Experimental Design With Multiple Kernels

Hanmo Wang*†, Liang Du*‡, Lei Shi*†, Peng Zhou*†, Yuhua Qian‡, Yi-Dong Shen*
*State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
†Graduate University, Chinese Academy of Sciences, Beijing 100049, China
{wanghm,duliang,shilei,zhoup,ydshen}@ios.ac.cn
‡Shanxi University, Shanxi 030006 ,China

***Abstract*—In classification tasks, labeled data is a necessity but sometimes difficult or expensive to obtain. On the contrary, unlabeled data is usually abundant. Recently, different active learning algorithms are proposed to alleviate this issue by selecting the most informative data points. One family of active learning methods comes from Optimum Experimental Design (OED) in statistics. Instead of selecting data points one by one iteratively, these approaches select data in a one-shot manner, that is, a fixed-sized subset is selected from the unlabeled dataset for manually labeling. These methods usually uses kernels to represent pair-wise similarities between different data points. However, in practice, choosing the optimal kernel type (e.g. Gaussian kernel) and kernel parameter (e.g. kernel width) can be tricky, and sometimes merging all the kernel candidates to form a consensus kernel can outperform methods that select the best single kernel via data-driven approaches (e.g. Cross Validation). The most common way of combining multiple kernels is to assign different weight to different candidate kernels to form a consensus kernel. This is the globalised approach. However, in practice, different pair-wise similarities in the same candidate kernel may not have the same contribution in the consensus kernel, so it is sometimes more appropriate to assign weight to each datum-kernel pair so that different data point have different weight in the same candidate kernel. This so-called localized approach performs better than the globalised one when the data have underlined local structure. In this paper, we propose two multiple kernel OED-based active learning approaches, namely globalised and localized multiple kernel active learning (GMKAL and LMKAL). We also show that our localized and globalised MK method have better or comparable results than the single kernel counterparts. Our method do not introduce addtional parameters and experiments on 6 different datasets demonstrate the effectiveness of our method.**

I. Introduction

Classification is one of the fundamental tasks in data mining and machine learning. One of the important perquisite of classification is sufficient labeled data. While unlabeled data is abundant (one can simply crawl unlabeled data from the Internet), the labeling process usually needs human experts to annotate, thus requiring considerable amount of time and effort. Active learning is the methodology to choose the most informative data points to label so that the classifier learned on the labeled data can have good generalization performance on unseen data. One of the famous algorithms of active learning is query-by-committee [1], which trains different classifiers on the labeled data and iteratively select the single unlabeled data point on which the classifiers disagree most. For probabilistic models, uncertainty sampling [2] is suitable because it regards the most informative data points as the one that is hard to label.

Another line of work in active learning is to develop algorithms based on Optimum Experimental Design (OED) [3] [4]. OED address similar problems as active learning in statistics. The difference between OED and active learning is that OED cares more about the variance of the data while active learning cares more about classification quality. However, this difference brings inspiration, an active learning algorithms via transductive experimental design (TED) [4] is proposed to minimize the variance of unlabeled data. This algorithm can be reinterpreted as minimizing the linear reconstruction error with regularizations. After that, the convex version of TED for text classification is also studied [3], denoted as CTED. Recently, a robust version of CTED is introduced [5] by replacing the loss function and there also exists an accelerated version [6]. Unlike query-by-committee, OED-based active learning methods selects all the needed data instead of selecting data points iteratively, which runs faster because they do not need to retrain classifiers.

OED-based active learning methods usually require kernels to select the best subset of the unlabeled data. In practice, what type of kernel and what kernel parameter to select can be a problem. Assume that we are given a set of candidate kernels, one of the solution is to use data-driven approach such as Cross Validation (CV) to select the best kernel type and kernel parameter. However, in active learning, Cross Validation might have some issues. Let's take the standard k-fold CV as an instance. At the very beginning of the active learning process, there is no label at all, which makes k-fold CV impossible to conduct because it requires a *labeled* training set so that it can split the training set int two k parts, train on the $(k-1)$ fold and *test* on the last fold. The trick lies in the *testing* part of k-fold CV. When testing, k-fold cross validation actually has to known all the *labels* of the training (unlabeled) set, but this situation is unrealistic (initially we have no label at all) and prevents active learning algorithms with parameters to be used in practice. Fortunately, when it comes to choose kernel types and kernel parameters, the extensively studied topic called Multiple Kernel Learning can be helpful.

In classification or clustering tasks, when facing the kernel selection problem, Multiple Kernel Learning (MKL) [7] [8] can be helpful to decide which kernel and what kernel combinations are suitable. In stead of selecting the best kernel from the candidate kernel sets, MKL algorithms uses linear or non-linear combinations of all the kernel candidates to jointly learn a optimal consensus kernel in the training process. Similarly, in active learning, we do not know in advance which kernel can best represent the unlabeled dataset to guide the data selection process and MKL algorithms can be adopted in active learning to merging all the candidate kernels so that the learned

consensus kernel has better or comparable performance in data selection with the best single kernel.

Usually, in MKL, the consensus kernel is defined as the weighted linear combination of the candidate kernels, denoted as *Globalised* MKL. Since the weight can be viewed as an indicator of pair-wise similarity over unlabeled data points in different candidate kernels, *Globalised* MKL assumes that this indicator remains unchanged in the unlabeled dataset with respect to the same candidate kernel. However, the indicator of a candidate kernel can change with samples for several reasons. First, the pair-wise similarity of unlabeled dataset is not necessarily equally effective across all samples. Second, some input features of the unlabeled dataset could be contaminated by noise in practice, thus making a candidate kernel calculated with these input features unreliable for this specific sample. All these cases can turn a candidate kernel useful for one sample but noisy to another.

In this paper, we develop simple and effective algorithms for multiple kernel active learning (MKAL). We first show that the most intuitive way to represent the consensus kernel as the weighed combination of all candidate kernels can be helpful. This intuitive solution can be solved by iteratively optimizing reconstruction matrix and kernel weights. The first algorithm is the globalised version because every pair-wise similarity in the same kernel contribute the same in the consensus kernel. In the second algorithm, the localized version, we assign weight to each datum-kernel pair, so that each pair-wise distance can have different weight in the consensus kernel.

Our contributions are as follows:

- We propose localized and globalised multiple kernel active learning algorithms based on Kernel CTED [3]. To the best of our knowledge, we are the first to introduce multiple kernel learning to active learning.
- We formulate our algorithms without introducing additional parameters and solve the formulation using coordinate descent methods. No additional parameters make our algorithm more easily to tune and more easily to apply in practice.
- Experiments on 6 different datasets demonstrate that our algorithms has similar or better performance than its single kernel counterpart.

Notations. The $l_{2,1}$ norm of a matrix $A \in R^{n\times m}$ is defined as $||A||_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{m} A_{ij}^2}$. The (pseudo) $l_{2,0}$ norm of a matrix is defined as the number of non-zero rows of the matrix. We also denote the i-th row and i-th column of a matrix $\mathbf{A}$ as $\mathbf{a}^i$ and $\mathbf{a}_i$, respectively. And $||\cdot||_F$ denotes the Frobenius Norm. For a matrix $\mathbf{A}$, the entry at the i-th row and the j-th column of $\mathbf{A}$ is denoted as A_{ij}.

II. Related Work

A. Active Learning

A number of active learning strategies are proposed to select the most informative data points for training a classifier which has good generalization performance. One of the common strategies is uncertainty sampling [2] [9]. It aims to select the samples about which the current model is least certain. For probabilistic models, uncertainty sampling is usually straightforward, criteria such as entropy [2] are used to query examples. For the non-probabilistic models such as Support Vector Machines, the data point which is closest to the decision boundary is considered uncertain and selected [9]. Query-by-committee [1] is another typical active learning strategy. It trains a group of classifiers and selects the unlabeled example about which the classifiers disagree the most.

Active learning is also known as Optimum Experimental Design (OED) in statistics. Classical optimal experimental design criteria include A-optimal design, D-optimal design, and I-optimal design [10]. Some active learning algorithms are inspired by OED. For example, TED [4] selects those points that minimize the average predictive variance over one predefined test set, similar to I-optimal Design. Along this line of work, many active learning algorithms are proposed according to different measures and purposes. Convex TED (CTED) [3] is proposed to obtain global optimal solution and [5] deals with potential noise in the unlabeled dataset. Additionally, spatial structure such as neighborhood reconstruction [11] [12] and manifold [13] are considered. There is also an accelerated version [6].

B. Multiple Kernel Learning

Kernel methods [14] [15] serves as an extensively studied topic in reaserch in the past decades. In practice, it is often hard to choose the right kernel type and kernel parameters, and it is of vital importance to the success of kernel methods to learn an optimal kernel. MKL algorithms [7] [8] provides an efficient way to learn an consensus optimal kernel. Additionally it serves as a framework to combine multiple data sources. The existing research work on MKL has made significant contributions in speeding up computation [16] and improving classification performance [7]. Given a predefined set of candidate kernels, a straight forward way to merge the kernels is to assign weight to different candidate kernels. However, this can be suboptimal in practice because each input instance may have different importance under the same similarity measure (kernel) for the task at hand. So-called localized MKL algorithms are proposed to alleviate the situation [17] [18]. MKL algorithms can also be categorized into classification and clustering. For clustering, multiple kernel k-means [19] [20] and spectral clustering [21] [22] are widely studied. As for classification, multiple kernel Support Vector Machines [23] and online algorithms [24] are proposed.

III. Background

In this section, we provide some background knowledge for OED-based active learning. Additionally, we formalize the multiple kernel active learning problem studied in this paper.

A. Background

Transductive Experimental Design (TED) is proposed to select a subset of the unlabeled dataset $\mathbf{X}$ to have a low reconstruction error [4]. In other words, this method chooses a subset $\mathbf{V}$ from the whole dataset $\mathbf{X}$ so that the linear reconstruction error is minimized, i.e.

$$\begin{aligned} \min_{\mathbf{A},\mathbf{V}} \quad & \sum_{i=1}^{n} (||\mathbf{x}_i - \mathbf{V}\mathbf{a}_i||_2^2 + \lambda ||\mathbf{a}_i||_2^2) \\ \text{s.t.} \quad & \mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n], \mathbf{V} \subset \mathbf{X}, |\mathbf{V}| = p \end{aligned} \tag{1}$$

TED chooses the most representative sample because the other unlabeled data points can be linearly reconstructed by the sample with low error. However, Eq. (1) has some issues in terms of optimization because it is a combinatorial optimization problem which is NP-hard. Therefore, a greedy algorithm is used to slove Eq. (1).

Eq. (1) can also be written with the help of the $l_{2,0}$ norm

$$\begin{aligned} \min_{\mathbf{A}} \quad & \sum_{i=1}^{n}(||\mathbf{x}_i - \mathbf{X}\mathbf{a}_i||_2^2 + \lambda||\mathbf{a}_i||_2^2) \\ \text{s.t.} \quad & \mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n], \quad ||\mathbf{A}||_{2,0} = p \end{aligned} \tag{2}$$

The constant p in Eq. (2) makes it difficult to analyze and we can relax it by formulating it as

$$\min_{\mathbf{A}} \quad ||\mathbf{X} - \mathbf{X}\mathbf{A}||_F^2 + \lambda||\mathbf{A}||_{2,0} \tag{3}$$

where $\mathbf{A}$ is a n by n matrix, indicating the reconstruction coefficients.

Later, a convex version of TED (CTED) is proposed for text classification [3]

$$\min_{\mathbf{A}} \quad ||\mathbf{X} - \mathbf{X}\mathbf{A}||_F^2 + \lambda||\mathbf{A}||_{2,1} \tag{4}$$

CTED relaxes the $l_{2,0}$ norm with its convex hull $l_{2,1}$ norm so that it is better than TED in the view of optimization because Eq. (4) is jointly convex with respect to the $\mathbf{A}$ and the global optimal solution can be reached. The square error (Frobenius Norm) of Eq. (4) is known to be sensitive to outliers. So the robust version of CTED is proposed to make the model robust to outliers [5].

$$\min_{\mathbf{A}} \quad ||(\mathbf{X} - \mathbf{X}\mathbf{A})^T||_{2,1} + \lambda||\mathbf{A}||_{2,1} \tag{5}$$

CTED (Eq. (4)) can be easily extended to a kernel version by replacing $\mathbf{x}_i$ with $\phi(\mathbf{x}_i)$, i.e.

$$\min_{\mathbf{A}} \quad \text{tr}(\mathbf{K} - 2\mathbf{K}\mathbf{A} + \mathbf{A}^T\mathbf{K}\mathbf{A}) + \lambda||\mathbf{A}||_{2,1} \tag{6}$$

where the kernel Gram matrix $\mathbf{K}$ is defined as $\mathbf{K}_{ij} =< \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) >$.

B. Multiple Kernel Active Learning Problem

The previous methods can be categorized into single kernel active learning, where the kernel is predefined or selected from a candidate sets by Cross Validation. In this section, we introduce the problem setting of our Multiple Kernel Active Learning (MKAL) methods as follows.

We have the unlabeled dataset $\mathbf{X} \in \mathcal{R}^{d\times n}$ consists of n unlabeled data points with d dimensions and each column of $\mathbf{X}$ is a data point, i.e. $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$. Here we abuse notations and use $\mathbf{X}$ to denote both the unlabeled dataset and the data matrix. Besides that, we also have m different kernels, namely $\{\mathcal{K}_i(\cdot,\cdot)\}_{i=1}^m$. Each kernel $\mathcal{K}_t(\cdot,\cdot)$ is associated with unique mapping function $\phi_t(\cdot)$ and according to definition of kernels, we have $\mathcal{K}_t(\mathbf{x}_i, \mathbf{x}_j) =< \phi_t(\mathbf{x}_i), \phi_t(\mathbf{x}_j) >$

The goal of MKAL algorithms is to choose a subset $\mathbf{L}$ of the dataset $\mathbf{X}$ with $|\mathbf{L}| = p$ so that the classifier trained on $\mathbf{L}$ have good classification performance on unseen data, where p is a predefined subset size. By combining different kernels, MKAL methods should have comparable classification results with the single kernel counterparts on the testing data.

IV. SOLVING CTED

Let's take a detour to solve Eq. (6) first because it is a subroutine of our globalised and localized algorithms. For clarity, we write the objective function of Eq. (6) as

$$\min_{\mathbf{A}} \quad f(\mathbf{A}) = \text{tr}(\mathbf{K} - 2\mathbf{K}\mathbf{A} + \mathbf{A}^T\mathbf{K}\mathbf{A}) + ||\mathbf{A}||_{2,1} \tag{7}$$

Our goal is to minimize function $f(\mathbf{A})$, and the most intuitive way is to take derivative w.r.t. $\mathbf{A}$ and set the derivative to zero. However, the $l_{2,1}$ norm can be tricky because its derivative does not exist when matrix $\mathbf{A}$ has a row of all zeros. Adding an offset ϵ to each row of $\mathbf{A}$ when it goes to zero, we get

$$-\mathbf{K} + \mathbf{K}\mathbf{A} + \lambda\mathbf{D}\mathbf{A} = 0 \tag{8}$$

where $\mathbf{D}$ is a diagonal matrix and

$$\mathbf{D}_{ii} = \frac{1}{2\sqrt{||\mathbf{a}^i||_2^2 + \epsilon}} \tag{9}$$

where $\mathbf{a}^i$ is the i-th row of matrix $\mathbf{A}$.

With Eq. (8), we can solve it by alternating optimizing $\mathbf{A}$ and $\mathbf{D}$ iteratively: When $\mathbf{D}$ is fixed, we can directly calculate matrix $\mathbf{A}$ according to Eq. (8) as

$$\mathbf{A} = (\mathbf{K} + \lambda\mathbf{D})^{-1}\mathbf{K} \tag{10}$$

And when $\mathbf{A}$ is fixed, we can calculate $\mathbf{D}$ as in Eq. (9). When the algorithm converges, the instances are sorted decreasingly by the row absolute sum of $\mathbf{A}$, and the largest p instances will be selected.

Algorithm 1 illustrate the details of solving Eq. (6).

Algorithm 1 The algorithm of Kernel CTED

Input: kernel matrix $\mathbf{K}$, size of selection p
Initialize reconstruction matrix $\mathbf{A}$ randomly
repeat
Update $\mathbf{D}$ by Eq. (9);
Update the reconstruction matrix $\mathbf{A}$ by Eq. (10)
until Converges
Sort all rows of $\mathbf{A}$ decreasingly according to the row absolute sum and let $\mathbf{L}$ be indexes of the instances with the p largest sum
Output: selected indexes $\mathbf{L}$

V. GLOBALISED MULTIPLE KERNEL ACTIVE LEARNING

In this section, we introduce our globalised multiple kernel active learning (GMKAL) algorithm. This globalised method is intuitive in that it assign different weight to different candidate kernels and the consensus kernel is a weighted combination of all the candidate kernels. In this way, the kernels that have good performance in data selection will have larger weight, and vise versa.

A. Motivation and Formulation

One of the central problems with kernel methods in general is that it is often unclear which kernel is the most suitable for a particular task. In this section, we extend kernel CTED to automatically learn an appropriate kernel from the convex linear combination of several predefined kernel matrices within the multiple kernel learning framework [25].

Suppose there are altogether m different kernel functions $\{\mathcal{K}_i(\cdot,\cdot)\}_{i=1}^m$ available for the active learning task at hand. Accordingly, there are m different associated feature spaces denoted as $\{\mathcal{H}\}_i^m$. To combine these kernels and also ensure that the resulted kernel still satisfies Mercer condition, we consider a nonnegative combination of these feature maps $\phi_w(\cdot)$, that is,

$$\phi_w(\mathbf{x}) = \sum_{i=1}^{m} w_i \phi_i(\mathbf{x}) \quad \text{with } w_i \geq 0. \tag{11}$$

Unfortunately, as these implicit mappings do not necessarily have the same dimensionality, such a linear combination may be unrealistic. Hence, we construct an augmented Hilbert space $\tilde{\mathcal{H}} = \oplus_{i=1}^m \mathcal{H}^i$ by concatenating all feature spaces $\phi_w(\mathbf{x}) = [w_1\phi_1(\mathbf{x})^T w_2\phi_2(\mathbf{x})^T \dots w_m\phi_m(\mathbf{x})^T]^T$ with different weight $w_i(w_i \geq 0)$, or equivalently the importance factor for kernel function $\mathcal{K}_i(\cdot,\cdot)$. So that the consensus kernel $\mathcal{K}_w(\cdot,\cdot)$ can be represented as

$$\mathcal{K}_w(\mathbf{x},\mathbf{z}) =< \phi_w(\mathbf{x}), \phi_w(\mathbf{z}) >= \sum_{i=1}^{m} w_i^2 \mathcal{K}_i(\mathbf{x},\mathbf{z}). \tag{12}$$

It is known that the convex combination, with $\mathbf{w}(w_i \geq 0)$, of the positive semi-definite kernel matrices $\{\mathbf{K}_i\}_{i=1}^m$ is still a positive semi-definite kernel matrix. By replacing the single kernel in Eq. (4) with the combined kernel, we propose a new Globalized Multiple Kernel Active Learning (GMKAL) method by solving

$$\begin{aligned} \min_{\mathbf{A},\mathbf{w}} \quad & \mathrm{tr}(\mathbf{K}_\mathbf{w} - 2\mathbf{K}_\mathbf{w}\mathbf{A} + \mathbf{A}^T\mathbf{K}_\mathbf{w}\mathbf{A}) + \lambda||\mathbf{A}||_{2,1} \\ \text{s.t.} \quad & \mathbf{K}_\mathbf{w} = \sum_{i=1}^{m} w_i^2 \mathbf{K}_i, \sum_{i=1}^{m} w_i = 1, w_i \geq 0 \end{aligned} \tag{13}$$

where $(\mathbf{K}_i)_{ab} = \mathcal{K}_i(\mathbf{x}_a, \mathbf{x}_b)$ is the kernel Gram matrix of the i-th predefined kernel function over the unlabeled dataset $\mathbf{X}$, and $(\mathbf{K}_\mathbf{w})_{ab} = \mathcal{K}_w(\mathbf{x}_a, \mathbf{x}_b)$ is the kernel matrix of the consensus kernel function $\mathcal{K}_w(\cdot,\cdot)$.

B. Algorithms

The optimization problem in Eq. (13) is convex w.r.t. $\mathbf{A}$ and $\mathbf{w}$ respectively. In the following, we introduce an iterative algorithm based on block coordinate descent to solve it. We separately update the value of $\mathbf{w}$ and $\mathbf{A}$, while holding the other variable as constant. Thus, a local minima can be expected by solving a sequence of convex optimization problems.

1) Optimizing w.r.t. $\mathbf{A}$ *when* $\mathbf{w}$ *is fixed:* When $\mathbf{w}$ is fixed, we can directly calculate $\mathbf{K}_\mathbf{w}$ as $\mathbf{K}_\mathbf{w} = \sum_{i=1}^m w_i^2 \mathbf{K}_i$, and the optimization problem becomes Eq. (6), and can be solved by Algorithm 1 with $\mathbf{K}_\mathbf{w}$ as the input kernel matrix.

2) Optimizing w.r.t. $\mathbf{w}$ *when* $\mathbf{A}$ *is fixed:* the optimization of Eq. (13) with respect to $\mathbf{w}$ can be simplified as solving the following problem

$$\min_{\mathbf{w}} \quad \sum_{i=1}^{m} w_i^2 e_i, \quad \text{s.t.} \quad \sum_{i=1}^{m} w_i = 1, w_i \geq 0. \tag{14}$$

where

$$e_i = \mathrm{tr}(\mathbf{K}_i(\mathbf{I} - 2\mathbf{A} + \mathbf{A}\mathbf{A}^T)) \tag{15}$$

The above optimization problem falls into the category of quadratic programming (QP) and can be solved by sophisticated QP tool boxes. However, since Eq. (14) has only quadratic terms of the same variable (without terms like $w_i w_j$), we can utilize this structure to form an analytic solution:

First we write the Lagrange function of Eq. (14) as

$$\mathcal{J}(\mathbf{w}) = \sum_{i=1}^{m} w_i^2 e_i + \lambda(1 - \sum_{i=1}^{m} w_i). \tag{16}$$

By combining the KKT condition $\frac{\partial \mathcal{J}(\mathbf{w})}{\partial w_i} = 0$ and the constraint $\sum_{i=1}^m w_i = 1$, the optimal solution of $\mathbf{w}$ can be obtained by

$$w_i = \frac{\frac{1}{e_i}}{\sum_{j=1}^m \frac{1}{e_j}}, \quad i = 1, 2, ..., m. \tag{17}$$

Algorithm 2 illustrates the steps of GMKAL algorithm in detail.

Algorithm 2 The algorithm of GMKAL

Input: A set of kernel matrices $\{\mathbf{K}_i\}_{i=1}^m$, the number of selected data points p
Initialize the kernel weight $w_i = 1/m$ for each kernel;
repeat
Update the estimated kernel $\mathbf{K}_\mathbf{w}$ by Eq. (12)
Update the reconstruction matrix $\mathbf{A}$ by Algorithm 1 with $\mathbf{K} = \mathbf{K}_w$
Calculate $\mathbf{e}$ by Eq. (15)
Update the kernel weight $\mathbf{w}$ by Eq. (17)
until Converges
Sort all rows of $\mathbf{A}$ decreasingly according to the row absolute sum and let $\mathbf{L}$ be indexes of the instances with the p largest sum
Output: selected indexes $\mathbf{L}$

VI. Localized Multiple Kernel Active Learning

As mentioned before, sometimes giving one weight to one candidate kernel may not be the optimal solution, because some data points may be corrupted by noise in practice and different data points in the unlabeled dataset may contribute differently to the consensus kernel. In this section, we propose a localized multiple kernel active learning algorithm (LMKAL) by assigning weights to each pair of unlabeled data point and candidate kernel.

A. Motivation and Formulation

In the globalised case, we assign a fixed weight to a kernel over the whole input space. However, assigning different weights to a kernel in different regions of the input space may produce a better performance. Especially, if the data has underlying local structure, different similarity measures may be suited in different regions. Additionally, some entries in some kernel datasets may be contaminated by noise, and assigning different weights to different regions of the same kernel may alleviate the problem.

In the localized setting, we assign weight to each datum-kernel pair, that is, we assign weight Z_{ij} for the i-th data point

and the j-th candidate kernel. Similar with the global case, we concatenating all the feature space as

$$\phi_z(\mathbf{x}_i) = [Z_{i1}\phi_1(\mathbf{x}_i)^T Z_{i2}\phi_2(\mathbf{x}_i)^T ... Z_{im}\phi_m(\mathbf{x}_i)^T]^T$$

According to the definition of kernel, the consensus kernel function $\mathcal{K}_z$ becomes

$$\begin{aligned}\mathcal{K}_z(\mathbf{x}_i, \mathbf{x}_j) &=< \phi_z(\mathbf{x}_i), \phi_z(\mathbf{x}_j) > \\ &= \sum_{t=1}^{m} Z_{it} Z_{jt} \mathcal{K}_t(\mathbf{x}_i, \mathbf{x}_j)\end{aligned} \tag{18}$$

Similar with the globalised case, we define consensus kernel matrix over the unlabeled dataset $\mathbf{X}$ as

$$(\mathbf{K_z})_{ij} = \mathcal{K}_z(\mathbf{x}_i, \mathbf{x}_j) = \sum_{t=1}^{m} Z_{it} Z_{jt} (\mathbf{K}_t)_{ij} \tag{19}$$

In the following, we prove that the function defined in Eq. (18) is indeed a kernel function. And then we propose our localized MKAL algorithm.

Theorem 1. *The function $\mathcal{K}_z(\cdot,\cdot)$ defined in Eq.* (19) *is a positive semi-definite kernel function.*

Proof: To prove that $\mathcal{K}_z(\cdot,\cdot)$ is a positive semi-definite kernel function, we introduce the following lemma:

Lemma 1. *Let $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$ be a symmetric function, the necessary and sufficient condition that $\mathcal{K}(\cdot,\cdot)$ is a positive semi-definite kernel function is that the Leading Principle Submatrix $\mathbf{K} = [\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)]_{1:m\times 1:m}$ of the kernel Gram matrix $[\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)]_{n\times n}$ over arbitrary n samples $\{\mathbf{x}_i\}_{i=1}^n$ ($\mathbf{x}_i \in \mathcal{X}$) is a positive semi-definite matrix for all positive integer n and all $m \in \{1, 2, ..., n\}$.*

According to Lemma 1, we just need to prove that for any $\mathbf{x}_1, ..., \mathbf{x}_n$, the Gram matrix $\mathbf{K_z}$ is positive semi-definite. Let $\mathbf{K_z}$ be the Gram matrix of consensus kernel $\mathcal{K}_z(\cdot,\cdot)$ and $\mathbf{K}_t$ be the Gram Matrix of the t-th candidate kernel $\mathcal{K}_t(\cdot,\cdot)$, we just need to prove that $(\mathbf{z}_t\mathbf{z}_t^T) \circ \mathbf{K}_t$ is positive semi-definite, where $\mathbf{z}_t$ is the t-th column of $\mathbf{Z}$ and $\circ$ is the element-wise dot product operator.

Since $\mathcal{K}_t(\cdot,\cdot)$ is positive semi-definite kernel function, according to Lemma 1, $\mathbf{K}_t$ is positive semi-definite. Thus all the leading principal minors of $\mathbf{K}_t$ are all positive. Now consider the j-th leading principal minor of $\mathbf{K_z}$:

$$\begin{aligned}&\begin{vmatrix} \mathcal{K}_z(\mathbf{x}_1,\mathbf{x}_1) & \cdots & \mathcal{K}_z(\mathbf{x}_1,\mathbf{x}_j) \\ \vdots & \vdots & \vdots \\ \mathcal{K}_z(\mathbf{x}_j,\mathbf{x}_1) & \cdots & \mathcal{K}_z(\mathbf{x}_j,\mathbf{x}_j) \end{vmatrix} \\ =&\left| \begin{pmatrix} Z_{1t}Z_{1t} & \cdot & Z_{1t}Z_{jt} \\ \vdots & \vdots & \vdots \\ Z_{jt}Z_{1t} & \cdot & Z_{jt}Z_{jt} \end{pmatrix} \circ \begin{pmatrix} (\mathbf{K}_t)_{11} & \cdot & (\mathbf{K}_t)_{1j} \\ \vdots & \vdots & \vdots \\ (\mathbf{K}_t)_{j1} & \cdot & (\mathbf{K}_t)_{jj} \end{pmatrix} \right| \\ =&Z_{1t}^2 Z_{2t}^2 \cdots Z_{jt}^2 \begin{vmatrix} \mathcal{K}_t(\mathbf{x}_1,\mathbf{x}_1) & \cdots & \mathcal{K}_t(\mathbf{x}_1,\mathbf{x}_j) \\ \vdots & \vdots & \vdots \\ \mathcal{K}_t(\mathbf{x}_j,\mathbf{x}_1) & \cdots & \mathcal{K}_t(\mathbf{x}_j,\mathbf{x}_j) \end{vmatrix}\end{aligned} \tag{20}$$

where $|\cdot|$ is determinant of a matrix.

Since $\mathbf{K}_t$ is positive semi-definite, we have

$$\begin{vmatrix} \mathcal{K}_t(\mathbf{x}_1,\mathbf{x}_1) & \cdots & \mathcal{K}_t(\mathbf{x}_1,\mathbf{x}_j) \\ \vdots & \vdots & \vdots \\ \mathcal{K}_t(\mathbf{x}_j,\mathbf{x}_1) & \cdots & \mathcal{K}_t(\mathbf{x}_j,\mathbf{x}_j) \end{vmatrix} \geq 0 \tag{21}$$

Thus, we obtain:

$$\begin{vmatrix} \mathcal{K}_z(\mathbf{x}_1,\mathbf{x}_1) & \cdots & \mathcal{K}_z(\mathbf{x}_1,\mathbf{x}_j) \\ \vdots & \vdots & \vdots \\ \mathcal{K}_z(\mathbf{x}_j,\mathbf{x}_1) & \cdots & \mathcal{K}_z(\mathbf{x}_j,\mathbf{x}_j) \end{vmatrix} \geq 0 \tag{22}$$

Eq. (22) holds for any $1 \leq j \leq n$, thus $\mathbf{K_z}$ is positive semi-definite and $\mathcal{K}_z(\cdot,\cdot)$ is a positive semi-definite kernel function. ■

Now that we have proved that the function defined in Eq. (18) is indeed a kernel function and kernel matrix $\mathbf{K_z}$ is positive semi-definite, substitute Eq. (19) into Eq. (6), we have our formulation:

$$\begin{aligned}\min_{\mathbf{A},\mathbf{Z}} \quad & \mathrm{tr}(\mathbf{K_z} - 2\mathbf{K_z}\mathbf{A} + \mathbf{A}^T\mathbf{K_z}\mathbf{A}) + \lambda||\mathbf{A}||_{2,1} \\ \text{s.t.} \quad & (\mathbf{K_z})_{ij} = \sum_{t=1}^{m} Z_{it} Z_{jt} (\mathbf{K}_t)_{ij}, \\ & \sum_{j=1}^{m} Z_{ij} = 1, Z_{ij} \geq 0, i = 1, 2, ..., n\end{aligned} \tag{23}$$

Now we illustrate intuitive explanation of our objective Eq. (23) in the perspective of different data samples and kernel matrices. In the unlabeled dataset $\mathbf{X}$, we do not have bias towards any data samples, because each row of matrix $\mathbf{Z}$ sums to 1 ($\sum_{j=1}^{m} Z_{ij} = 1$). In other words, each data sample has the same importance to the consensus kernel, but when the same sample faces different kernels, we assign different non-negative weight to the pair of i-th sample and t-th kernel (i.e. Z_{it}). So each data sample have different importance to the same candidate kernel. The entry in the i-th row and the j-th column consensus kernel matrix $\mathbf{K_z}$ demonstrates how the pair of i-th and j-th sample contributes to the kernel matrix $\mathbf{K_z}$.

B. Algorithms

Similar to the globalised case, the optimization problem in Eq. (23) is convex w.r.t. $\mathbf{A}$ and $\mathbf{Z}$ respectively. In the following, we also introduce an iterative algorithm based on block coordinate descent to solve it. We separately update the value of $\mathbf{Z}$ and $\mathbf{A}$, while holding the other variables as constant. By solving a sequence of convex optimization problems, we obtain a local minimum solution.

1) Optimizing w.r.t. $\mathbf{Z}$ when $\mathbf{A}$ is fixed: When $\mathbf{A}$ is fixed, the original optimization problem can be written in a more condensed form:

$$\begin{aligned}\min_{\mathbf{Z}} \quad & \mathrm{tr}(\mathbf{K_z} - 2\mathbf{K_z}\mathbf{A} + \mathbf{A}^T\mathbf{K_z}\mathbf{A}) \\ \text{s.t.} \quad & \mathbf{K_z} = \sum_{i=1}^{m} (\mathbf{z}_i\mathbf{z}_i^T) \circ \mathbf{K}_i, \quad \mathbf{Z}\mathbf{1}_m = \mathbf{1}_n\end{aligned} \tag{24}$$

where $\mathbf{z}_i$ is the i-th column of matrix $\mathbf{Z}$.

We can also eliminate $\mathbf{K_z}$ and obtain a optimization problem w.r.t. $\mathbf{Z}$:

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \sum_{i=1}^{m} \mathbf{z}_i^T(\mathbf{K}_i \circ ((\mathbf{A}-\mathbf{I})(\mathbf{A}^T-\mathbf{I})))\mathbf{z}_i \\ \text{s.t.} \quad & \mathbf{Z}\mathbf{1}_m = \mathbf{1}_n \end{aligned} \tag{25}$$

where $\mathbf{z_i}$ is the i-th column of matrix $\mathbf{Z}$ and $\circ$ is the element-wise matrix dot product.

The above equation is formulated using matrix properties

$$\begin{aligned} &\mathrm{tr}(\mathbf{D}^T((\mathbf{c}\mathbf{c}^T)\circ\mathbf{B})\mathbf{D}) = \mathbf{c}^T(\mathbf{D}\mathbf{D}^T\circ\mathbf{B})\mathbf{c} \\ &\mathrm{tr}(((\mathbf{c}\mathbf{c}^T)\circ\mathbf{B})\mathbf{D}) = \mathbf{c}^T(\mathbf{D}^T\circ\mathbf{B})\mathbf{c} \end{aligned}$$

for matrix $\mathbf{B}$, $\mathbf{D}$ and vector $\mathbf{c}$ of suitable size.

Denote $\mathbf{G}$ as $\mathbf{G} = ((\mathbf{A}-\mathbf{I})(\mathbf{A}^T-\mathbf{I}))$, it is trivial to see that matrix $\mathbf{G}$ is positive-semidefinite, and Eq. (25) becomes

$$\begin{aligned} \min_{\mathbf{Z}} \quad & f(\mathbf{Z}) = \sum_{i=1}^{m} \mathbf{z}_i^T(\mathbf{K}_i \circ \mathbf{G})\mathbf{z}_i \\ \text{s.t.} \quad & \mathbf{Z}\mathbf{1}_m = \mathbf{1}_n \end{aligned} \tag{26}$$

To optimize Eq. (26), we can apply Proximal Gradient Descent [26] to solve $\mathbf{Z}$. More precisely, we denote $\mathbf{M}_t = \mathbf{K}_t \circ \mathbf{G}$ and $f(\mathbf{Z}) = \sum_{t=1}^{m} \mathbf{z}_t^T\mathbf{M}_t\mathbf{z}_t$, then linearize $f(\mathbf{Z})$ at $\mathbf{Z}^k$ and add a proximal term:

$$g_\mu(\mathbf{Z}, \mathbf{Z}^k) = f(\mathbf{Z}^k) + \langle \nabla f(\mathbf{Z}^k), \mathbf{Z}-\mathbf{Z}^k\rangle + \frac{\mu}{2}\|\mathbf{Z}-\mathbf{Z}^k\|_F^2 \tag{27}$$

where $\nabla f(\cdot)$ is the gradient of $f(\cdot)$, and $\mu > L(f)$ where $L(f)$ is Lipschitz constant of $f(\cdot)$ and $\mathbf{Z}^k$ denote $\mathbf{Z}$ at the k-th iteration.

Then we update $\mathbf{Z}$ by solving:

$$\mathbf{Z}^{k+1} = \arg\min_{\mathbf{Z}\geq 0, \mathbf{Z}\mathbf{1}_m=\mathbf{1}_n} \|\mathbf{Z} - \left(\mathbf{Z}^k - \frac{1}{\mu}\nabla f(\mathbf{Z}^k)\right)\|_F^2 \tag{28}$$

Let $\mathbf{H} = \mathbf{Z}^k - \frac{1}{\mu}\nabla f(\mathbf{Z}^k)$, to get $\mathbf{Z}^{k+1}$, we need to solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \|\mathbf{Z}-\mathbf{H}\|_F^2 \\ \text{s.t.} \quad & \mathbf{Z} \geq 0, \mathbf{Z}\mathbf{1}_m = \mathbf{1}_n \end{aligned} \tag{29}$$

Eq. (29) is row-decoupled and can be decomposed into n similar subproblems. The i-th subproblem can be seen as

$$\begin{aligned} \min_{\mathbf{z}^i} \quad & \|\mathbf{z}^i - \mathbf{h}^i\|_F^2 \\ \text{s.t.} \quad & \mathbf{z}^i \geq 0, |\mathbf{z}^i| = 1. \end{aligned} \tag{30}$$

Eq. (30) is equivalent to finding the point in the simplex which has smallest Euclidean distance to a given point. This problem is also known as Euclidean Projection onto Simplex and can be efficiently solved by root finding algorithm [27]. For completeness of this paper, we introduce this method as Algorithm 3.

According to [27], Algorithm 3 provides the global optimal solution of (30). So we can use the result of Algorithm 3 to update $\mathbf{Z}$.

Algorithm 3 The optimization algorithm of Euclidean Projection onto Simplex to solve Eq. (30)

Input: $\mathbf{h}$
sort $\mathbf{h}$ into $\mathbf{b}$ where $b_1 \geq b_2 \geq, ..., b_n$
find $\rho = \max\{1 \leq j \leq n : b_j + \frac{1}{j}(1 - \sum_{i=1}^{j} b_i) > 0\}$
define $z = \frac{1}{\rho}(1 - \sum_{i=1}^{\rho} b_i)$
Output: $\mathbf{z}$ with $z_j = \max\{h_j + z, 0\}, j = 1, ..., n$

Algorithm 4 The accelerated Proximal Gradient Descent algorithm to solve Eq. (26)

Input: The initial constant L_0, $\mathbf{Y}^0 = \mathbf{Z}^0$, γ.
Set $t = 0$, $\bar{L}_{\text{candi}} = L_0$, $a_0 = 1$
repeat
  Set $\bar{L}_{\text{candi}} = L_t$
  While $f(p_{\bar{L}_{\text{candi}}}(\mathbf{Y}^t)) > g_{\bar{L}_{\text{candi}}}(p_{\bar{L}_{\text{candi}}}(\mathbf{Y}^t), \mathbf{Y}^t)$ do
    Set $\bar{L}_{\text{candi}} = \gamma\bar{L}_{\text{candi}}$
  end while
  Set $L_{t+1} = \bar{L}_{\text{candi}}$
  Set $\mathbf{Z}^{t+1} = p_{L_t}(\mathbf{Y}^t)$
  Set $a_{t+1} = \frac{1+\sqrt{1+4a_t^2}}{2}$
  Set $\mathbf{Y}^{t+1} = \mathbf{Z}^{t+1} + (\frac{a_t-1}{a_{t+1}})(\mathbf{Z}^{t+1} - \mathbf{Z}^t)$
  Set $t = t + 1$
until Converges
Output: $\mathbf{Z}_t$

Although Proximal Gradient Descent can be used to solve Eq. (26), the converge rate is slow, i.e. $O(\frac{1}{\epsilon})$ [28] [29]. To achieve more efficient optimization, we apply accelerated Proximal Gradient Descent [30] to accelerate the proximal gradient descent, which has the convergence rate as $O(\frac{1}{\sqrt{\epsilon}})$.

We construct a linear combination of $\mathbf{Z}^k$ and $\mathbf{Z}^{k-1}$ to update $\mathbf{Y}^k$ as follows:

$$\mathbf{Y}^k = \mathbf{Z}^k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\mathbf{Z}^k - \mathbf{Z}^{k-1}) \tag{31}$$

Then we substitute $\mathbf{Z}^k$ in Eq. (28) with $\mathbf{Y}^k$,

$$\mathbf{Z}^{k+1} = \arg\min_{\mathbf{Z}\geq 0, \mathbf{Z}\mathbf{1}_m=\mathbf{1}_n} \|\mathbf{Z} - \left(\mathbf{Y}^k - \frac{1}{\mu}\nabla f(\mathbf{Y}^k)\right)\|_F^2 \tag{32}$$

Algorithm 5 The algorithm of LMKAL

Input: A set of kernel matrices $\{\mathbf{K}_i\}_{i=1}^m$, size of selected subset p
Output: selected subset $\mathbf{L}$
Initialize the kernel weight $Z_{ij} = 1/m$ for each kernel
repeat
  Update the reconstruction matrix $\mathbf{A}$ by Algorithm 1
  Update the kernel weight $\mathbf{Z}$ by Algorithm 4
  Update the estimated kernel $\mathbf{K_z}$ by Eq. (19)
until Converge
Sort all rows of $\mathbf{A}$ decreasingly according to the row absolute sum and let $\mathbf{L}$ be indexes of the instances with the p largest sum
Output: selected indexes $\mathbf{L}$

Eq. (32) can be solved by Algorithm 3 as discussed before. Algorithm 4 shows the process of the accelerated Proximal

Gradient Descent to solve Eq. (26)) where $g_L(\cdot)$ is defined in Eq. (27), and $p_L(\cdot)$ is defined as Eq. (29) and solved with Algorithm 3.

The convergence of this algorithm is stated in the following theorem.

Theorem 2. *[30] Let $\mathbf{Z}^k$ be the sequence generated by Algorithm 4, then for any $k \geq 1$, we have*

$$f(\mathbf{Z}^k) - f(\mathbf{Z}^*) \leq \frac{2\gamma L \|\mathbf{Z}^1 - \mathbf{Z}^*\|_F^2}{(k+1)^2}, \tag{33}$$

where L is the Lipschitz constant of the gradient of $f(\mathbf{Z})$, and $\mathbf{Z}^* = \arg\min_{\mathbf{Z}} f(\mathbf{Z})$.

It is easy to verify that $f(\mathbf{Z})$ is Lipschitz continuous. Thus Theorem 2 shows that the convergence rate of the accelerated proximal gradient descent method is $O(\frac{1}{\sqrt{\epsilon}})$.

2) Optimizing w.r.t. $\mathbf{A}$ *when* $\mathbf{Z}$ *is fixed:* When $\mathbf{Z}$ is fixed, The optimization problem becomes Eq. (4) and can be solved exactly as in the globalised method using Algorithm 1. We summarize algorithm of localized multiple kernel active learning (LMKAL) in Algorithm 5.

VII. Experimental Results

To demonstrate the effectiveness of out method, we apply GMKAL and LMKAL for active learning tasks and compare them with several state-of-the-art single kernel active learning methods.

A. Datasets Description

In our experiment, we evaluate the performance of our proposed GMKAL and LMKAL algorithms on six datasets, four from the UCI machine learning repository and two real world datasets, namely Reuters21578 and ORL. Detailed descriptions of each dataset are illustrated on table III.

TABLE III. Datasets Description

Dataset	#Instance	#Feature	#Class
australian	690	14	2
sonar	208	60	2
image segmentation	210	19	2
heart	270	13	2
ORL	400	1024	40
Reuters	2919	18933	4

UCI datasets. The first four dataset comes from UCI repository, including dataset *australian*, *sonar*, *image segmentation* and *heart*. All these four datasets are binary datasets.

Reuters21578. The fifth dataset is a subset of the Reuters21578 text dataset. This subset has 2,919 documents, including categories 'acq', 'crude', 'trade', and 'money', each with 2,025, 321, 298, and 245 documents respectively.

ORL. The last datasets contains ten different images of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). We use the resized version of ORL where each image has resolution of 32×32[13].

B. Experimental Details

In our experiment, we apply in total 9 different kernel functions as basis for MKAL. These kernels are 6 RBF kernels, 2 polynomial kernels and a linear kernel. The RBF kernel we use is defined as $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2t^2})$ with $t = t_0 \times d_{max}$, where d_{max} is the maximal distance between samples and t_0 varies in the range of $\{1, 10^1, 10^2, \ldots, 10^5\}$, and polynomial kernels as $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^a$ with $a = 2, 4$. Finally, all kernels are normalized by dividing each entry by the largest pair-wise squared distance so that each entry lies in $[0, 1]$.

We randomly divide each dataset into unlabeled set (70%) and testing set (30%). Each active learning algorithm selects data instances in the unlabeled set (70%) to query for labels and then the performance of each algorithm is measured by the classification accuracy on testing set (30%). For each single kernel algorithm, we fix the kernel as one of the 9 candidate kernels. And then we use the fixed single kernel active learning algorithm to select data examples form the unlabeled dataset. In total, we get 9 different results for each single kernel method. We sort the 9 different results in terms of average accuracy on all the evaluation points and report the top-3 and the median results in terms of average accuracy. Additionally, we average all the candidate kernels and report the single kernel methods with this averaged kernel. The evaluation point of each dataset is set as $\{5, 10, 15, \ldots\}$ depending on the dataset size. So when the algorithms select k samples ($k = 5, 10, 15...$), a classifier is trained on the selected labeled data samples and tests on the testing data.

Three baseline methods are compared in our experiment.

- The first baseline is random sampling
- The second is the kernel version of CTED [3], denoted as Kernel CTED (See Section IV).
- The third is the robust version of CTED [5] based on robust representation and structured sparsity, denoted as RRSS

In our experiment, all the algorithms start with the same unlabeled and testing dataset. The experiment is repeated 10 times and the average result is reported. Support Vector Machines is used as classification model to evaluate the performance of the labeled instances. Trade-off parameters of all methods are chosen from $\{10^{-3}, 10^{-2}, ..., 10^3\}$. All the trade-off parameters are selected via 5-fold Cross Validation. We use this scheme to all the methods in the experiment for fair comparison.

C. Comparative study

In this section, we evaluate the performance of our methods (GMKAL and LMKAL) in terms of classification accuracy. We fix the single kernel methods with 9 different candidate kernels used in our multiple kernel versions. For the 9 different results, we sort them according to the average accuracy over all the evaluation points. For each method, we compare our multiple kernel versions with the top-3, the median, and the results

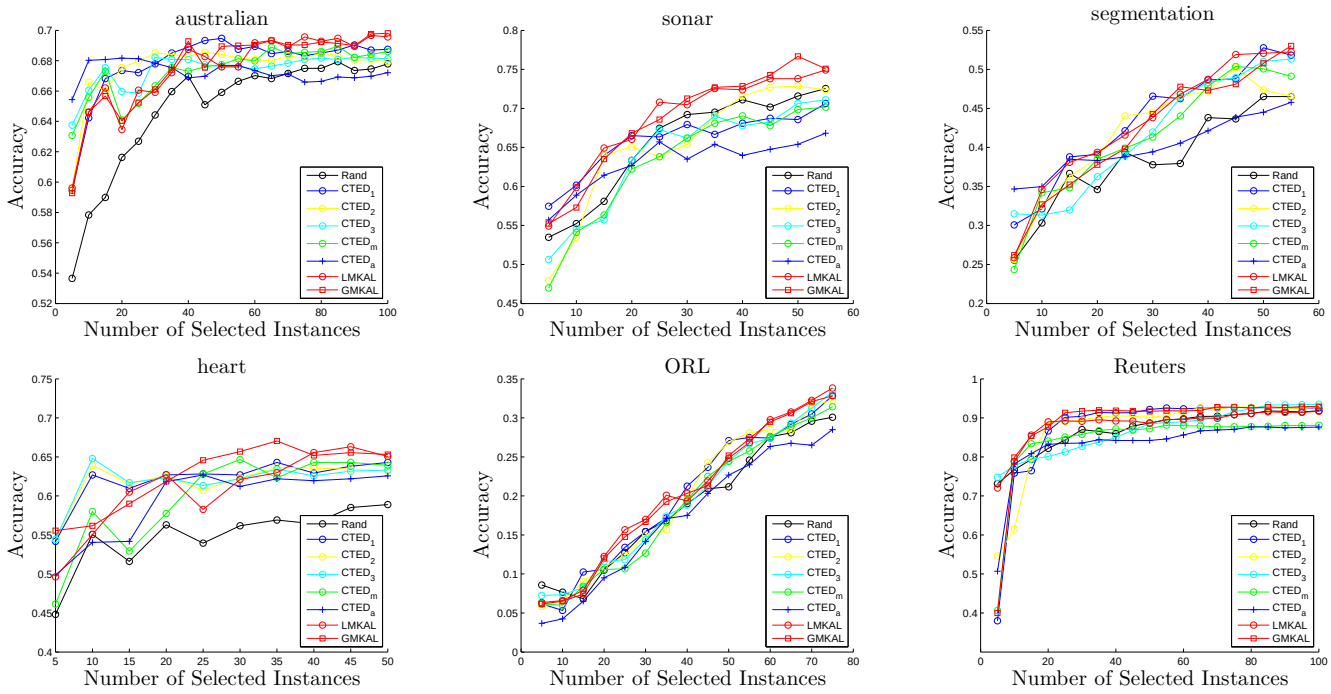

Fig. 1. Average accuracy of GMKAL and LMKAL against single kernel CTED over 6 datasets: australian, sonar, image segmentation, heart, ORL, Reuters, $CTED_1$, $CTED_2$, $CTED_3$, $CTED_m$, $CTED_a$ represent the top-3 single kernel, the median kernel and the average kernel, respectively.

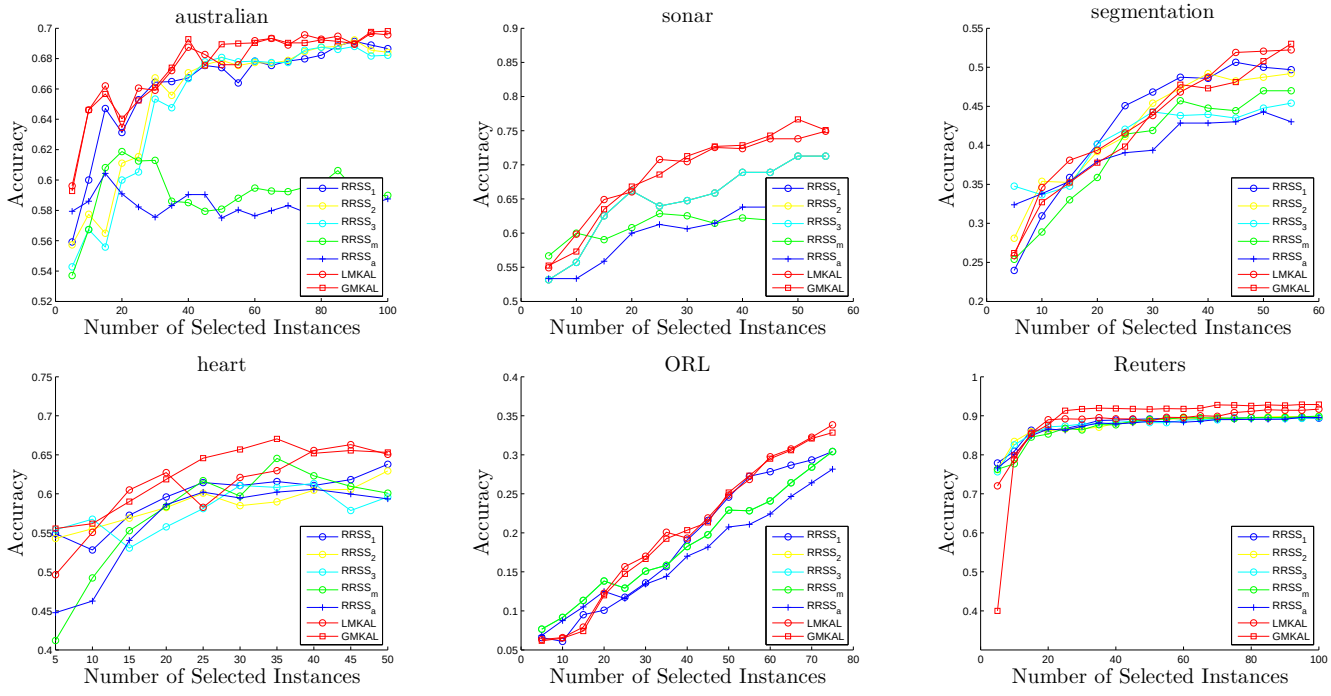

Fig. 2. Average accuracy of GMKAL and LMKAL against single kernel RRSS over 6 datasets: australian, sonar, image segmentation, heart, ORL, Reuters, $RRSS_1$, $RRSS_2$, $RRSS_3$, $RRSS_m$, $RRSS_a$ represent the top-3 single kernel, the median kernel and the average kernel, respectively.

TABLE I. THE WIN/LOSS(%) PERCENTAGE OF GLOBALISED MKAL AGAINST KERNEL CTED AND RRSS ON EVALUATION POINTS (WIN% OVER 70% ARE HIGHLIGHTED)

	australian		sonar		image segmentation		heart		ORL		Reuters	
	win(%)	**loss**(%)	**win**(%)	**loss**(%)	**win**(%)	**loss**(%)	**win**(%)	**loss**s(%)	**win**(%)	**loss**(%)	**win**(%)	**loss**(%)
$CTED_1$	**75**	25	**73**	27	64	36	**80**	20	**100**	0	**95**	5
$CTED_2$	55	45	**100**	0	**91**	9	**90**	10	**87**	13	**75**	25
$CTED_3$	**75**	25	**82**	18	**91**	9	**80**	20	**100**	0	70	30
$CTED_m$	60	40	**100**	0	27	73	70	30	**73**	27	**95**	5
$CTED_a$	65	35	**91**	9	**100**	0	**90**	10	67	33	**95**	5
$RRSS_1$	**80**	20	**100**	0	36	64	**100**	0	**80**	20	**85**	15
$RRSS_2$	**85**	15	**100**	0	27	73	**100**	0	**73**	27	**85**	15
$RRSS_3$	**95**	5	**100**	0	54	46	**90**	10	**73**	27	**90**	10
$RRSS_m$	**100**	0	**82**	18	**91**	9	**100**	0	**73**	27	**95**	5
$RRSS_a$	**100**	0	**100**	0	64	36	**100**	0	**73**	27	**90**	10

TABLE II. THE WIN/LOSS(%) PERCENTAGE OF LOCALIZED MKAL AGAINST KERNEL CTED AND RRSS ON EVALUATION POINTS (WIN% OVER 70% ARE HIGHLIGHTED)

	australian		sonar		image segmentation		heart		ORL		Reuters	
	win(%)	**loss**(%)	**win**(%)	**loss**(%)	**win**(%)	**loss**(%)	**win**(%)	**loss**s(%)	**win**(%)	**loss**(%)	**win**(%)	**loss**(%)
$CTED_1$	70	30	**73**	27	**73**	27	40	60	**100**	0	60	40
$CTED_2$	50	50	**100**	0	**91**	9	**80**	20	**80**	20	20	80
$CTED_3$	65	35	**91**	9	**91**	9	**80**	20	**100**	0	25	75
$CTED_m$	60	40	**100**	0	54	46	**80**	20	**80**	20	**100**	0
$CTED_a$	60	40	**100**	0	**100**	0	70	30	60	40	**100**	0
$RRSS_1$	**90**	10	**91**	9	64	36	**80**	20	**80**	20	**80**	20
$RRSS_2$	**85**	15	**91**	9	54	46	70	30	**73**	27	**80**	20
$RRSS_3$	**90**	10	**91**	9	64	36	**80**	20	**73**	27	**90**	10
$RRSS_m$	**100**	0	**82**	18	**100**	0	**80**	20	**73**	27	**85**	15
$RRSS_a$	**100**	0	**100**	0	**91**	9	**90**	10	**73**	27	**90**	10

with averaged kernel. Note that using average of kernels as a baseline is a common strategy in multiple kernel learning, and the effectiveness of MKL mainly depends on the performance against single kernel methods with averaged kernels.Table I and Table II shows the win/loss percentage of GMKAL and LMKAL against the single kernel methods on all different evaluation points for every dataset.

1) Compare with kernel CTED: Figure 1 reveals the average accuracy of our two methods against the kernel CTED algorithms on each evaluation point of 6 datasets over 10 runs. Table I and Table II shows that GMKAL has comparable data selection performance against the top-3 single kernels in datasets *heart*, *ORL* and *sonar*, for which the best kernel are not known in advance in practice applications. Additionally, we find that GMKAL outperforms all the single kernel algorithms with averaged candidate kernels ($CTED_a$ in Table I), along with 5 of the median kernel results (except $CTED_m$ in *image segmentation*). Table II shows that the Localized MKAL outperform baseline methods significantly in terms of top-3 results in dataset *sonar*, *segmentaion*, *ORL*. Additionally, LMKAL has better performance (over 50% win) than the average kernel CTED and RRSS results in all the datasets.

2) Compare with RRSS: RRSS [5] is the robust version of kernel CTED in that it replaces the Frobenius Norm with the $l_{2,1}$ norm which suffers less from outliers. We compare our GMKAL and LMKAL methods with RRSS on the same 6 datasets as in the previous section. RRSS runs slowly in practice because it has time complexity of $\mathcal{O}(n^4)$ per iteration, so it is very hard to conduct Cross Validation with this method. Empirically, we fix the trade-off parameter in RRSS that controls the sparsity as 0.01, which may lead to suboptimal solutions.

Our proposed GMKAL and LMKAL algorithms has significant improvement against RRSS on the 6 datasets illustrated in Fig. 2. Our two methods usually outperform the top-3 best fixed kernel RRSS in datasets *sonar* and *ORL*. From Table I and Table II, we can conclude that our GMKAL and LMKAL method outperforms the methods with averaged kernels and median-performance kernels (over 50% win), which shows the superiority of our methods against the common baseline RRSS.

VIII. DISCUSSION OF GLOBAL AND LOCAL METHODS

In the 6 different datasets above, we can conclude that global and local methods has similar performance on all the 6 datasets. We find it necessary to compare the global and local methods because they all have advantages and disadvantages. For the globalised MKAL algorithms, The formulation is straightforward, and the calculation is easier because the weight $\mathbf{w}$ on candidate kernels has analytic solution (Eq. (17)). The disadvantage of the globalised method is that it assigns the same weight to the same kernel, which is prone to noise and cannot discover local structures. On the other hand, the localized MKAL methods assign weight to each datum-kernel pair, because some kernels may be useful to some input data points and noisy to others.

However, the localized MKAL methods comes with a price: There are more variables to calculate. Instead of selecting m different variables, it assigns $n \times m$ variables which is linear to the number of input data points. When the size of input data goes up, the localized MKAL algorithms will suffer from computation and storage issues. The experimental results in Table I and Table II illustrate the empirical performance

of globalised MKAL and localized MKAL. We can see from the two tables that localized MKAL outperforms globalised MKAL in dataset *image segmentation*.

IX. Conclusion

Active Learning is a methodology that selects informative instances to label and trains a classifier using the selected samples. One family of active learning algorithms is based on Optimum Experimental Design in statistics. This line of work usually can be transformed into kernel based algorithms, and the performance of such active learning algorithms is highly related to the kernel of choice. However, in practice, the optimal kernel for data selection is usually not known in advance, and data-driven kernel selection approaches such as Cross Validation can not be applied in the early stage of active learning where no labeled data can be found. Fortunately, the already extensively studied topic Multiple Kernel Learning can be of help. MKL algorithms usually combines different candidate kernels from a predefined set to form a consensus kernel. Additionally, MKL methods can be embedded in to existing framework such as active learning to gain better results. In this paper, we propose two multiple kernel active learning (MKAL) algorithms, namely Globalised Multiple Kernel Active Learning (GMKAL) and Localized Multiple Kernel Active Learning (LMKAL) without additional parameters. The globalised one treats all candidate kernels equally, while the localized one assign different weight to the datum-kernel pair. Extensive experimental results on 6 different datasets demonstrate the effectiveness of our algorithm against other single kernel methods.

X. Acknowledgments

This work is supported in part by China National 973 program 2014CB340301, NSFC grant 61379043, 61322211, 61502289. Yi-Dong Shen and Liang Du are corresponding authors.

References

[1] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine learning*, vol. 28, no. 2-3, pp. 133–168, 1997.

[2] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 1070–1079.

[3] K. Yu, S. Zhu, W. Xu, and Y. Gong, "Non-greedy active learning for text categorization using convex ansductive experimental design," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008, pp. 635–642.

[4] K. Yu and J. Bi, "Active learning via transductive experimental design," in *In Machine Learning, Proceedings of the Twenty-Third International Conference (ICML*. ACM Press, 2006, pp. 1081–1088.

[5] F. Nie, H. Wang, H. Huang, and C. Ding, "Early active learning via robust representation and structured sparsity," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 1572–1578.

[6] F. Zhu and B. Fan, "10,000+ times accelerated robust subset selection (arss)," in *Proceedings of the Twenty-Ninth international joint conference on Artificial Intelligence*. AAAI Press, 2015, pp. 1572–1578.

[7] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 795–828, 2012.

[8] ——, "Multi-class classification with maximum margin multiple kernel," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 46–54.

[9] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[10] A. C. Atkinson, A. N. Donev, and R. D. Tobias, "Optimum experimental designs, with sas," 2007.

[11] Y. Hu, D. Zhang, Z. Jin, D. Cai, and X. He, "Active learning via neighborhood reconstruction," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013, pp. 1415–1421.

[12] L. Zhang, C. Chen, J. Bu, D. Cai, X. He, and T. S. Huang, "Active learning based on locally linear reconstruction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 10, pp. 2026–2038, 2011.

[13] D. Cai and X. He, "Manifold adaptive experimental design for text categorization," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 4, pp. 707–719, 2012.

[14] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[15] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[16] A. Afkanpour, A. György, C. Szepesvári, and M. Bowling, "A randomized mirror descent algorithm for large scale multiple kernel learning," *arXiv preprint arXiv:1205.0288*, 2012.

[17] M. Gönen and E. Alpaydin, "Localized multiple kernel learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 352–359.

[18] X. Liu, L. Wang, J. Zhang, and J. Yin, "Sample-adaptive multiple kernel learning," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[19] G. Tzortzis and A. Likas, "Kernel-based weighted multi-view clustering." in *ICDM*, 2012, pp. 675–684.

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

[21] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 551–556.

[22] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Affinity aggregation for spectral clustering," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 773–780.

[23] M. Hu, Y. Chen, and J.-Y. Kwok, "Building sparse multiple-kernel svm classifiers," *Neural Networks, IEEE Transactions on*, vol. 20, no. 5, pp. 827–839, 2009.

[24] S. C. Hoi, R. Jin, P. Zhao, and T. Yang, "Online multiple kernel classification," *Machine Learning*, vol. 90, no. 2, pp. 289–316, 2013.

[25] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[26] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[27] W. Wang and M. A. Carreira-Perpinán, "Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application," *arXiv preprint arXiv:1309.1541*, 2013.

[28] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[29] Y. Nesterov *et al.*, "Gradient methods for minimizing composite objective function," 2007.

[30] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 457–464.