L}>_{\mathrm{F}} = <\mathbf{K}, \bar{\mathbf{L}}>_{\mathrm{F}} = \mathrm{tr}\mathbf{KHLH} = n^2 HSIC(\mathbf{K}, \mathbf{L})$, we can rewrite the first term of (8) as

$$\begin{aligned}
&\frac{1}{2}\left\|\bar{\mathbf{L}} - \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m\right\|_{\mathrm{F}}^2 \\
&= \frac{1}{2}\left\langle \bar{\mathbf{L}} - \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m, \bar{\mathbf{L}} - \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m \right\rangle_{\mathrm{F}} \\
&= \frac{1}{2}\langle \bar{\mathbf{L}}, \bar{\mathbf{L}} \rangle_{\mathrm{F}} - \left\langle \bar{\mathbf{L}}, \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m \right\rangle_{\mathrm{F}} + \frac{1}{2}\left\langle \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m, \sum_{m=1}^{M} \mu_m \bar{\mathbf{K}}_m \right\rangle_{\mathrm{F}} \\
&= \frac{1}{2}\langle \bar{\mathbf{L}}, \bar{\mathbf{L}} \rangle_{\mathrm{F}} - \sum_{m=1}^{M} \mu_m \langle \bar{\mathbf{L}}, \bar{\mathbf{K}}_m \rangle_{\mathrm{F}} + \frac{1}{2}\sum_{m=1}^{M}\sum_{o=1}^{M} \mu_m \mu_o \langle \bar{\mathbf{K}}_m, \bar{\mathbf{K}}_o \rangle_{\mathrm{F}} \\
&= \frac{n^2}{2} HSIC(\mathbf{L}, \mathbf{L}) - n^2 \sum_{m=1}^{M} \mu_m HSIC(\mathbf{L}, \mathbf{K}_m) + \frac{n^2}{2}\sum_{m=1}^{M}\sum_{o=1}^{M} \mu_m \mu_o HSIC(\mathbf{K}_m, \mathbf{K}_o)
\end{aligned} \tag{9}$$

In (9), 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 (9) 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 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 one word, HSIC Lasso tends to find non-redundant kernels with strong dependence on output $\mathbf{L}$, which is a preferable property in kernel learning.

To solve the HSIC Lasso problem in (9), many Lasso optimization techniques can be applied in practice, such as dual augmented Lagrangian (DAL) [12, 13], which has been successfully employed for high-dimensional feature selection [8, 9].

We sketch the overall procedure of the proposed HSIC-MKL in Algorithm 1, where the centered kernel matrix can be calculated by

$$\bar{\mathbf{K}}_{ij} = \mathbf{K}_{ij} - \frac{1}{n}\sum_{i=1}^{n} \mathbf{K}_{ij} - \frac{1}{n}\sum_{j=1}^{n} \mathbf{K}_{ij} + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{K}_{ij} \tag{10}$$

Algorithm 1. HSIC-MKL

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

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

2: Calculate the kernel matrix $\mathbf{L} = \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}$, where $\boldsymbol{y} = (y_1, \cdots, y_n)^{\mathrm{T}}$.

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

4: Obtain $\boldsymbol{\mu}$ by solving (8).

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

6: Combine the kernel matrices using the weight $\boldsymbol{\mu}$ and train an SVM classifier.

We analyze the computational complexity of Algorithm 1 with the O notation. Firstly, the computational complexity of calculating centered kernel matrices in Step 3 is $O(Mn^2)$. Secondly, the complexity of the quadratic programming solver in Step 4 is $O(TM^3)$ with T being the number of iterations in solving (8). Finally, note that empirically the SVM training complexity is $O(n^{2.3})$ [14], the computational complexity of Step 6 is $O(M + n^{2.3})$. Thus, the total computational complexity of our proposed HSIC-MKL is

$$O(Mn^2) + O(M^3) + O(M + n^{2.3}) = O(Mn^2 + M^3 + n^{2.3}) \tag{11}$$

It should be noted that we here suppose that multiple base kernels (kernel matrices) can be precomputed and loaded into memory before the HSIC-MKL training. Then, the computational cost of calculating the base kernels is ignored.

4 Experimental Evaluation

In this section, we perform extensive experiments on binary classification problems to evaluate the efficacy of the proposed HSIC-MKL approach. We compare HSIC-MKL with the following state-of-the-art kernel learning algorithms:

- AvgMKL: The average combination of multiple base kernels. It was reported that AvgMKL is competitive with many algorithms [3, 4].
- SimpleMKL [10]: An algorithm reformulates the mixed-norm regularization of MKL problem as the weighted 2-norm regularization, and L_1-norm is imposed on kernel weights.
- LpMKL [11]: 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.

- CKA-MKL [15]: The two-stage MKL with centered kernel alignment. The two-stage MKL first learns the optimal kernel weights according some criteria, and then applies the learned optimal kernel to train a kernel classifier.

For parameter settings, the regularization parameters C and λ are determined by 5-fold cross-validation on the training set. Specifically, 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. For our proposed HSIC-MKL approach, we perform grid-search over two dimensions, i.e., $C = \{10^{-2}, 10^{0}, \cdots, 10^{2}\}$ and $\lambda = \{10^{-2}, 10^{-1}, \cdots, 10^{2}\}$. In addition, for LpMKL, we examine $p = 2, 3, 4$ and report the best results. In the aspect of implementation, all the methods are implemented using MATLAB in the framework of SVM-KM toolbox[2]. Note that SimpleMKL has been implemented in the SimpleMKL software package[3], which needs the SVM-KM toolbox.

We select eight popular binary classification data sets, i.e., *Australian Credit Approval*, *Breast Cancer Wisconsin (Original)*, *Pima Indians Diabetes*, *German Credit Data*, *Heart*, *Ionosphere*, *Liver Disorders*, and *Sonar*, from the UCI machine learning repository [16]. For *Breast Cancer Wisconsin (Original)*, we directly eliminated the samples that contain missing attribute values. Table 1 provides the statistics of these data sets. It presents, for each data set, the short name of data set, the number of samples, the number of features, and the original name of data set.

Table 1. Statistics of the selected eight data sets from UCI

Data set	#Samples	#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

For each data set, we partition it into a training set and a test set by stratified sampling (by which the object generation follows the class prior probabilities): 50% of the data set serves as training set and the left 50% as 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 of the training data. Following the settings of previous For each data set, we partition it into a training set and a test set by stratified sampling (by which the object generation follows the class prior probabilities): 50% of the data set serves as

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

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

training set and the left 50% as 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 of the training data. Following the settings of previous MKL studies [10], we use the Gaussian kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 \Big/ 2\sigma^2)$ 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 ten different widths $\sigma \in \{2^{-3}, 2^{-2}, \cdots, 2^6\}$ for each individual dimension as well as all dimensions.
- Polynomial kernels with three different degrees $d \in \{1, 2, 3\}$ for each individual feature as well as all features.

All kernel matrices are normalized to unit trace and precomputed prior to running the algorithms.

Table 2. Classification accuracy comparison among different MKL algorithms on UCI data sets

Data set	Classification accuracy (%)				
	AvgMKL	SimpleMKL	LpMKL	CKA-MKL	HSIC-MKL
Australian	66.8 ± 4.5	85.1 ± 1.3	84.6 ± 1.7	**87.2 ± 0.4**	86.7 ± 1.3
Breast	95.4 ± 0.9	**96.6 ± 0.7**	96.1 ± 0.6	96.4 ± 0.9	96.6 ± 1.0
Diabetes	65.3 ± 1.8	75.9 ± 2.3	72.7 ± 2.4	75.3 ± 3.5	**77.1 ± 2.2**
German	69.6 ± 1.4	71.5 ± 2.6	**74.4 ± 1.5**	72.0 ± 1.2	72.4 ± 0.9
Heart	75.5 ± 5.3	83.1 ± 2.8	80.6 ± 3.6	82.1 ± 1.8	**83.3 ± 2.6**
Ionosphere	91.2 ± 1.8	93.5 ± 1.2	94.8 ± 2.1	93.7 ± 1.0	**95.5 ± 0.8**
Liver	57.4 ± 2.1	62.4 ± 4.3	69.3 ± 2.8	68.8 ± 1.6	**70.0 ± 2.9**
Sonar	59.0 ± 8.7	78.2 ± 3.5	**84.7 ± 3.3**	81.3 ± 2.8	81.8 ± 3.2

To get stable results, we independently repeat splitting each data set, and then run each algorithm on it for 20 times. The average classification accuracy and the standard deviations of each algorithm are reported in Table 2. The bold numbers denote the best performance of MKL methods on each data set. To conduct a rigorous comparison, the paired t-test [17] is performed. The paired t-test is used to analyze if the difference between two compared algorithms on one data set is significant or not. The p-value of the paired t-test represents the probability that two sets of compared results come from the distributions with an equal mean. A p-value of 0.05 is considered statistically significant. The win-tie-loss (W-T-L) summarizations based on the paired t-test are listed in Table 3, where HSIC-MKL and SimpleMKL, HSIC-MKL and LpMKL, and HSIC-MKL and CKA-MKL are compared, respectively. For two compared algorithms, assuming 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 have the same performance.

From Tables 2 and 3, we find that the proposed HSIC-MKL consistently achieves the overall best classification performance. Among the evaluated 8 data sets, SimpleMKL, LpMKL and CKA-MKL report 1, 2 and 1 best results, respectively, while our

Table 3. Significance test of classification results on UCI data sets

Data set	Win-tie-loss (W-T-L)		
	HSIC-MKL vs. SimpleMKL	HSIC-MKL vs. LpMKL	HSIC-MKL vs. CKA-MKL
Australian	W	W	T
Breast	T	T	T
Diabetes	W	W	W
German	W	L	T
Heart	T	W	W
Ionosphere	W	W	W
Liver	W	W	W
Sonar	W	L	T

HSIC-MKL reports 4 best results. From the viewpoint of significance test, we have the following observations. For HSIC-MKL, although it 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* and *Liver* data sets. Compared with SimpleMKL, HSIC-MKL significantly outperforms SimpleMKL on the *Australian*, *Diabetes*, *German*, *Ionosphere*, *Liver*, *Sonar* data sets, and yields the same performance on the rest of the data sets. 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 SimpleMKl, LpMKL and CKA-MKL.

5 Conclusion

We have presented an effective two-stage MKL algorithm based on the notion of HSIC. By 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 the global optimal solution can be computed efficiently by solving a Lasso optimization problem. Comprehensive experiments on a number of benchmark data sets demonstrate the promising results of our proposed algorithm. Future investigation will focus on the further validation of the use of the proposed algorithm on more real-world applications, such as computer vision, speech and signal processing, and natural language processing. Moreover, expending the proposed model to extreme learning machine and domain transfer learning, as well as investigating theoretical properties of the proposed algorithm are important issues to be investigated.

Acknowledgements. This work is supported in part by the National Natural Science Foundation of China (No. 61562003).

References

1. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York (2004)
2. Wang, T., Zhao, D., Tian, S.: An overview of kernel alignment and its applications. Artif. Intell. Rev. **43**(2), 179–192 (2015)
3. Gönen, M., Alpayın, E.: Multiple kernel learning algorithms. J. Mach. Learn. Res. **12**, 2211–2268 (2011)
4. Bucak, S.S., Jin, R., Jain, A.K.: Multiple kernel learning for visual object recognition: a review. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1354–1369 (2014)
5. Gu, Y., Chanussot, J., Jia, X., Benediktsson, J.A.: Multiple kernel learning for hyperspectral image classification: a review. IEEE Trans. Geosci. Remote Sens. **55**(11), 6547–6565 (2017)
6. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with Hilbert-Schmidt norms. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 63–77. Springer, Heidelberg (2005). https://doi.org/10.1007/11564089_7
7. Wang, T., Li, W.: Kernel learning and optimization with Hilbert-Schmidt independence criterion. Int. J. Mach. Learn. Cybern. 1–11 (2017). https://doi.org/10.1007/s13042-017-0675-7
8. Yamada, M., Kimura, A., Naya, F., Sawada, H.: 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, pp. 1827–1833 (2013)
9. Yamada, M., Jitkrittum, W., Sigal, L., Xing, E.P., Sugiyama, M.: High-dimensional feature selection by feature-wise kernelized Lasso. Neural Comput. **26**(1), 185–207 (2014)
10. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: SimpleMKL. J. Mach. Learn. Res. **9**, 2491–2521 (2008)
11. Kloft, M., Brefeld, U., Sonnenburg, S., Zien, A.: l_p-norm multiple kernel learning. J. Mach. Learn. Res. **12**, 953–997 (2011)
12. Tomioka, R., Sugiyama, M.: Dual-augmented Lagrangian method for efficient sparse reconstruction. IEEE Sig. Process. Lett. **16**(12), 1067–1070 (2009)
13. Tomioka, R., Sugiyama, M.: Super-linear convergence of dual augmented Lagrangian algorithm for sparsity regularized estimation. J. Mach. Learn. Res. **12**, 1537–1586 (2011)
14. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods: Support Vector Learning, pp. 185–208 (1999)
15. Cortes, C., Mohri, M., Rostamizadeh, A.: Algorithms for learning kernels based on centered alignment. J. Mach. Learn. Res. **13**, 795–828 (2012)
16. Lichman, M.: UCI machine learning repository. University of California, School of Information and Computer Science, Irvine (2013). http://archive.ics.uci.edu/ml/
17. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)