Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

Self-weighted Multiple Kernel Learning for Graph-based Clustering and Semi-supervised Classification

Zhao Kang[1*], **Xiao Lu**[1], **Jinfeng Yi**[2], **Zenglin Xu**[1*],
[1] SMILE Lab, School of Computer Science and Engineering
University of Electronic Science and Technology of China, Sichuan 611731, China
[2] JD AI Research, Beijing 100101, China
zkang@uestc.edu.cn, nbshawnlu@hotmail.com, jinfengyi.ustc@gmail.com, zenglin@gmail.com

Abstract

Multiple kernel learning (MKL) method is generally believed to perform better than single kernel method. However, some empirical studies show that this is not always true: the combination of multiple kernels may even yield an even worse performance than using a single kernel. There are two possible reasons for the failure: (i) most existing MKL methods assume that the optimal kernel is a linear combination of base kernels, which may not hold true; and (ii) some kernel weights are inappropriately assigned due to noises and carelessly designed algorithms. In this paper, we propose a novel MKL framework by following two intuitive assumptions: (i) each kernel is a perturbation of the consensus kernel; and (ii) the kernel that is close to the consensus kernel should be assigned a large weight. Impressively, the proposed method can automatically assign an appropriate weight to each kernel without introducing additional parameters, as existing methods do. The proposed framework is integrated into a unified framework for graph-based clustering and semi-supervised classification. We have conducted experiments on multiple benchmark datasets and our empirical results verify the superiority of the proposed framework.

1 Introduction

As a principled way of introducing non-linearity into linear models, kernel methods have been widely applied in many machine learning tasks [Hofmann *et al.*, 2008; Xu *et al.*, 2010]. Although improved performance has been reported in a wide variety of problems, the kernel methods require the user to select and tune a single pre-defined kernel. This is not user-friendly since the most suitable kernel for a specific task is usually challenging to decide. Moreover, it is time-consuming and impractical to exhaustively search from a large pool of candidate kernels. Multiple kernel learning (MKL) was proposed to address this issue as it offers an automatical way of learning an optimal combination of distinct base kernels [Xu *et al.*, 2009]. Generally speaking, MKL method should yield a better performance than that of single kernel approach.

A key step in MKL is to assign a reasonable weight to each kernel according to its importance. One popular approach considers a weighted combination of candidate kernel matrices, leading to a convex quadratically constraint quadratic program. However, this method over-reduces the feasible set of the optimal kernel, which may lead to a less representative solution. In fact, these MKL algorithms sometimes fail to outperform single kernel methods or traditional non-weighted kernel approaches [Yu *et al.*, 2010; Gehler and Nowozin, 2009]. Another issue is the inappropriate weights assignment. Some attempts aim to learn the local importance of features by assuming that samples may vary locally [Gönen and Alpaydin, 2008]. However, they induce more complex computational problems.

To address these issues, in this paper, we model the differences among kernels by following two intuitive assumptions: (i) each kernel is a perturbation of the consensus kernel; and (ii) the kernel that is close to the consensus kernel should receive a large weight. As a result, instead of enforcing the optimal kernel being a linear combination of predefined kernels, this approach allows the most suitable kernel to reside in some kernels' neighborhood. And our proposed method can assign an optimal weight for each kernel automatically without introducing an additive parameter as existing methods do.

Then we combine this novel weighting scheme with graph-based clustering and semi-supervised learning (SSL). Due to its effectiveness in similarity graph construction, graph-based clustering and SSL have shown impressive performance [Nie *et al.*, 2017a; Kang *et al.*, 2017b]. Finally, a novel multiple kernel learning framework for clustering and semi-supervised learning is developed.

In summary, our main contributions are two-fold:

- We proposed a novel way to construct the optimal kernel and assign weights to base kernels. Notably, our proposed method can find a better kernel in the neighborhood of candidate kernels. This weight is a function of kernel matrix, so we do not need to introduce an additional parameter as existing methods do. This also eases the burden of solving the constraint quadratic program.

*Corresponding author.

- A unified framework for clustering and SSL is developed. It seamlessly integrates the components of graph construction, label learning, and kernel learning by incorporating the graph structure constraint. This allows them to negotiate with each other to achieve overall optimality. Our experiments on multiple real-world datasets verify the effectiveness of the proposed framework.

2 Related Work

In this section, we divide the related work into two categories, namely graph-based clusteirng and SSL, and paremeter-weighted multiple kernel learning.

2.1 Graph-based Clustering and SSL

Graph-based clustering [Ng *et al.*, 2002; Yang *et al.*, 2017] and SSL [Zhu *et al.*, 2003] have been popular for its simple and impressive performance. The graph matrix to measure the similarity of data points is crucial to their performance and there is no satisfying solution for this problem. Recently, automatic learning graph from data has achieved promising results. One approach is based on adaptive neighbor idea, i.e., s_{ij} is learned as a measure of the probability that X_i is neighbor of X_j. Then S is treated as graph input to do clustering [Nie *et al.*, 2014; Huang *et al.*, 2018b] and SSL [Nie *et al.*, 2017a]. Another one is using the so-called self-expressiveness property, i.e., each data point is expressed as a weighted combination of other points and this learned weight matrix behaves like the graph matrix. Representative work in this category include [Huang *et al.*, 2015; Li *et al.*, 2015; Kang *et al.*, 2018]. These methods are all developed in the original feature space. To make it more general, we develop our model in kernel space in this paper. Our purpose is to learn a graph with exactly c number of connected components if the data contains c clusters or classes. In this work, we will consider this condition explicitly.

2.2 Parameter-weighted Multiple Kernel Learning

It is well-known that the performance of kernel method crucially depends on the kernel function as it intrinsically specifies the feature space. MKL is an efficient way for automatic kernel selection and embedding different notions of similarity [Kang *et al.*, 2017a]. It is generally formulated as follows:

$$\min_{\theta} \quad f(K) \quad s.t. \quad K = \sum_i \theta_i H^i, \ \sum_i (\theta_i)^q = 1, \ \theta_i \geq 0, \tag{1}$$

where f is the objective function, K is the consensus kernel, H^i is our artificial constructed kernel, θ_i represents the weight for kernel H^i, $q > 0$ is used to smoothen the weight distribution. Therefore, we frequently sovle θ and tune q. Though this approach is widely used, it still suffers the following problems. First, the linear combination of base kernels over reduces the feasible set of optimal kernels, which could result in the learned kernel with limited representation ability. Second, the optimization of kernel weights may lead to inappropriate assignments due to noise and carelessly designed algorithms. Indeed, contrary to the original intention of MKL, this approach sometimes obtains lower accuracy than that of using equally weighted kernels or merely single kernel method. This will hinder the practical use of MKL. This phenomenon has been observed for many years [Cortes, 2009] but rarely studied. Thus, it is vital to develop some new approaches.

3 Methodology

3.1 Notations

Throughout the paper, all the matrices are written as uppercase. For a matrix X, its ij-th element and j-th column is denoted as x_{ij} and X_j, respectively. The trace of X is denoted by $Tr(X)$. The i-th kernel of X is written as H^i. The p-norm of vector x is represented by $\|x\|_p$. The Frobenius norm of matrix X is denoted by $\|X\|_F$. I is an identity matrix with proper size. $S \geq 0$ means all entries of S are nonnegative.

3.2 Self-weighted Multiple Kernel Learning

Aforementioned self-expressiveness based graph learning method can be formulated as:

$$\min_S \|X - XS\|_F^2 + \gamma \|S\|_F^2 \ \ s.t. \ \ S \geq 0, \tag{2}$$

where γ is the trade-off parameter. To recap the powerfulness of kernel method, we extend Eq. (2) to its kernel version by using kernel mapping ϕ. According to the kernel trick $K(x, z) = \phi(x)^T \phi(z)$, we have

$$\begin{aligned} &\min_S \ \|\phi(X) - \phi(X)S\|_F^2 + \gamma \|S\|_F^2, \\ \Longleftrightarrow &\min_S \ Tr(K - 2KS + S^T KS) + \gamma \|S\|_F^2 \\ &s.t. \quad S \geq 0 \end{aligned} \tag{3}$$

Ideally, we hope to achieve a graph with exactly c connected components if the data contain c clusters or classes. In other words, the graph S is block diagonal with proper permutations. It is straightforward to check that S in Eq. (3) can hardly satisfy to such a constraint condition.

If the similarity graph matrix S is nonnegative, then the Laplacian matrix $L = D - S$, where D is the diagonal degree matrix defined as $d_{ii} = \sum_j s_{ij}$, associated with S has an important property as follows [Mohar *et al.*, 1991]

Theorem 1 *The multiplicity c of the eigenvalue 0 of the Laplacian matrix L is equal to the number of connected components in the graph associated with S.*

Theorem 1 indicates that if $rank(L) = n - c$, then the constraint on S will be held. Therefore, the problem (3) can be rewritten as:

$$\begin{aligned} &\min_S \ Tr(K - 2KS + S^T KS) + \gamma \|S\|_F^2 \\ &s.t. \quad S \geq 0, \ rank(L) = n - c \end{aligned} \tag{4}$$

Suppose $\sigma_i(L)$ is the i-th smallest eigenvalue of L. Note that $\sigma_i(L) \geq 0$ because L is positive semi-definite. The problem (4) is equivalent to the following problem for a large enough α:

$$\begin{aligned} &\min_S \ Tr(K - 2KS + S^T KS) + \gamma \|S\|_F^2 + \alpha \sum_{i=1}^{c} \sigma_i(L) \\ &s.t. \quad S \geq 0 \end{aligned} \tag{5}$$

According to the Ky Fan's Theorem [Fan, 1949], we have:

$$\sum_{i=1}^{c}\sigma_i(L) = \min_{P\in\mathcal{R}^{n\times c}, P^TP=I} Tr(P^TLP) \tag{6}$$

P can be cluster indicator matrix or label matrix. Therefore, the problem (5) is further equivalent to the following problem

$$\begin{aligned}&\min_{S,P} Tr(K-2KS+S^TKS)+\gamma\|S\|_F^2+\alpha Tr(P^TLP)\\ &\quad s.t.\quad P^TP=I,\quad S\geq 0\end{aligned} \tag{7}$$

This problem (7) is much easier to solve compared with the rank constrained problem (4). We name this model as **K**ernel-based **G**raph **L**earning (KGL). Note that this model's input is kernel matrix K. It is generally recognized that its performance is largely determined by the choice of kernel. Unfortunately, the most suitable kernel for a particular task is often unknown in advance. Although MKL as in Eq. (1) can be applied to resolve this issue, it is still not satisfying as we discussed in subsection 2.2.

In this work, we design a novel MKL strategy. It is based on the following two intuitive assumptions: 1) each kernel is a perturbation of the consensus kernel, and 2) the kernel that is close to the consensus kernel should receive a large weight. Motivated by these, we can have the following MKL form:

$$\min_{K}\sum_{i} w_i\|H^i-K\|_F^2 \tag{8}$$

and

$$w_i = \frac{1}{2\|H^i-K\|_F} \tag{9}$$

We can see that w_i is dependent on the target variable K, so it is not directly available. But w_i can be set to be stationary, i.e., after obtaining K, we update w_i correspondingly [Nie *et al.*, 2017b]. Instead of enforcing the optimal kernel being a linear combination of candidate kernels as in Eq. (1), Eq. (8) allows the most suitable kernel to reside in some kernels' neighborhood [Liu *et al.*, 2009]. This enhances the representation ability of the learned optimal kernel [Liu *et al.*, 2017; 2013]. Furthermore, we don't introduce an additive hyperparameter θ, which often leads to a quadratic program. The optimal weight w_i for each kernel H^i is directly calculated according to kernel matrices. Then our Self-weighted Multiple Kernel Learning (SMKL) framework can be formulated as:

$$\begin{aligned}&\min_{S,P,K} Tr(K-2KS+S^TKS)+\gamma\|S\|_F^2+\alpha Tr(P^TLP)\\ &+\beta\sum_{i=1}^{r} w_i\|H^i-K\|_F^2 \quad s.t.\quad P^TP=I,\ S\geq 0.\end{aligned} \tag{10}$$

This model enjoys the following properties:

1. This unified framework sufficiently considers the negotiation between the process of learning the optimal kernel and that of graph/label learning. By iteratively updating S, P, K, they can be repeatly improved.
2. By treating the optimal kernel as a perturbation of base kernels, it effectively enlarges the region from which an optimal kernel can be chosen, and therefore is in a better position than the traditional ones to identify a more suitable kernel.
3. The kernel weight is directly calculated from kernel matrices. Therefore, we avoid solving quadratic program.

To see the effect of our proposed MKL method, we need to examine the approach with traditional kernel learning. For convenience, we denote it as Parameterized MKL (PMKL). It can be written as:

$$\begin{aligned}&\min_{S,P,\theta} Tr(K-2KS+S^TKS)+\gamma\|S\|_F^2+\alpha Tr(P^TLP)\\ &\quad s.t.\quad K=\sum_{i=1}^{r}\theta_iH^i,\ \sum_{i=1}^{r}\sqrt{\theta_i}=1,\ \theta_i\geq 0,\\ &\qquad P^TP=I,\ S\geq 0.\end{aligned} \tag{11}$$

3.3 Optimization

We divide the problem in Eq. (10) into three subproblems, and develop an alternative and iterative algorithm to solve them.

For S, we fix P and K. The problem in Eq. (10) becomes:

$$\begin{aligned}&\min_{S} Tr(-2KS+S^TKS)+\gamma\|S\|_F^2+\alpha Tr(P^TLP),\\ &\quad s.t.\quad S\geq 0.\end{aligned} \tag{12}$$

Based on $\sum_{ij}\frac{1}{2}\|P_{i,:}-P_{j,:}\|_2^2 s_{ij} = Tr(P^TLP)$, we can equivalently solve the following problem for each sample:

$$-2K_{i,:}S_i+S_i^TKS_i+\gamma S_i^TS_i+\frac{\alpha}{2}G_i^TS_i, \tag{13}$$

where $G_i\in\mathcal{R}^{n\times 1}$ with $g_{ij}=\|P_{i,:}-P_{j,:}\|_2^2$. By setting its first derivative w.r.t. S_i to be zero, we obtain:

$$S_i=(\gamma I+K)^{-1}(K_{i,:}-\frac{\alpha G_i}{4}). \tag{14}$$

Thus S can be achieved in parallel.

For K, we fix S and P. The problem in Eq. (10) becomes:

$$\min_{K} Tr(K-2KS+S^TKS)+\beta\sum_{i} w_i\|H^i-K\|_F^2 \tag{15}$$

Similar to (14), it yields:

$$K=\frac{2S^T-SS^T-I+2\beta\sum_i w_iH^i}{2\beta\sum_i w_i}. \tag{16}$$

From Eq. (16) and Eq. (14), we can observe that S and K are seamlessly coupled, hence they are allowed to negotiate with each other to achieve better results.

For P, we fix S and K. The problem in Eq. (10) becomes:

$$\min_{P} Tr(P^TLP)\quad s.t.\quad P^TP=I. \tag{17}$$

The optimal solution P is the c eigenvectors of L corresponding to the c smallest eigenvalues.

3.4 Extend to Semi-supervised Classification

Model (10) also lends itself to semi-supervised classification. Graph construction and label inference are two fundamental stages in SSL. Solving two separate problems only once is suboptimal since label information is not exploited when it learns the graph. SMKL unifies these two fundamental components into a unified framework. Then the given labels and estimated labels will be utilized to build the graph and to predict the unknown labels.

Based on a similar approach, we can reformulate SMKL for semi-supervised classification as:

$$\min_{S,P,K} Tr(K - 2KS + S^T KS) + \gamma\|S\|_F^2 + \alpha Tr(P^T LP) + \beta \sum_i w_i \|H^i - K\|_F^2 \quad s.t. \quad S \geq 0,\ P_l = Y_l \tag{18}$$

where $Y_l = [y_1, \cdots, y_l]^T$ denote the label matrix and l is the number of labeled points. $y_i \in \mathcal{R}^{c\times 1}$ is one-hot and $y_{ij} = 1$ indicates that the i-th sample belongs to the j-th class. $P = [P_l; P_u] = [Y_l; P_u]$, where the unlabeled u points in the back. (18) can be solved in the same procedure as (10), the difference lies in updating P.

To solve P, we take the derivative of (18) with respect to P, we have $LP = 0$, i.e.,

$$\begin{bmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{bmatrix} \begin{bmatrix} Y_l \\ P_u \end{bmatrix} = 0.$$

It yields:

$$P_u = -L_{uu}^{-1} L_{ul} Y_l. \tag{19}$$

Finally, the class label for unlabeled points could be assigned according to the following decision rule:

$$y_i = \operatorname*{argmax}_j P_{ij}. \tag{20}$$

ALGORITHM 1: The Proposed Framework SMKL

Input: Kernel matrices $\{H^i\}_{i=1}^r$, parameters α, β, γ.
Initialize: Random matrix S, $K = \sum_i H^i/r$.
REPEAT
1: For each i, update the i-th column of S according to (14).
2: Calculate K by (16).
3: Update w by (9).
4: For clustering, calculte P as the c smallest eigenvectors of $L = D - S$ correspond to the c smallest eigenvalues. For SSL, calculte P according to (19).
UNTIL stopping criterion is met.

4 Clustering

In this section, we conduct clustering experiments to demonstrate the efficacy of our method.

Table 1: Statistics of the data sets

	# instances	# features	# classes
YALE	165	1024	15
JAFFE	213	676	10
YEAST	1484	1470	10
TR11	414	6429	9
TR41	878	7454	10
TR45	690	8261	10

4.1 Data Sets

We implement experiments on six publicly available data sets. We summarize the information of these data sets in Table 1. In specific, the first two data sets YALE and JAFFE consist of face images. YEAST is microarray data set. Tr11, Tr41, and Tr45 are derived from NIST TREC Document Database.

We design 12 kernels. They are: seven Gaussian kernels of the form $K(x, y) = exp(-\|x - y\|_2^2/(td_{max}^2))$, where d_{max} is the maximal distance between samples and t varies over the set $\{0.01, 0.05, 0.1, 1, 10, 50, 100\}$; a linear kernel $K(x, y) = x^T y$; four polynomial kernels $K(x, y) = (a + x^T y)^b$ with $a \in \{0, 1\}$ and $b \in \{2, 4\}$. Besides, all kernels are rescaled to $[0, 1]$ by dividing each element by the largest pairwise squared distance.

4.2 Comparison Methods

We compare with a number of single kernel and multiple kernel learning based clustering methods. They include: Spectral Clustering (SC) [Ng *et al.*, 2002], Simplex Sparse Representation (SSR) [Huang *et al.*, 2015], Multiple Kernel k-means (MKKM) [Huang *et al.*, 2012b], Affinity Aggregation for Spectral Clustering (AASC) [Huang *et al.*, 2012a], Robust Multiple Kernel k-means (RMKKM)[1] [Du *et al.*, 2015]. Among them, SC, SSR, AASC are graph based methods, while the others are k-means variants. Although there are much recent work focused on multiview [Huang *et al.*, 2018a] and deep learning based clustering [Peng *et al.*, 2016], they are not our focus in this paper.

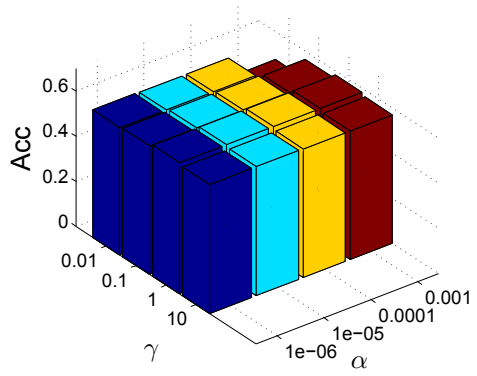
4.3 Performance Evaluation

SMKL is compared with other techniques. We show the clustering results in terms of accuracy (Acc), NMI in Table 2. For SC and KGL, we report its best performance achieved from those 12 kernels. It can clearly be seen that SMKL achieves the best performance in most cases. Compared to PMKL, SMKL[2] works better. As shown in Eq. (10) and (11), this is attributed to our new MKL scheme. Note that, KGL outperforms PMKL in several experiments. This is consistent with previous work's claim that MKL may degrade the performance. However, our proposed SMKL can beat KGL in all cases. This also demonstrates the effectiveness of our MKL strategy. With respect to recently developed methods SSR and RMKKM, we also observe considerable improvement. Remember that SSR is based on self-expressiveness

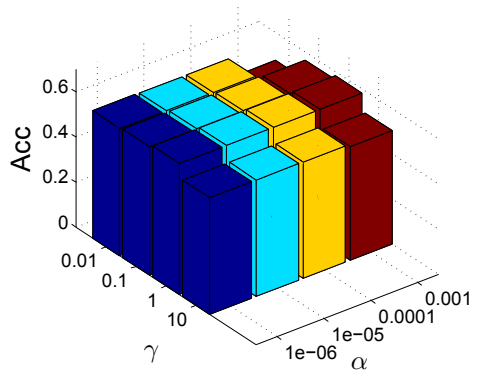
[1] https://github.com/csliangdu/RMKKM
[2] https://github.com/sckangz/IJCAI2018

Data	Metric	SC	SSR	MKKM	RMKKM	AASC	KGL	PMKL	SMKL
YALE	Acc	0.4942	0.5455	0.4570	0.5218	0.4064	0.5549	0.5605	**0.6000**
	NMI	0.5292	0.5726	0.5006	0.5558	0.4683	0.5498	0.5643	**0.6029**
JAFFE	Acc	0.7488	0.8732	0.7455	0.8707	0.3035	0.9877	0.9802	**0.9906**
	NMI	0.8208	0.9293	0.7979	0.8937	0.2722	0.9825	0.9806	**0.9834**
YEAST	Acc	0.3555	0.2999	0.1304	0.3163	0.3538	0.3892	0.3952	**0.4326**
	NMI	0.2138	0.1585	0.1029	0.2071	0.2119	0.2315	0.2361	**0.2652**
TR11	Acc	0.5098	0.4106	0.5013	0.5771	0.4715	0.7425	0.7485	**0.8309**
	NMI	0.4311	0.2760	0.4456	0.5608	0.3939	0.6000	0.6137	**0.7167**
TR41	Acc	0.6352	0.6378	0.5610	0.6265	0.4590	0.6942	0.6724	**0.7631**
	NMI	0.6133	0.5956	0.5775	0.6347	0.4305	0.6008	**0.6500**	0.6148
TR45	Acc	0.5739	0.7145	0.5846	0.6400	0.5264	0.7425	0.7468	**0.7536**
	NMI	0.4803	0.6782	0.5617	0.6273	0.4190	0.6824	**0.7523**	0.6965

Table 2: Clustering results on benchmark data sets. The best results are in bold font.



(a) $\beta = 10$



(b) $\beta = 100$

Figure 1: The influence of parameters on accuracy of YALE data set.

in the original space. Compared to traditional methods SC, MKKM, AASC, our advantages become more obvious.

4.4 Parameter Analysis

There are three parameters in our model (10). Figure 1 shows the clustering accuracy of YALE data set with varying α, β, and γ. We can observe that the performance is not so sensitive to those parameters. This conclusion is also true for NMI.

5 Semi-supervised Classification

In this section, we assess the effectiveness of SMKL on semi-supervised classification task.

5.1 Data Sets

1) **Evaluation on Face Recognition**: We examine the effectiveness of our graph learning for face recognition on two frequently used face databases: YALE and JEFFE. The YALE face data set contains 15 individuals, and each person has 11 near frontal images taken under different illuminations. Each image is resized to 32×32 pixels. The JAFFE face database consists of 10 individuals, and each subject has 7 different facial expressions (6 basic facial expressions +1 neutral). The images are resized to 26×26 pixels.

2) **Evaluation on Digit/Letter Recognition**: In this experiment, we address the digit/letter recognition problem on the BA database. The data set consists of digits of "0" through "9" and letters of capital "A" to "Z". Therefore, there are 39 classes and each class has 39 samples.

3) **Evaluation on Visual Object Recognition**: We conduct visual object recognition experiment on the COIL20 database. The database consists of 20 objects and 72 images for each object. For each object, the images were taken 5 degrees apart as the object is rotating on a turntable. The size of each image is 32×32 pixels.

Similar to clustering experiment, we construct 7 kernels for each data set. They include: four Gaussian kernels with t varies over $\{0.1, 1, 10, 100\}$; a linear kernel $K(x, y) = x^T y$; two polynomial kernels $K(x, y) = (a + x^T y)^2$ with $a \in \{0, 1\}$.

Data	Labeled (%)	GFHF	LGC	S^3R	S^2LRR	SCAN	SMKL
YALE	10	38.0±11.91	47.33±13.96	38.83±8.60	28.77±9.59	45.07±1.30	**55.87**±12.26
	30	54.13±9.47	63.08±2.20	58.25±4.25	42.58±5.93	60.92±4.03	**74.08**±1.92
	50	60.28±5.16	69.56±5.42	69.00±6.57	51.22±6.78	68.94±4.57	**82.44**±3.61
JAFFE	10	92.85±7.76	96.68±2.76	97.33±1.51	94.38±6.23	96.92±1.68	**97.57**±1.55
	30	98.50±1.01	98.86±1.14	99.25±0.81	98.82±1.05	98.20±1.22	**99.67**±0.33
	50	98.94±1.11	99.29±0.94	99.82±0.60	99.47±0.59	99.25±5.79	**99.91**±0.27
BA	10	45.09±3.09	48.37±1.98	25.32±1.14	20.10±2.51	**55.05**±1.67	46.62±1.98
	30	62.74±0.92	63.31±1.03	44.16±1.03	43.84±1.54	68.84±1.09	**68.99**±0.93
	50	68.30±1.31	68.45±1.32	54.10±1.55	52.49±1.27	72.20±1.44	**84.67**±1.06
COIL20	10	87.74±2.26	85.43±1.40	**93.57**±1.59	81.10±1.69	90.09±1.15	91.05±2.03
	30	95.48±1.40	87.82±1.03	96.52±0.68	87.69±1.39	95.27±0.93	**97.89**±2.00
	50	96.27±0.71	88.47±0.45	97.87±0.10	90.92±1.19	97.53±0.82	**99.97**±0.04

Table 3: Classification accuracy (%) on benchmark data sets (mean±standard deviation). The best results are in bold font.

5.2 Comparison Methods

We compare our method with several other state-of-the-art algorithms.

- **Local and Global Consistency (LGC)** [Zhou *et al.*, 2004]: LGC is a popular label propagation method. For this method, kernel matrix is used to compute L.
- **Gaussian Field and Harmonic function (GFHF)** [Zhu *et al.*, 2003]: Different from LGC, GFHF is another mechanics to infer those unknown labels as a process of propagating labels through the pairwise similarity.
- **Semi-supervised Classification with Adaptive Neighbours (SCAN)** [Nie *et al.*, 2017a]: Based on adaptive neighbors method, SCAN shows much better performance than many other techniques.
- **A Unified Optimization Framework for SSL** [Li *et al.*, 2015]: Li et al. propose a unified framework based on self-expressiveness approach. By using low-rank and sparse regularizer, they have S^2LRR and S^3R method, respectively.

5.3 Performance Evaluation

We randomly choose 10%, 30%, 50% portions of samples as labeled data and repeat 20 times. Classification accuracy and deviation are shown in Table 3. More concretely, for GFHF and LGC, the constructed seven kernels are tested and the best performance is reported. Unlike them, SCAN, S^2LRR, S^3R, and SMKL, the label prediction and graph learning are conducted in a unified framework.

As expected, the classification accuracy for all methods monotonically increase with the increase of the percentage of labeled samples. As can be observed, our SMKL method outperforms other state-of-the-art methods in most cases. This confirms the effectiveness of our proposed method on SSL task.

6 Conclusion

This paper proposes a novel multiple kernel learning framework for clustering and semi-supervised classification. In this model, a more flexible kernel learning strategy is developed to enhance the representation ability of the learned optimal kernel and to assign weight for each base kernel. An iterative algorithm is designed to solve the resultant optimization problem, so that graph construction, label learning, kernel learning are boosted by each other. Comprehensive experimental results clearly demonstrates the superiority of our method.

Acknowledgments

This paper was in part supported by two Fundamental Research Funds for the Central Universities of China (Nos. ZYGX2017KYQD177, ZYGX2016Z003), Grants from the Natural Science Foundation of China (No. 61572111) and a 985 Project of UESTC (No. A1098531023601041) .

References

[Cortes, 2009] Corinna Cortes. Can learning kernels help performance. In *Invited talk at International Conference on Machine Learning (ICML 2009). Montréal, Canada*, 2009.

[Du *et al.*, 2015] Liang Du, Peng Zhou, Lei Shi, Hanmo Wang, Mingyu Fan, Wenjian Wang, and Yi-Dong Shen. Robust multiple kernel k-means using l21-norm. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3476–3482. AAAI Press, 2015.

[Fan, 1949] Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

[Gehler and Nowozin, 2009] Peter Gehler and Sebastian Nowozin. On feature combination for multiclass object classification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 221–228. IEEE, 2009.

[Gönen and Alpaydin, 2008] Mehmet Gönen and Ethem Alpaydin. Localized multiple kernel learning. In *Proceedings of the 25th international conference on Machine learning*, pages 352–359. ACM, 2008.

[Hofmann *et al.*, 2008] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.

[Huang *et al.*, 2012a] Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. Affinity aggregation for spectral clustering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 773–780. IEEE, 2012.

[Huang *et al.*, 2012b] Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 20(1):120–134, 2012.

[Huang *et al.*, 2015] Jin Huang, Feiping Nie, and Heng Huang. A new simplex sparse learning model to measure data similarity for clustering. In *IJCAI*, pages 3569–3575, 2015.

[Huang *et al.*, 2018a] Shudong Huang, Zhao Kang, and Zenglin Xu. Self-weighted multi-view clustering with soft capped norm. *Knowledge-Based Systems*, 2018.

[Huang *et al.*, 2018b] Shudong Huang, Zenglin Xu, and Jiancheng Lv. Adaptive local structure learning for document co-clustering. *Knowledge-Based Systems*, 148:74–84, 2018.

[Kang *et al.*, 2017a] Zhao Kang, Chong Peng, and Qiang Cheng. Kernel-driven similarity learning. *Neurocomputing*, 267:210–219, 2017.

[Kang *et al.*, 2017b] Zhao Kang, Chong Peng, and Qiang Cheng. Twin learning for similarity and clustering: A unified kernel approach. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17). AAAI Press*, 2017.

[Kang *et al.*, 2018] Zhao Kang, Chong Peng, Qiang Cheng, and Zenglin Xu. Unified spectral clustering with optimal graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18). AAAI Press*, 2018.

[Li *et al.*, 2015] Chun-Guang Li, Zhouchen Lin, Honggang Zhang, and Jun Guo. Learning semi-supervised representation towards a unified optimization framework for semi-supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2767–2775, 2015.

[Liu *et al.*, 2009] Jun Liu, Jianhui Chen, Songcan Chen, and Jieping Ye. Learning the optimal neighborhood kernel for classification. In *IJCAI*, pages 1144–1149, 2009.

[Liu *et al.*, 2013] Xinwang Liu, Jianping Yin, Lei Wang, Lingqiao Liu, Jun Liu, Chenping Hou, and Jian Zhang. An adaptive approach to learning optimal neighborhood kernels. *IEEE transactions on cybernetics*, 43(1):371–384, 2013.

[Liu *et al.*, 2017] Xinwang Liu, Sihang Zhou, Yueqing Wang, Miaomiao Li, Yong Dou, En Zhu, Jianping Yin, and Han Li. Optimal neighborhood kernel clustering with multiple kernels. In *AAAI*, pages 2266–2272, 2017.

[Mohar *et al.*, 1991] Bojan Mohar, Y Alavi, G Chartrand, and OR Oellermann. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898):12, 1991.

[Ng *et al.*, 2002] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[Nie *et al.*, 2014] Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 977–986. ACM, 2014.

[Nie *et al.*, 2017a] Feiping Nie, Guohao Cai, and Xuelong Li. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *AAAI*, pages 2408–2414, 2017.

[Nie *et al.*, 2017b] Feiping Nie, Jing Li, and Xuelong Li. Self-weighted multiview clustering with multiple graphs. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2564–2570, 2017.

[Peng *et al.*, 2016] Xi Peng, Shijie Xiao, Jiashi Feng, Wei-Yun Yau, and Zhang Yi. Deep subspace clustering with sparsity prior. In *IJCAI*, pages 1925–1931, 2016.

[Xu *et al.*, 2009] Zenglin Xu, Rong Jin, Irwin King, and Michael Lyu. An extended level method for efficient multiple kernel learning. In *Advances in neural information processing systems*, pages 1825–1832, 2009.

[Xu *et al.*, 2010] Zenglin Xu, Rong Jin, Haiqin Yang, Irwin King, and Michael R Lyu. Simple and efficient multiple kernel learning by group lasso. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 1175–1182. Citeseer, 2010.

[Yang *et al.*, 2017] Yang Yang, Fumin Shen, Zi Huang, , Heng Tao Shen, and Xuelong Li. Discrete nonnegative spectral clustering. *IEEE Transactions on Knowledge and Data Engineering*, 29(9):1834–1845, 2017.

[Yu *et al.*, 2010] Shi Yu, Tillmann Falck, Anneleen Daemen, Leon-Charles Tranchevent, Johan AK Suykens, Bart De Moor, and Yves Moreau. L 2-norm multiple kernel learning and its application to biomedical data fusion. *BMC bioinformatics*, 11(1):309, 2010.

[Zhou *et al.*, 2004] Denny Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004.

[Zhu *et al.*, 2003] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003.