Neurocomputing 97 (2012) 201–213

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Regularized multi-view learning machine based on response surface technique

Zhe Wang [a,*], Jin Xu [a], Songcan Chen [b], Daqi Gao [a]

[a] Department of Computer Science & Engineering, East China University of Science & Technology, Shanghai 200237, PR China
[b] Department of Computer Science & Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, PR China

ARTICLE INFO

Article history:
Received 2 July 2011
Received in revised form
6 April 2012
Accepted 24 May 2012
Communicated by J. Kwok
Available online 30 June 2012

Keywords:
Multi-view learning
Response surface technique
Pattern representation
Rademacher complexity
Classifier design
Pattern recognition

ABSTRACT

Multi-view learning was supposed to process data with multiple information sources. Our previous work extended multi-view learning and proposed one effective learning machine named MultiV-MHKS. MultiV-MHKS firstly changed a base classifier into M different sub-classifiers, and then designed one joint learning process for the generated M sub-ones. Each sub-classifier was taken as one view of MultiV-MHKS. However, MultiV-MHKS assumed that each sub-classifier should play an equal role in the ensemble. Thus the weight values r_q, $q=1\ldots M$ for each sub-classifier were set to the equal value. In practice, this hypothesis was neither flexible nor appropriate since r_qs should reflect different effects of their corresponding views. In order to make r_qs flexible and appropriate, in this paper we propose a regularized multi-view learning machine named RMultiV-MHKS with the optimized r_qs. In this case, we optimize r_qs through using the Response Surface Technique (RST) on cross-validation data and thus can obtain a regularized multi-view learning machine. Doing so can assign a certain view with zero weight in the combination, which means that this specific view does not carry discriminative information for the problem and hence can be pruned. The experimental results here validate the effectiveness of the proposed RMultiV-MHKS and meanwhile explore the effect of some important parameters. The characters of the RMultiV-MHKS are: (1) distributing more weight to the favorable views which can reflect the property of the problem; (2) owning a tighter generalization risk bound than its corresponding single-view learning machine in terms of the Rademacher complexity; (3) having a statistically superior classification performance to the original MultiV-MHKS.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Multi-view learning is working with multi-view data represented by multiple information sources. Each information source can form one set of attributes and each attribute set is taken as one view of the original data. Multi-view learning was first proposed by de Sa [1]. De Sa [1] thought that when labels of the data were not available, different sensory modalities could be used to substitute for the labels. Here each sensory modality was taken as one view of the given data. Then based on the collocation and discourse senses, Yarowsky [2] applied the multi-view technique into the word sense disambiguation. Blum and Mitchell [3] further boosted the performance of learning algorithms by the co-training style and took the web page classification as an instance. Dasgupta et al. [4] gave an upper bound for the generalization error of multi-view learning algorithms, which was based on maximizing the agreement-based objective function suggested by Collins and Singer [5]. Moreover, Abney [6] showed the intuitive reason why multi-view learning worked, which was first gave by Blum and Mitchell [3]. The successful reason was that multi-view learning could maximize the agreement on unlabeled data between classifiers based on different views of the data [6].

In the literature [7], we generalized multi-view learning. Specifically, the acquired data for an object could be sorted into single-view data and multi-view data. Correspondingly, learning machine could also be sorted into single-view learning machine with only one architecture and multi-view learning machine with multiple architectures. Thus it gave four combinations, i.e. single-view learning machine on single-view data, single-view learning machine on multi-view data, multi-view learning machine on single-view data, and multi-view learning machine on multi-view data. Our work mainly focused on multi-view learning on single-view data due to the advantages of single-view data in terms of the acquisition cost and storage compared with multi-view data. Following this research line [7], we further changed the original architecture of a given base classifier into different matrixized sub-classifiers. Each matrixized sub-one could be taken as one view of the original base classifier. Consequently, we developed

* Corresponding author.
E-mail address: wangzhe@ecust.edu.cn (Z. Wang).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.05.027

one joint rather than separable learning process for different sub-classifiers, which was named as MultiV-MHKS [8].

MultiV-MHKS [8] was supposed to introduce the creation of multiple views from one single view and meanwhile mitigate the model selection problem of our another algorithm MatMHKS [9]. Moreover, MultiV-MHKS adopted the data representation in terms of multiple views and thus was different from some other learning strategies for creating good ensembles of classifiers such as sampling pattern sets or feature sets [10–12]. Finally, the technique behind MultiV-MHKS was a wrapper way and could act like the state-of-the-art kernelization technique [13] applied to linear algorithms. The experiments in the literature [8] also convincingly validated the feasibility and effectiveness of MultiV-MHKS.

On the other hand, it could be found that MultiV-MHKS took each matrixized sub-classifier in an equal way. Each sub-classifier played an equal important role in the final classification. Suppose that there were M views in MultiV-MHKS and the regularized parameters r_q, $q=1\ldots M$ denoted the weights of their corresponding views. It was known that the bigger the r_q was, the more important its corresponding view was. However, MultiV-MHKS set r_q with $1/M$ for simplicity, which meant that all the views were given an equal value in advance. Such a simple setting for r_q was not fit for the real-world case since different views would exhibit different information. It could not be thought that different information should play an equal role in classification.

In order to solve this problem, in this paper we introduce the Response Surface Technique (RST) [14] into MultiV-MHKS and develop a regularized and flexible multi-view learning machine named RMultiV-MHKS. It is known that RST is a statistical and mathematical technique for processing optimization and can improve the regularization of the model [14]. To be more exact, the proposed RMultiV-MHKS utilizes RST to distribute the heavier weights to the favorable views which can more likely reflect the properties of the problem. In contrast, it can also distribute the lighter, even zero, weights to the unfavorable views with no sufficient discriminative information. In other words, introducing RST into MultiV-MHKS amounts to assigning a certain matrix representation with zero weight(s) in the ensemble. As a result, a matrix associating with zero weight can be pruned due to its insufficient discriminative information. In practice, we realize RST through implementing MultiV-MHKS on the validation sets of the given datasets. Then we can optimize the weight parameters r_q based on the got validation errors. By comparing RMultiV-MHKS with MultiV-MHKS, it can be found that the former has a superior classification performance to the latter. More importantly, the designed algorithm is demonstrated to own a tighter generalization risk bound than its corresponding single-view learning machine in terms of the Rademacher complexity. The experimental results here further validate the effectiveness of the proposed algorithm and give the discussion for RMultiV-MHKS in terms of: (1) the initialized weight r_q^0; (2) the length of the searching step Δ; (3) the number of the matrixized views M.

The rest of this paper is organized as follows. Section 2 reviews the work including the optimization methods for the weights in multi-view learning and the family of different improved Ho-Kashyap (HK) [15] algorithms. Then we show the introduction of RST and describe the architecture of the proposed RMultiV-MHKS in Section 3. Section 4 demonstrates the feasibility and effectiveness of RMultiV-MHKS through different experimental strategies. Following that, we discuss the Rademacher complexity of RMultiV-MHKS in terms of theory and experiments. Finally, we conclude and give the future work in Section 6.

2. Related work

This section first reviews the related optimization method for the weight r_q in the multi-view learning. Then since the proposed algorithm is based on our previous work MultiV-MHKS [8] that is an improved HK algorithm, we also review the series of the related HK algorithms including the Modification of HK algorithm with Squared approximation of the misclassification errors (MHKS) [16], the matrixized MHKS (MatMHKS) [9], and the multi-view learning machine MultiV-MHKS.

2.1. The optimization for the weight of the multi-view learning

One typical example of multi-view learning worked for web-page classification [3], where each web page could be represented by the words on itself (view one) and the words contained in anchor texts of inbound hyperlinks (view two). In the literature [3], Blum and Mitchell designed a co-training algorithm for labeled and unlabeled web pattern sets composed of the two naturally split views. On the labeled web set, the two sub-classifiers of the co-training algorithm were incrementally built with their corresponding views. On each cycle, each sub-classifier labeled the unlabeled webs and picked the unlabeled webs with the highest confidence into the labeled set. Such a learning process was repeated until the terminated condition was satisfied. The final decision function was constructed through the average combination of the two sub-classifier. In this case, the weight value r_q for each view was set to $\frac{1}{2}$, $q=1,2$. Further, Collins and Singer [5], Dasgupta et al. [4], and Abney [6] developed multi-view learning, respectively. The workshop at International Conference on Machine Learning (2005) gave a special discussion for multi-view learning including unsupervised learning [17], semi-supervised learning [18], and supervised learning [19]. But in the above work, it could be found that the weigh value r_qs for all the views were all simply set to $1/M$, where M was the size of the views.

Multiple Kernel Discriminant Analysis (MKDA) [20–22] could be viewed as one kind of multi-view learning [7]. MKDA tried to introduce the advantage of multiple kernels into the discriminant analysis learning, where each kernel was used to generate one view of the original data [23,24]. MKDA transformed the original criterion of minimizing the within-class distance and meanwhile maximizing the between-class distance into a convex optimization problem, which could be solved by different methods. Firstly, Kim et al. [20] calculated the MKDA problem as a trackable Semi-Definite Programming (SDP), which could be efficiently solved through the interior-point technique. Secondly, MKDA focused on the norm of multi-kernel combination coefficients [21,22,25,26]. The ℓ_1 norm was the most commonly used regularization since it could bring the sparsity. However, the ℓ_1 norm regularization might lose some potential kernel information [21]. In order to overcome the problem caused by the ℓ_1 norm, Fei et al. [21] used ℓ_2 instead which was only fit for binary-class problem. For the multi-class problem, Fei et al. [22] constructed a general ℓ_p, $p\geq 1$ norm regularized MKDA for the combination coefficients of multiple kernels. In this case, the MKDA with ℓ_p could achieve a superior performance through the Semi-Infinite Programming (SIP) [22]. It was also supposed that MKDA tried to learn the optimal scaling of the feature space so as to maximize the separation of different classes in the transformed feature space. Thirdly, Gaïffas and Lecué [26] developed a hyper-sparse aggregation for the ensemble problem $\overline{f}=\sum_{i=1}^{M}\theta_i f_i$. Letting $F=\{f_1,\ldots,f_M\}$, Gaïffas and Lecué demonstrated that when the F contained the irrelevant functions which should not appear in the $\overline{f}$, the aggregation should maintain only two function f_i. In this case, there were only two coefficients with non-zero. It meant that the two was the minimal number of the elements in the F and their corresponding coefficients were exactly required for the construction of an optimal aggregation procedure. Moreover, Zhang and Yeung [25] applied the multi-view viewpoint into multi-task

learning. Thus they proposed a regularized convex formulation to learn the relationships between different tasks, where the proposed formulation was viewed as one novel generalization for single-task learning.

2.2. The family of HK algorithms

2.2.1. MHKS

The original HK algorithm was expected to obtain a good classification performance. But HK was sensitive to outliers [16]. In order to solve this problem, Leski proposed a modified HK algorithm named MHKS [16]. MHKS bases on the regularized least squares and tries to maximize the separating margin [27–29]. To be more specific, MHKS gives its separating hyperplane as follows:

$$Yw \geq 1_{N\times 1}. \tag{1}$$

Consequently, the criterion function of MHKS is changed as

$$\min_{\omega\in\mathbb{R}^{d+1},b\geq 0} L(w,b)=(Yw-1_{N\times 1}-b)^T(Yw-1_{N\times 1}-b)+c\tilde{w}^T\tilde{w}, \tag{2}$$

where $c\geq 0$ is the regularized hyper-parameter that adjusts the tradeoff between the model complexity and the training error. The procedure of MHKS is almost the same as that of the original HK classifier. The difference between MHKS and HK is that the argument weight vector w_k in MHKS becomes

$$w_k=(Y^TY+c\tilde{I})^{-1}Y^T(b_k+1_{N\times 1}), \tag{3}$$

where $\tilde{I}$ is an identity matrix with the last element on the main diagonal set to zero.

2.2.2. MatMHKS

Since vector representation for patterns fails in some image-based learning, some matrix-based algorithms were proposed in terms of both feature extraction [30–32] and classifier design [9]. MatMHKS was a typical matrixized classifier and could directly classify patterns represented with matrix. As a consequence, MatMHKS was viewed as a matrixized improvement of MHKS. In the matrix case, suppose that there is a binary-class classification problem with N matrix samples (A_i,φ_i), $i=1\ldots N$, where $A_i\in\mathbb{R}^{m\times n}$ and its corresponding class label $\varphi_i\in\{+1,-1\}$. The decision function of MatMHKS for the binary problem is given as

$$g(A_i)=u^TA_i\tilde{v}\begin{cases}>0, & \text{if } \varphi_i=+1\\ <0, & \text{if } \varphi_i=-1\end{cases}, \tag{4}$$

where both $u\in\mathbb{R}^m$ and $\tilde{v}\in\mathbb{R}^n$ are the weight vectors. The corresponding optimization function of MatMHKS is defined as follows:

$$\begin{aligned}\min_{u\in\mathbb{R}^m,\tilde{v}\in\mathbb{R}^n,v_0,b\geq 0} J(u,\tilde{v},v_0,b)=&\sum_{i=1}^{N}(\varphi_i(u^TA_i\tilde{v}+v_0)-1-b_i)^2\\&+c(u^TS_1u+\tilde{v}^TS_2\tilde{v}),\end{aligned} \tag{5}$$

where $S_1=mI_{m\times m}$, $S_2=nI_{n\times n}$ are the two regularized matrices corresponding to the weight vectors u and $\tilde{v}$ respectively, and the regularized parameter c ($c\in\mathbb{R},c\geq 0$) controls the generalization ability of the classifier through making a tradeoff between the classifier complexity and the training error. The vectors u, $\tilde{v}$, and the bias v_0 can be obtained by the gradient optimization of the formulation (5) with respect to u, $\tilde{v}$, and v_0 respectively. The detailed processing optimization can be referred in the literature [9].

2.2.3. MultiV-MHKS

In the literature [8], MHKS was supposed to be a single-view classifier and could be multiviewized into multiple matrixized MatMHKS. Then we adopted a joint learning for different MatMHKSs and proposed a multi-view learning machine MultiV-MHKS. In mathematics, suppose that there is an original vector pattern $x_i\in\mathbb{R}^d$. The x_i can be represented by different matrices $A_i^q\in\mathbb{R}^{m_q\times n_q}, q=1\ldots M$, where d is equal to $m_q\times n_q$. In MultiV-MHKS, we set $Y^q=[y_1^q,\ldots,y_N^q]^T$, $y_i^q=\varphi_i[u^{qT}A_i^q,1]^T$, $i=1\ldots N$, $b^q=[b_1^q,\ldots b_N^q]^T$, $v^q=[\tilde{v}^{qT},v_0^q]^T$, where the q denotes the index number of the view in MultiV-MHKS. Then the criterion function of MultiV-MHKS is given as follows:

$$\begin{aligned}\min_{\substack{u^q\in\mathbb{R}^{m^q},v^q\in\mathbb{R}^{n^q+1}\\ q=1,\ldots,M}} J'=&\sum_{q=1}^{M}((Y^qv^q-1_{N\times 1}-b^q)^T(Y^qv^q-1_{N\times 1}-b^q)\\&+c^q(u^{qT}S_1u^q+v^{qT}\tilde{S_2}v^q))\\&+\gamma\sum_{q=1}^{M}\left(Y^qv^q-\frac{1}{M}\sum_{p=1}^{M}(Y^pv^p)\right)^T\\&\times\left(Y^qv^q-\frac{1}{M}\sum_{p=1}^{M}(Y^pv^p)\right),\end{aligned} \tag{6}$$

where $S_1=m^qI_{m^q\times m^q}$, $S_2=n^qI_{n^q\times n^q}$, $\tilde{S_2}=\left(\begin{smallmatrix}S_2 & 0\\ 0 & 0\end{smallmatrix}\right)$ is a matrix with a dimensionality of $(n^q+1)\times(n^q+1)$, c^q is the regularized parameter for each view, and the γ is the coupling parameter. In the formulation (6), the weight value of each view is simply set to $1/M$. In this case, each MatMHKS plays an equal role in the whole classification. Then for optimizing the criterion function (6), we make the gradient of J' with respect to both u^q and v^q be zero. Therefore we can get the following optimal results:

$$\begin{aligned}u^q=&\left(\left(1+\gamma\left(\frac{M-1}{M}\right)^2\right)\sum_{i=1}^{N}A_i^q\tilde{v}^q(A_i^q\tilde{v}^q)^T+c^qS_1\right)^{-1}\\&\times\sum_{i=1}^{N}\left(A_i^q\tilde{v}^q\left(\varphi_i(b_i^q+1)\right.\right.\\&\left.\left.-\left(1+\gamma\left(\frac{M-1}{M}\right)^2\right)v_0^q+\gamma\frac{M-1}{M^2}\sum_{p=1,p\neq q}^{N-1}(u^{pT}A_i^p\tilde{v}^p+v_0^p)\right)\right),\end{aligned} \tag{7}$$

$$\begin{aligned}v^q=&\left(\left(1+\gamma\left(\frac{M-1}{M}\right)^2\right)Y^{qT}Y^q+c^q\tilde{S_2}\right)^{-1}Y^{qT}\\&\times\left(1_{N\times 1}+b^q+\gamma\frac{M-1}{M^2}\sum_{p=1,p\neq q}^{M}Y^pv^p\right).\end{aligned} \tag{8}$$

The iteration for both u^q and v^q is the same as that in MatMHKS. Since MultiV-MHKS is a joint learning for multiple views, its decision function integrates multiple MatMHKSs and is given as follows:

$$g(z)=\frac{1}{M}\sum_{q=1}^{M}(u^{qT}Z^q\tilde{v}^q+v_0^q)\begin{cases}>0 & \text{then } z\in\text{class}+1\\ <0 & \text{then } z\in\text{class}-1\end{cases}, \tag{9}$$

where z is the test sample and Z^q is the qth matrix representation for the z.

3. Proposed regularized multi-view learning machine (RMultiV-MHKS)

MultiV-MHKS was expected to make a full use of the advantage of different matrix representations. But the equal value with $1/M$ was a simple setting for the weight of each MatMHKS in MultiV-MHKS, which might be not sensible in some real-world cases. For example, one certain matrix representation supplied less even no useful information for discrimination, while the decision function (9) still took the less useful matrix representation into the final classification like the other useful ones. It urges us to assign different weights to the matrix representation with different matrix representations. In order to realize such an

assignment, in this paper we introduce RST into MultiV-MHKS and thus develop a regularized multi-view learning machine named RMultiV-MHKS. RMultiV-MHKS is supposed to optimize the weight of each matrix representation through RST. In doing so, RMultiV-MHKS can distribute the heavier weight to the favorable view which owns more discriminative information and the lighter even zero weight to the unfavorable view which might not carry discriminative information, which is expected to lead to a superior classification performance.

RST is supposed to take the advantage of both mathematics and statistics and to make the variable called response optimized through updating the variable called corresponding. RST has been proven an effective optimization technique in terms of model selection. Chapelle et al. [33] adopted RST to give a parameter selection for kernel-base methods. In their work, they achieved the selection through minimizing the estimated test error bound and updating the parameter with a gradient-descent step calculated from the error bound. Momma and Bennett [34] used RST to select the parameters of support vector regression. Rather than fitting a response surface, they performed a moving grid search strategy through changing the center point of the grid. Gönen and Alpaydın [35] applied RST into the ensemble of kernels. They firstly fitted an approximate response surface based on the validation error and then searched the minimal point of the fitted response. Blum et al. [36] introduced RST into the protein structure prediction and optimized the parameters of a specific function called Rosetta energy function. In detail, they first formulated a feature function and eliminated some of the features. Further, they calculated the response surface using the remaining features.

These RST-based work [33–36] showed that using RST on validation error to optimize parameters or models could obtain more regularized solutions and meanwhile illustrated that optimizing the regularization parameters using the RST-based approach could lead to more sparse ensembles, where some sub-models would be given zero weight without the loss of discriminant information. In this paper, we introduce the advantage of RST into MultiV-MHKS. Doing so can assign one certain matrix representation with zero weight in the combination (9). In this case, it means that this specific matrix representation or the data represented with this matrix size does not carry discriminative information for the problem and hence can be pruned. In practice, we use RST to construct a response surface from some known points. Then we can get the extreme point of the surface through some optimization methods. It should be stated that both the proposed work here and the work in [35] adopt RST since RST is a general statistical and mathematical technique for processing optimization. But differently from the work in [35], our work applies RST into a generalized multi-view learning that is especially designed for multiple matrix representations. Meanwhile the proposed algorithm falls into a linear learning framework.

In the proposed method, the form of the response surface can be represented by a quadratic equation or the other equation with higher degrees. Here, we adopt a quadratic equation to fit the response surface with rationality and simplicity. For the quadratic equation, we give its general form as follows:

$$R=\beta_0+\sum_{i=1}^{M}\beta_i X_i+\sum_{i=1}^{M}\beta_{ii}X_i^2+\sum_{i=1}^{M}\sum_{j=i+1}^{M}\beta_{ij}X_iX_j, \tag{10}$$

where the R is named as the *response* variable, the X_i is named as the corresponding variable, and the β_0, β_i, β_{ii}, and $\beta_{ij}\in\mathbb{R}$ are the model parameters. For fitting the quadratic equation R, we need $(M+1)(M+2)/2$ groups of points $(X_i, i=1\ldots M)$ so as to estimate the model parameters. Through the optimized model parameters, we can get the response surface. In practice, we use the Newton optimization method [15] to get the minimal value of the R.

We firstly assign the response and corresponding variables of Eq. (10) with the parameters of the proposed RMultiV-MHKS. We define the variable r_q as the weight of the qth view in the whole model. The r_q can reflect the corresponding role of the qth view. Here, we take the r_q as the corresponding variable and the validation classification error of the whole RMultiV-MHKS as the response variable R. The aim of RMultiV-MHKS is to get the lowest validation error through optimizing the weight r_q. In processing, we give the natural logarithm scale of r_q so as to keep $r_q>0$. As a consequence, we can introduce the r_q and the validation classification error into the quadratic response surface (10) as follows:

$$R=\beta_0+\sum_{q=1}^{M}\beta_q\ln(r_q)+\sum_{q=1}^{M}\beta_{qq}\ln^2(r_q)+\sum_{q=1}^{M}\sum_{p=q+1}^{M}\beta_{qp}\ln(r_q)\ln(r_p). \tag{11}$$

At first, RMultiV-MHKS solves the model parameters β_0, β_i, β_{ii}, and β_{ij} of Eq. (11) through a group of initialized $\{r_q\}_{q=1}^{M}$. Then according to the got response surface (11), RMultiV-MHKS adopts the Newton optimization to get the optimal r_qs. Therefore RMultiV-MHKS is supposed to consist of the two-cycle procedure which is described in Fig. 1.

In the first cycle of Fig. 1 with the purpose of estimating the response surface (11), we create a matrix $H\in\mathbb{R}^{(M+1)\times N}$ with each zero element. The H is used to store the $(M+1)(M+2)/2$ groups of the got $\{\ln(r_q)\}_{q=1}^{M}$ and the R. We initialize the r_q^0, $q=1\ldots M$ with the value $1/M$ and set the k as the index number of the first cycle. Further, we can obtain the following corresponding variables $\{r_q^k\}_{q=1}^{M}$ through moving them towards positive or negative direction from the original corresponding variables $\{r_q^0\}_{q=1}^{M}$, where the length of the searching step is set to Δ. At each iteration, we introduce the got $\{r_q^k\}_{q=1}^{M}$ into the original Multi-V-MHKS that is named the base classifier here. Then we can get the corresponding validation error cr^k that is viewed as the

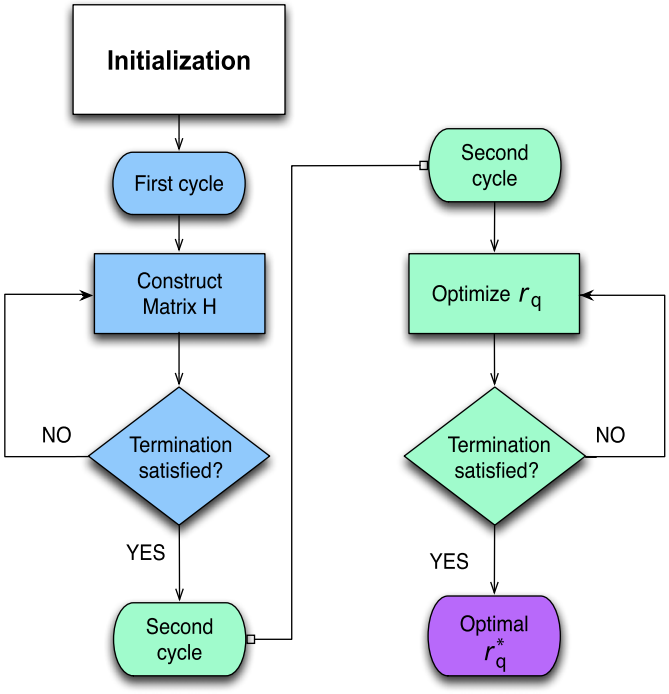


Fig. 1. The flowchart of the two-cycle procedure in RMultiV-MHKS.

response variable R. Each column of the matrix H here can be set to $[\ln(r_1^k),\ldots,\ln(r_M^k),cr^k]^T$.

Through the second cycle of Fig. 1, RMultiV-MHKS can obtain the optimal r_q^*, $q=1\ldots M$ according to the matrix H. Here we firstly give a variable $j=1$ that denotes the index number of the second cycle. Then we solve the model parameters β_0, β_i, β_{ii}, and β_{ij} through the got response and corresponding variables which are stored in some columns of the H and denoted as the $H(:,1:(M+1)(M+2)/2+j-1)$. Further we set the symbol W^1 with the $H(1:M,(M+1)(M+2)/2)$. Secondly, we use the Newton optimization for the surface (11) so as to obtain the current optimal corresponding variables $\{\ln(r_q^j)\}_{q=1}^M$. Then we utilize the current optimal weights $\{r_q^j\}_{q=1}^M$ to train the base classifier MultiV-MHKS and thus can get its corresponding validation error cr^j. Thirdly, we set the $H(1:M,(M+1)(M+2)/2+j)$ with $[\ln(r_1^j),\ldots,\ln(r_M^j)]^T$ and $H(M+1,(M+1)(M+2)/2+j)$ with cr^j. Fourthly, through letting $j=j+1$ and $W^j=H(1:M,(M+1)(M+2)/2+j-1)$, we give the terminating condition $||W^j-W^{j-1}||_2<\theta$, where the threshold $\theta\in\mathbb{R}$ is a small positive value. If the terminating condition is not satisfied, the second cycle goes on. If the terminating condition is satisfied, we return the optimal weight $\{r_q^*\}_{q=1}^M$ through setting $[\ln(r_1^*),\ldots,\ln(r_M^*)]^T$ with $H(1:M,(M+1)(M+2)/2+j-1)$.

In the above processing, we adopt MultiV-MHKS as the base classifier. Differently from the original MultiV-MHKS, the weight r_q of each sub-classifier is changed from $1/M$ to an optimized value. Thus the criterion function (6) is rewritten as

$$\min_{\substack{u^q\in\mathbb{R}^{m^q},v^q\in\mathbb{R}^{n^q+1}\\ v_0^q\in\mathbb{R},p=1,\ldots,M}} L=\sum_{q=1}^{M}\left(\sum_{i=1}^{N}(\varphi_i g^q(A_i^q)-1-b_i^q)^2+c^q(u^{qT}S_1u^q+\tilde{v}^{qT}S_2\tilde{v}^q)\right)$$

$$+\gamma\sum_{i=1}^{N}\sum_{q=1}^{M}\left(\varphi_i g^q(A_i^q)-\sum_{p=1}^{M}r_p\varphi_i g^p(A_i^p)\right)^2, \qquad (12)$$

where $r_p\geq 0$ shows the weight of the pth matrix representation and the bigger r_p means that the corresponding matrix representation plays a more important role for classification. We make the gradient of L in the (12) with respect to u^q and v^q be zero respectively and thus get the following results:

$$u^q=\left((1+\gamma(1-r_q)^2)\sum_{i=1}^{N}A_i^q\tilde{v}^q(A_i^q\tilde{v}^q)^T+c^qS_1\right)^{-1}\sum_{i=1}^{N}\left(A_i^q\tilde{v}^q\left(\varphi_i(b_i^q+1)\right.\right.$$

$$\left.\left.-(1+\gamma(1-r_q)^2)v_0^q+\gamma(1-r_q)\sum_{p=1,p\neq q}^{N-1}r_p(u^{pT}A_i^p\tilde{v}^p+v_0^p)\right)\right), \qquad (13)$$

$$v^q=((1+\gamma(1-r_q)^2)Y^{qT}Y^q+c^q\tilde{S_2})^{-1}Y^{qT}$$

$$\times\left(1_{N\times 1}+b^q,+\gamma(1-r_q)\sum_{p=1,p\neq q}^{M}r_pY^pv^p\right). \qquad (14)$$

Further, the gradient of the improved criterion function L in the (12) with respect to b^q is given as follows:

$$\frac{\partial L}{\partial b^q}=-2(Y^qv^q-1_{N\times 1}-b^q). \qquad (15)$$

Based on Eq. (15), we denote the vector b of the qth view at the kth iteration by b_k^q and thus obtain

$$\begin{cases} b_1^q\geq 0\\ b_{k+1}^q=b_k^q+\rho^q(e_k^q+|e_k^q|)\end{cases}, \qquad (16)$$

where at the kth iteration, the vector of the qth view e_k^q is set with $Y_k^qv_k^q-1_{N\times 1}-b_k^q$, and the learning rate of the qth view $0<\rho^q<1$. The termination criterion for the base classifier is designed as

$$\frac{\|L_{k+1}-L_k\|_2}{\|L_k\|_2}\leq\xi, \qquad (17)$$

where the $\xi\in\mathbb{R}$ is a small positive value. Finally, the whole decision function for the proposed RMultiV-MHKS is defined as

$$g(z)=\sum_{q=1}^{M}r_q(u^{qT}Z^q\tilde{v}^q+v_0^q)\begin{cases}>0 & \text{then } z\in\text{class}+1\\ <0 & \text{then } z\in\text{class}-1\end{cases}, \qquad (18)$$

where z is the test sample and Z^q is the qth matrix representation of the z. The whole procedure of the proposed method is summarized in Table 1.

Table 1
Algorithm: RMultiV-MHKS.

Input: The sample set $\{(x_i,\varphi_i)\}_{i=1}^N$;
The initialized parameters: b_1^q, S_1, S_2, γ, c^q, ρ, ξ, e_0, $r_q^0=\frac{1}{M}$, u_1^q, v_1^q, $q=1\ldots M$.
OutPut: The optimal weight $\{r_q^*\}_{q=1}^M$, and the solution to RMultiV-MHKS $\{u^q,\tilde{v}^q,v_0^q\}_{q=1}^M$.
1. Reshape x_i to $\{A_i^q\}_{q=1}^M$ with the defined M ways, where $mn=d$,
Set $Y_1^q=[y_1^q,\ldots,y_N^q]^T$, $y_i^q=\varphi_i[u_1^{qT}A_i^q,1]^T$, $q=1,\ldots,M$;
2. Create the matrix $H\in\mathbb{R}^{(M+1)\times N}$ with each zero element so as to store the $\ln(r_q)$ and R,
define $Lr_q^0=\ln(r_q^0)$, $q=1\ldots M$;
3. Let $k=1$, do
(a). Add or decrease one Δ for Lr_q^0 and form a new known corresponding variables Lr_q^k, $q=1\ldots M$;
(b). Set $H(1:M,k)=[Lr_1^k,\ldots,Lr_M^k]^T$;
(c). Train the base classifier MultiV-MHKS (A_i^q,r_q) and get the response variable cr^k;
(d). Set $H(M+1,k)=cr^k$, $k=k+1$;
until $k>(M+1)(M+2)/2$;
4. Define $W^j\in\mathbb{R}^M$ and set $W^j=H(1:M,(M+1)(M+2)/2),j=1$;
5. Do
(a). Solve the $\beta_0,\beta_i,\beta_{ii},\beta_{ij}$ in the (11) according to $H(:,1:(M+1)(M+2)/2+j-1)$.
(b). Get the current optimal $Lr_q^j,q=1\ldots M$ in the surface (11) through the Newton optimization;
(c). Train MultiV-MHKS (A_i^q,r_q^j) and get the cr^j;
(d). Set $H(1:M,(M+1)(M+2)/2+j)=[Lr_1^j,\ldots,Lr_M^j]^T$ and $H(M+1,(M+1)(M+2)/2+j)=cr^j$;
(e). Set $j=j+1$ and $W^j=H(1:M,(M+1)(M+2)/2+j-1)$;
until $||W^j-W^{j-1}||_2<\theta$;
6. Get the optimal weight $\ln(r_q^*)=H(1:M,(M+1)(M+2)/2+j-1)$;
7. Train MultiV-MHKS (A_i^q,r_q^*) and get the solution $\{u^q,\tilde{v}^q,v_0^q\}_{q=1}^M$.

4. Experiments

In order to validate the feasibility and effectiveness of the proposed RMultiV-MHKS, we compare RMultiV-MHKS with MultiV-MHKS in terms of classification and computational cost. Meanwhile, we carry out the MKDA with SDP [20] and the ℓ_p-MKDA with SIP [22] for comparison. In addition, the other two state-of-the-art multiple kernel algorithms denoted as SVM-2K [37] and MKL [38] respectively, are both compared with our method since multiple kernel learning is supposed to be an effective non-linear learning and also be one kind of multi-view learning machines. Further, we give the discussion for RMultiV-MHKS in terms of: (1) the initialized weight r_q^0; (2) the length of the searching step Δ; (3) the size of the matrixized views M.

4.1. Experimental setting

In the implementation of MultiV-MHKS, the vector b_1^q is initialized to 10^{-6}. The learning rate ρ^q is set to 0.99. Both the regularized parameter c^q and the coupling parameter γ are selected from the set $\{2^{-4},2^{-2},2^0,2^2,2^4\}$. The termination variable ξ is fixed to 10^{-4}. In the proposed RMultiV-MHKS, the initializations for the parameters b^q, ρ^q, c^q, γ, and ξ are given the same as those in MultiV-MHKS. The termination threshold θ is set with 0.02. In practice, we add another termination condition with the maximal number of the iteration $maxIter=50$ so as to avoid the endless loop. The length of the searching step Δ is optimized from the set $\{0.3,0.5,0.8,1.0,2.0\}$. For the compared MKDA with SDP [20], ℓ_p-MKDA with SIP [22], SVM-2K [37], and MKL [38], the used kernels are the polynomial kernel $ker(x_i,x_j)=(x_ix_j+1)^d$ and the Radial Basis Function (RBF) kernel $ker(x_i,x_j)=\exp(-\|x_i-x_j\|_2^2/\sigma^2)$. Here we classify the implemented experiments into the two cases with different M. For $M=2$, the corresponding kernel parameters are set with $d=2$ and $\sigma=\overline{\sigma}$, where $\overline{\sigma}$ is the average value of all the l_2-norm distances $\|x_i-x_j\|_2, i,j=1\ldots N$ as used in [39]. For $M>2$, the corresponding kernel parameters are set with $d=2$, $\sigma=\overline{\sigma}/\ell$, where ℓ is selected from the set $\{0.1,0.01,0.001\}$. The classification performance of all the algorithms implemented here are reported by Monte Carlo Cross Validation (MCCV) [40]. MCCV randomly splits the samples into the two parts including the training and validation sets, and repeats the procedure N times. In our experiments, N is set with 10. The benchmark datasets used here are obtained from [41] and their detail and description are shown in Table 2, where Breast-Cancer-Wisconsin and Contraceptive-Method-Choice are denoted as BCW and CMC for short respectively. The one-against-one classification strategy [42] is adopted for the used multi-class datasets.

The way of generating the views for data is the same as that in MultiV-MHKS [8]. For the single-view patterns $\{z_i\}_{i=1}^N, z_i\in\mathbb{R}^d$, we employ a simple reshaping way to create multiple views. The reshaping way does not involve overlapping between the components of pattern, i.e., the pattern z_i is partitioned into many equal-sized sub-vectors in non-overlapping way, and then reshaped column-by-column as a corresponding matrix. In this way, different sizes of the sub-vectors naturally lead to different matrix representations for the z_i. Therefore mathematically, each pattern $z_i\in\mathbb{R}^d$ can have multiple different matrix representations denoted as $A_i^q\in\mathbb{R}^{m^q\times n^q}, q=1,\ldots,M$, where the value of d is equal to $m^q\times n^q$. For the Wine used here, we remove the last one attribute from the original Wine with 13 attributes so as to produce more assembling matrix representations.

4.2. Classification performance comparison

In this section, we compare RMultiV-MHKS with MultiV-MHKS, MKDA with SDP [20], ℓ_p-MKDA with SIP [22], SVM-2K [37], and MKL [38]. Here we give the experiment for the case $M=2$, which means that the size of different matrix reshaping ways is two for both RMultiV-MHKS and MultiV-MHKS. Correspondingly, the size of the kernels is also two for both the other four implemented multiple kernel algorithms. The weight value for the r_q, $q=1,2$ is set with $1/2$ in MultiV-MHKS, which means that each matrix representation for the original pattern would play the same role into the final classification. It should be stated that since there would be more than two different matrix representations for some used datasets such as Hill-Valley with 2×50, 4×25, 5×20, and 10×10, we choose the matrix representations which would give the first and second best classification accuracies in MatMHKS. Table 3 gives the averaged accuracy and the corresponding standard deviation of all the compared methods on the validation sets generated by the 10-folds MCCV. Table 4 shows the weight values comparison between RMultiV-MHKS and MultiV-MHKS. From Table 3, it can be found that: (1) the classification accuracy of RMultiV-MHKS is better than that of MultiV-MHKS on almost all the used datasets, especially for the datasets Hill-Valley, Pima, Letter, Lenses. The improving performance is attributed to that RMultiV-MHKS can give a more reasonable weight values r_q than MultiV-MHKS, which is clearly shown in Table 4. Taking the dataset Pima for example, RMultiV-MHKS assigns r_1 with 15.45 for the view with the matrix representation 2×4 and r_2 with 2.58×10^{-5} for the view with the 4×2. Consequently, the performance of the RMultiV-MHKS increases by around 10% over MultiV-MHKS with the equal weights $r_1=r_2=1/2$. In this situation, the weight r_2 of RMultiV-MHKS is very small and thus its corresponding matrix view is supposed to supply less discriminative information than the other view. The results validate that RMultiV-MHKS can assign a heavier weight to the favorable view and a lighter even zero weight to the unfavorable view that does not carry discriminative information. In practice, we can remove the useless view. (2) Compared with the MKDA used here, RMultiV-MHKS achieves a better accuracy on more than half of the used datasets though it is just a linear algorithm. Especially for Breast-Cancer-Wisconsin, the accuracy of RMultiV-MHKS has an increasing value with about 37.1% over MKDA with SDP and 40.5% over ℓ_p-MKDA with SIP. (3) Compared with SVM-2K and MKL used here, the proposed algorithm also shows a competitive even better classification accuracy.

Table 2
The description for the used datasets.

Datasets	# of variables	# of class	# of instances
Breast-Cancer-Wisconsin (BCW)	10	2	699
Iris	4	3	150
Hill-Valley	100	2	1212
Pima	8	2	768
Water	38	2	116
Musk	166	2	476
Sonar	60	2	208
Letter	432	10	500
Semeion	256	10	1593
Lenses	4	3	24
Contraceptive-Method-Choice (CMC)	9	3	1473
Secom	590	2	1567
Dermatology	34	6	366
Glass	10	6	214
House-votes	16	2	435
Arrhythmia	279	16	452
Balance	4	3	625
Housing	13	2	506
Ionosphere	34	2	351
Wine	12	3	178

Table 3
Classification accuracy (%) and t-test comparison between RMultiV-MHKS, MultiV-MHKS, MKDA (SDP) [20], ℓ_p-MKDA (SIP) [22], SVM-2K [37], and MKL [38]. (The best accuracy of each dataset is in bold. The p-values are from the t-test comparing each classifier to RMultiV-MHKS. The asterisk * denotes that the difference from RMultiV-MHKS is significant at 5% significance level, i.e. p-value less than 0.05.)

Datasets	RMultiV-MHKS	MultiV-MHKS	SDP	SIP	SVM-2K	MKL
	Accuracy	Accuracy p-value	Accuracy p-value	Accuracy p-value	Accuracy p-value	Accuracy p-value
BCW	**98.49 ± 1.03**	97.32 ± 0.78* 8.46×10^{-4}	61.38 ± 0.00* 1.95×10^{-28}	57.99 ± 1.91* 1.50×10^{-21}	65.62 ± 0.00* 2.01×10^{-33}	59.82 ± 13.13* 2.59×10^{-8}
Iris	**98.66 ± 0.17**	97.73 ± 1.09* 0.0109	96.93 ± 1.78* 0.0014	94.80 ± 3.11* 3.76×10^{-4}	97.67 ± 1.30* 1.79×10^{-54}	96.80 ± 1.43* 6.33×10^{-4}
Hill-Valley	**81.32 ± 2.78**	75.74 ± 0.94* 5.70×10^{-5}	49.50 ± 0.00* 3.23×10^{-20}	63.20 ± 9.15* 1.76×10^{-5}	50.49 ± 0.00* 6.62×10^{-21}	50.61 ± 1.06* 1.35×10^{-14}
Pima	**80.76 ± 1.56**	71.14 ± 0.27* 4.21×10^{-15}	65.10 ± 0.00* 1.20×10^{-17}	71.07 ± 2.00* 8.36×10^{-12}	73.65 ± 1.75 1.41×10^{-9}	75.41 ± 1.94* 1.88×10^{-6}
Water	**98.99 ± 0.45**	96.97 ± 2.71* 0.0490	56.14 ± 0.00* 1.33×10^{-38}	92.63 ± 3.86* 1.54×10^{-6}	84.56 ± 6.76* 3.80×10^{-39}	88.59 ± 5.18* 5.53×10^{-6}
Musk	81.43 ± 1.01	78.51 ± 0.94* 9.64×10^{-6}	56.54 ± 0.00* 4.87×10^{-28}	84.77 ± 2.66* 0.0288	78.61 ± 1.84* 1.81×10^{-32}	**88.23 ± 3.09*** 4.59×10^{-6}
Sonar	79.73 ± 3.33	76.67 ± 3.23* 0.0065	46.60 ± 0.00* 6.26×10^{-18}	**80.00 ± 4.02** 0.6565	71.07 ± 4.26* 7.76×10^{-24}	74.08 ± 5.77* 0.0271
Letter	89.60 ± 1.07	83.20 ± 0.56* 4.24×10^{-6}	91.56 ± 0.89* 4.04×10^{-5}	91.56 ± 0.93* 2.39×10^{-4}	91.40 ± 1.38* 0.0013	**92.72 ± 0.96*** 8.47×10^{-7}
Semeion	90.16 ± 1.71	87.64 ± 1.02* 3.46×10^{-6}	**94.00 ± 1.05*** 7.52×10^{-7}	93.39 ± 1.09* 4.08×10^{-5}	78.34 ± 3.59* 5.98×10^{-9}	93.36 ± 0.00* 0.0237
Lenses	72.73 ± 9.08	53.85 ± 10.90* 0.012	71.82 ± 6.71* 0.0246	**75.45 ± 9.63** 0.4085	63.077 ± 4.86* 0.0075	71.53 ± 6.33 0.2928
CMC	**51.66 ± 2.94**	50.01 ± 1.41 0.1742	46.54 ± 1.61* 1.58×10^{-4}	45.70 ± 1.74* 3.25×10^{-5}	42.68 ± 0.00* 3.58×10^{-6}	46.79 ± 7.87 0.0618
Secom	**93.37 ± 1.82**	92.95 ± 2.17 0.9635	93.36 ± 0.00 0.8561	88.30 ± 0.67* 2.43×10^{-7}	6.63 ± 0.00* 1.36×10^{-35}	93.36 ± 0.00* 0.0237
Dermatology	**97.75 ± 0.76**	97.17 ± 0.96 0.3168	94.51 ± 1.58* 3.74×10^{-7}	96.65 ± 1.38* 0.0017	44.18 ± 10.84* 7.03×10^{-12}	94.34 ± 1.87* 2.68×10^{-5}
Glass	**99.35 ± 0.21**	99.24 ± 0.60 0.7065	94.67 ± 2.92* 2.89×10^{-4}	98.10 ± 1.42* 0.0188	97.90 ± 1.33* 2.18×10^{-47}	96.95 ± 1.94* 0.0013
House-votes	**94.77 ± 2.86**	92.81 ± 1.69 0.1211	38.71 ± 0.00* 6.26×10^{-20}	93.04 ± 2.34 0.2351	91.71 ± 1.83* 5.93×10^{-28}	93.73 ± 1.95* 0.0013
Arrhythmia	60.52 ± 1.43	59.92 ± 1.21 0.0706	61.89 ± 3.34 0.0782	67.75 ± 1.61* 5.79×10^{-10}	63.56 ± 1.07* 1.00×10^{-4}	**68.03 ± 2.49*** 1.24×10^{-7}
Balance	89.23 ± 2.21	88.85 ± 1.02 0.1310	93.43 ± 2.14* 4.13×10^{-4}	**93.78 ± 1.64*** 4.91×10^{-6}	88.08 ± 1.60* 0.0155	89.90 ± 1.15* 0.0400
Housing	92.91 ± 0.61	92.91 ± 0.00 0.3074	**93.25 ± 0.00*** 2.70×10^{-4}	88.45 ± 1.59* 4.02×10^{-9}	93.25 ± 0.00* 0.0167	92.25 ± 0.00 0.2595
Ionosphere	**90.03 ± 2.24**	8 9.33 ± 1.50 0.3335	64.00 ± 0.00* 1.13×10^{-16}	89.09 ± 1.22 0.1674	87.03 ± 2.26* 0.0194	85.66 ± 16.29 0.4078
Wine	95.54 ± 1.49	94.43 ± 1.74 0.2892	93.86 ± 2.69 0.1172	**95.91 ± 1.22** 0.7091	80.00 ± 4.05* 3.45×10^{-30}	92.45 ± 1.75* 9.29×10^{-4}

In order to further discuss the implemented algorithms, we perform the paired t-test [43] by comparing RMultiV-MHKS with the other five algorithms. Doing so can test how significant the classification accuracy changes. The t-test is a statistical test for a null hypothesis H_0. If the null hypothesis H_0 is correct, it would demonstrate that there is no significant difference between the mean number of samples correctly classified by RMultiV-MHKS and the MultiV-MHKS, MKDA with SDP, ℓ_p-MKDA with SIP, SVM-2K, and MKL. Under this assumption, the p-value of each test is the probability of a significant difference in correctness values occurring between the two validation sets. Therefore, the smaller the p-value, the less likely that the observed difference results from the identical validation set correctness distributions. The threshold for the p-value is set to 0.05 in our experiments. Table 3 also shows the p-value. From this table, it can be found that the average classification accuracy of RMultiV-MHKS is superior to that of MultiV-MHKS on more than half of the used datasets.

Table 4
The weight r_1 and r_2 of RMultiV-MHKS and MultiV-MHKS in their two corresponding views.

Datasets	BCW	Iris	Hill-Valley	Pima	Water
View1–View2	2×5–5×2	2×2–4×1	5×20–10×10	2×4–4×2	2×19–19×2
RMultiV-MHKS	1.05–0.55	0.29–7.24×10^{-5}	0.52–0.16	15.45–2.58×10^{-5}	2.13–2.11
MultiV-MHKS	0.5–0.5	0.5–0.5	0.5–0.5	0.5–0.5	0.5–0.5
	Musk	Sonar	Letter	Semeion	Lenses
	166×1–83×2	5×12–6×10	6×72–12×36	4×64–16×16	2×2–4×1
RMultiV-MHKS	0.01–2.43×10^{-5}	0.02–2.43	0.13–2.52	0.41–3.48×10^{-5}	2.06×10^{-2}–3.60
MultiV-MHKS	0.5–0.5	0.5–0.5	0.5–0.5	0.5–0.5	0.5–0.5
	CMC	Secom	Dermatology	Glass	House-votes
	3×3–9×1	10×59–59×10	2×17–17×2	2×5–5×2	2×8–4×4
RMultiV-MHKS	6.28×10^{-3}–39.63	0.28–0.27	0.13–2.29×10^{-4}	1.69–7.94	2.00×10^{-3}–14.52
MultiV-MHKS	0.5–0.5	0.5–0.5	0.5–0.5	0.5–0.5	0.5–0.5
	Arrhythmia	Balance	Housing	Ionosphere	Wine
	3×93– 9×31	2×2–4×1	13×1–1×13	2×17–17×2	2×6–3×4
RMultiV-MHKS	0.51–1.58	1.10–0.49	1.0–1.0	0.48–1.01	0.005–0.5
MultiV-MHKS	0.5–0.5	0.5–0.5	0.5–0.5	0.5–0.5	0.5–0.5

Table 5
Training time (in seconds) comparison between RMultiV-MHKS, MultiV-MHKS, MKDA (SDP) [20], ℓ_p-MKDA (SIP) [22], SVM-2K [37], and MKL [38].

Datasets	RMultiV-MHKS	MultiV-MHKS	MKDA (SDP)	ℓ_p-MKDA (SIP)	SVM-2K	MKL
BCW	22.52	2.81	61.38	1.12	2.71	178.30
Iris	45.27	4.24	4.52	0.07	0.25	57.95
Hill-Valley	1021.29	126.58	225.58	3.48	1.03	430.08
Pima	5.24	0.83	74.68	1.21	5.22	178.30
Water	87.45	9.95	1.49	0.02	0.18	20.76
Musk	420.10	58.31	18.36	1.35	0.56	88.23
Sonar	3.08	0.22	3.41	0.11	0.41	24.91
Letter	1272.31	198.03	48.58	1.46	3.53	909.51
Semeion	1712.62	263.88	903.42	67.28	54.45	1974.44
Lenses	3.42	0.34	0.41	0.02	0.05	28.16
CMC	20.11	2.63	704.41	7.90	21.06	1014.72
Secom	38.93	5.60	469.60	22.30	1.97	830.33
Dermatology	181.01	26.47	27.63	0.49	13.47	623.68
Glass	669.27	99.48	3.93	0.08	1.46	238.89
House-votes	10.13	1.34	17.60	0.55	0.82	52.17
Arrhythmia	6537.40	914.12	52.34	0.70	4.11	1170.63
Balance	51.26	7.13	60.94	0.98	5.87	154.65
Housing	20.51	3.27	28.93	0.33	0.58	48.71
Ionosphere	10.27	1.87	10.84	0.33	0.61	47.77
Wine	34.97	4.33	5.84	0.06	0.64	64.65

4.3. Computational cost comparison

In this section, we give a comparison between the training time of RMultiV-MHKS, MultiV-MHKS, MKDA with SDP, ℓ_p-MKDA with SIP, SVM-2K, and MKL in Table 5. All the computations are run in the same condition that includes Intel® Xeon® 5520 Series processors 2.26 GHz, 6 G RAM DDR3, Windows Server 2008 RC2 and MATLAB environment. It should be stated that for a fair comparison, we report the average training time of the 10-folds MCCV, where the parameters for the compared algorithms are set with the same values as those reported in Table 3. From Table 5, it can be found that: (1) both ℓ_p-MKDA and SVM-2K spend less time than the other four algorithms on most of the used datasets. (2) The training cost of RMultiV-MHKS takes a competitive time to that of MKDA with SDP and MKL. (3) For most of the used datasets, the training time of RMultiV-MHKS costs more than six times than that of MultiV-MHKS, which can be explained through the analysis for the internal structure of RMultiV-MHKS. According to Table 1, the first cycle carries out the base classifier MultiV-MHKS for $(M+1)(M+2)/2$ times. When $M=2$, the $(M+1)(M+2)/2=6$. Moreover, RMultiV-MHKS still needs to carry out the second cycle. Therefore the training cost for RMultiV-MHKS takes more than six times of that of the original MultiV-MHKS, which accords with the experimental results of the $M=2$ shown in Table 5. It should be stated that in the experimental processing, the computation of RMultiV-MHKS takes a larger cost in terms of the first cycle as shown in Fig. 1 while RMultiV-MHKS can get a fast convergence for the second cycle. Therefore our future work aims to introduce a more efficient technique into RMultiV-MHKS so as to decrease its computational cost in terms of the first cycle.

4.4. Further discussion

In the above experiments, we find that some parameters of RMultiV-MHKS play an important role in the performance. Thus in this section we give the further discussion in terms of: (1) the initialized weight r_q^0; (2) the length of the searching step Δ; (3) the number of the matrixized views M. For the discussion about the r_q^0 and the Δ, we select some representative datasets with the maximal size of the instances or the minimal dimensionality. In detail, Semeion is the dataset in which its instances are maximal in all the used datasets. Secom is the dataset in which its dimensionality is maximal. Lenses is the dataset in which its instances are minimal. Arrhythmia is the dataset in which its classes are maximal. Balance is the dataset in which its dimensionality is minimal. Sonar is just a normal dataset. For the discussion on the M, we choose the six datasets including Hill-Valley, House-votes, Letter, Semeion, Sonar, and Wine since the number of their matrix reshaping ways is more than two, i.e. $M>2$.

4.4.1. Analysis for the initialized weight r_q^0

Here we explore how influence the initialized weight r_q^0 plays. We set three different initialized weight r_q^0 with [0.5,0.5], [1, 1], and the random value [0.3,0.7] as the original searching corresponding variables. In order to keep the comparison fair, we require the same experimental setting for the three initializations. Fig. 2 shows the classification accuracy of RMultiV-MHKS comparison for different initialized weight r_q^0 on the six datasets (from left to right): Arrhythmia, Balance, Lenses, Secom, Semeion, and Sonar. From Fig. 2, it can be found that on the used datasets only except Lenses, different initialized weight r_q^0 does not lead to a

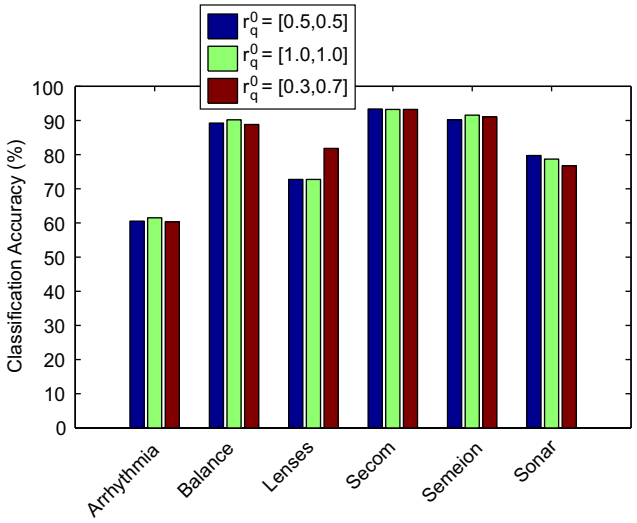


Fig. 2. Classification accuracy (%) of RMultiV-MHKS comparison for different initialized weight r_q^0 s on the six datasets (from left to right): Arrhythmia, Balance, Lenses, Secom, Semeion, and Sonar.

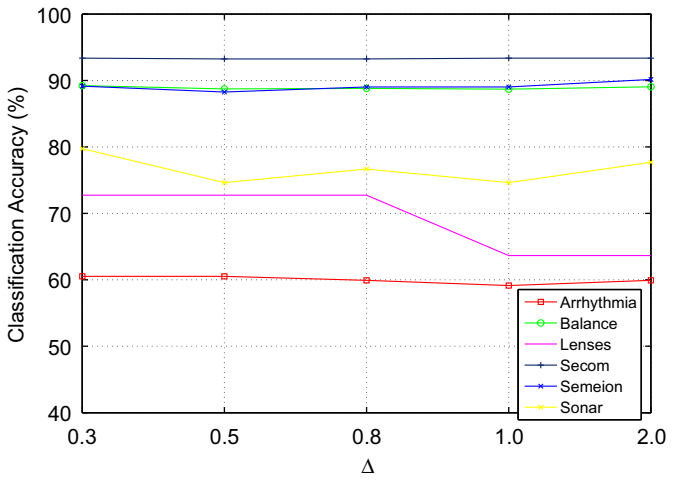


Fig. 3. Classification accuracy (%) of RMultiV-MHKS as a function of the length of the searching step Δ on the datasets: Arrhythmia, Balance, Lenses, Secom, Semeion, and Sonar.

Table 6
The optimal weight r_q^* comparison with different initialized r_q^0.

Datasets	$r_q^0=[0.5,0.5]$	$r_q^0=[1.0,1.0]$	$r_q^0=[0.3,0.7]$
	$r_1^*-r_2^*$	$r_1^*-r_2^*$	$r_1^*-r_2^*$
Arrhythmia	0.51–1.58	0.54–0.98	0.48–1.47
Balance	1.10–0.49	1.00–0.52	1.04–0.56
Lenses	0.02–3.60	0.02–3.78	4.28–8.95
Secom	0.28–0.27	0.29–0.26	0.24–0.21
Semeion	$0.41–3.48\times10^{-5}$	2.89–0.02	3.30–0.06
Sonar	0.02–2.43	0.01–0.98	0.32–3.66

significant influence for the classification accuracy. Especially on the Secom, different initialized weight r_q^0 almost causes an equal value 93.37%. In order to further analyze this phenomenon, we also give the corresponding optimal weight r_q^* for the used datasets in Table 6. From this table, we can find that although it is different for the optimal r_q^* generated from different initialized weight r_q^0, it is similar for the relative relationship between r_1^* and r_2^* on each used datasets, which might be the reason for the similar performance induced from different r_q^0 on most of the used datasets here.

4.4.2. Analysis for the step length Δ

Here we explore the role of the length of the searching step Δ for RMultiV-MHKS. Fig. 3 shows the classification accuracy of RMultiV-MHKS as a function of the Δ on the used datasets including Arrhythmia, Balance, Lenses, Secom, Semeion, and Sonar. The range of the Δ is from 0.3 to 2.0. From Fig. 3, we can find that: (1) the different Δs almost lead to an unchanged classification accuracy on half of the used datasets. Especially on the Secom, we can obtain a flat classification accuracy line. (2) On the Lenses and Sonar, different Δs have some impacts for the performance. Taking Lenses for example, there is a flat change in the range of the Δ from 0.3 to 0.8, but there is a decline in the range from 0.8 to 1.0. For the Sonar, the performance is a fluctuation between 0.3 and 2.0. It is known that the constructed response surface is crucial with the got corresponding variables generated through moving different Δs from the original r_q^0. Different Δs would lead to their corresponding characteristics in terms of constructing the response surface, which means that the Δ has an important impact on the final performance.

4.4.3. Analysis for the size of the views M

In this section, we discuss how role the size of the matrix representations (views) M plays on the performance of RMultiV-MHKS. For each dataset here, we add one matrix view every time to increase the number of views while keeping the previous matrix views unchanged. Specifically, we first arrange the matrix representations in a descend order in terms of classification accuracies of MatMHKS. For $M=2$, we select the matrix representations giving the first and second best classification accuracies of MatMHKS. For $M=3$, we add the new matrix representation corresponding to the third best accuracy of MatMHKS to the previous two views. Taking Hill-Valley as an example, we adopt the first two best matrix representations 5×20, 10×10 for $M=2$, the first three best 5×20, 10×10, 4×25 for $M=3$, and the first four best 5×20, 10×10, 4×25, 2×50 for $M=4$. Fig. 4 gives the classification accuracies of RMultiV-MHKS, MultiV-MHKS, MKDA with SDP, and ℓ_p-MKDA with SIP as a function of the M on the given datasets Hill-Valley, House-votes, Letter, Semeion, Sonar, and Wine.

From Fig. 4, it can be found that the size of the M has an impact on the classification accuracy of all the compared algorithms. First, RMultiV-MHKS always has a superior performance to the original MultiV-MHKS whatever the M is. For the Letter and Hill-Valley, RMultiV-MHKS has a significant improvement by about 6% and 5% in terms of the recognition rate. Secondly, although RMultiV-MHKS is one linear algorithm, it still has a competitive performance to the kernel-based MKDA with SDP and ℓ_p-MKDA with SIP. Thirdly, when the value of the M grows, RMultiV-MHKS can induce a better combination for different views on some used datasets. To be more exact, for the Wine and Semeion, RMultiV-MHKS with $M=3$ has a better performance than that with $M=2$, where RMultiV-MHKS can search and adjust the weight so as to get a better r_q. For the House-votes and Sonar, the performance curve of RMultiV-MHKS changes flat while the M changes. Fourthly, the performance curve of MultiV-MHKS shows that its accuracy gives a downward trend when the M is over three. One possible reason is that some useless views are merged into the final classification when the number of the views increases. Fortunately, the proposed RMultiV-MHKS is supposed to solve this problem since it optimizes the weight r_q, which is validated on most of the used datasets. Moreover, we analyze the weight value

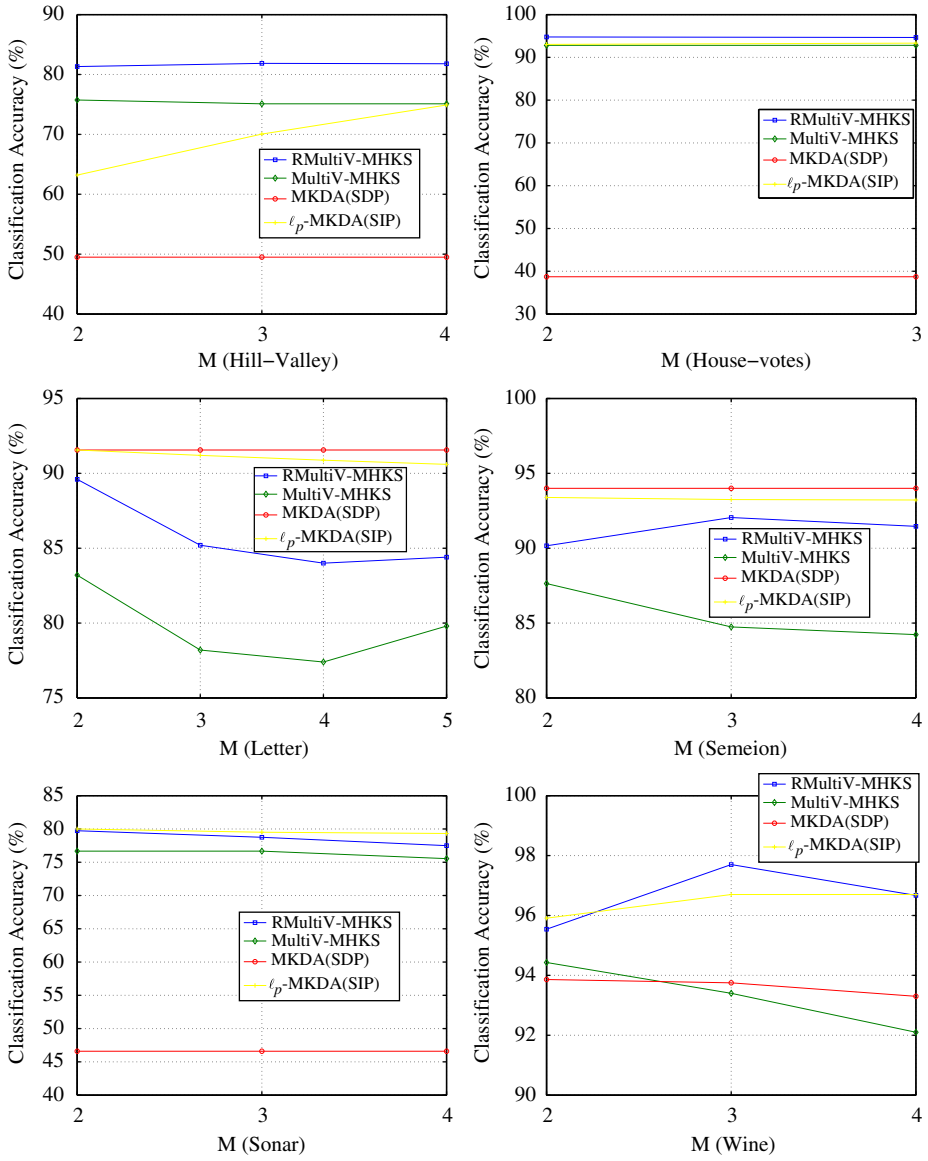


Fig. 4. The classification accuracies (%) of RMultiV-MHKS, MultiV-MHKS, MKDA (SDP) [20] and ℓ_p-MKDA (SIP) [22] as a function of the M on the given datasets: Hill-Valley, House-votes, Letter, Semeion, Sonar, and Wine.

r_q between RMultiV-MHKS, MultiV-MHKS, MKDA with SDP and ℓ_p-MKDA with SIP, and find that RMultiV-MHKS can filtrate some useless views through reducing their weights. Meanwhile, RMultiV-MHKS can also increase the weights of those more informative views so as to maintain the classification accuracy effectively.

5. Rademacher complexity analysis

In this section, we discuss the Rademacher complexity of the proposed RMultiV-MHKS, MHKS, MatMHKS, and MultiV-MHKS. Further we give their relationship in terms of theory and experiments. It is known that the analysis of the generalization risk bound is important for interpreting the performance behavior of one learning algorithm [44–47]. The Rademacher complexity is widely used for measuring the generalization risk bound. The classical risk bound theory was proposed by Vapnik and Chervonenkis [48] and can be described through Theorem 1.

Theorem 1. *Let P be a probability distribution on $\chi \times \{\pm 1\}$ and $\{x_i,y_i\}_{i=1}^n$ be chosen independently according to P. Then for a $\{\pm 1\}$-valued function class $\mathbf{F}$ with the domain χ, there is a constant $c \geq 0$ such that for any integer n, with probability at least $1-\delta$ over $\{x_i,y_i\}_{i=1}^n$, every g in $\mathbf{F}$ satisfies*

$$P(y \neq g(x)) \leq \hat{P_n}(y \neq g(x)) + c\sqrt{\frac{VC(\mathbf{F})}{n}}, \tag{19}$$

where $VC(\mathbf{F})$ denotes the Vapnik–Chervonekis dimension of $\mathbf{F}$ and $\hat{P_n}$ denotes the empirical risk error of the function g on the sample set $\{x_i,y_i\}_{i=1}^n$.

In this case, the VC($\mathbf{F}$) measures the complexity of the $\mathbf{F}$. The Rademacher complexity was proposed as an alternative notion for the complexity of a function class $\mathbf{F}$ [45]. Here, the Rademacher complexity is used to measure the complexity of the proposed RMultiV-MHKS. Definition 1 gives the definition of the Rademacher complexity [45].

Definition 1. Let μ be a probability distribution on a set χ and suppose that $\{x_i\}_{i=1}^n$ are independent samples selected from χ according to μ. Let $\mathbf{F}$ be a class of functions mapping from χ to $\mathbb{R}$. Let $\{\sigma_i\}_{i=1}^n$ be independent uniform $\{\pm 1\}$-valued random variables and define the empirical Rademacher complexity of $\mathbf{F}$ with the random variable

$$\hat{R}_n(\mathbf{F}) = \mathbf{E}\left[\sup_{g\in\mathbf{F}}\left|\frac{2}{n}\sum_{i=1}^n \sigma_i g(x_i)\right| x_1,\ldots,x_n\right], \tag{20}$$

where $\mathbf{E}$ is the operator of the expected value of a random variable. Then the Rademacher complexity of $\mathbf{F}$ is

$$R_n(\mathbf{F}) = \mathbf{E}\hat{R}_n(\mathbf{F}). \tag{21}$$

The following Theorem 2 [49] gives the generalization risk bound of $\mathbf{F}$ with the Rademacher complexity $R_n(\mathbf{F})$.

Theorem 2. *Let P be a probability distribution on $\chi\times\{\pm 1\}$ and $\{x_i,y_i\}_{i=1}^n$ be chosen independently according to P. Then for a $\{\pm 1\}$-valued function class $\mathbf{F}$ with the domain χ, with probability at least $1-\delta$ over $\{x_i,y_i\}_{i=1}^n$, every g in $\mathbf{F}$ satisfies*

$$P(y\neq g(x)) \le \hat{P}_n(y\neq g(x)) + \frac{R_n(\mathbf{F})}{2} + \sqrt{\frac{\ln(1/\delta)}{2n}}. \tag{22}$$

We adopt the $R_n(g_{RMMHKS})$,$R_n(g_{MMHKS})$,$R_n(g_{MatMHKS})$, and $R_n(g_{MHKS})$ to denote the Rademacher complexity of RMultiV-MHKS, MultiV-MHKS, MatMHKS, and MHKS respectively. We firstly discuss the relationship between $R_n(g_{RMMHKS})$,$R_n(g_{MMHKS})$, and $R_n(g_{MHKS})$. Here we give a temp decision function as follows:

$$g'(z) = \sum_{q=1}^M r_q(u^{qT}Z^q\tilde{v}^q + v_0^q)\begin{cases} >0 & \text{then } z\in \text{class}+1 \\ <0 & \text{then } z\in \text{class}-1 \end{cases}, \tag{23}$$

where $r_q \ge 0$ without any other restriction. The decision functions of RMultiV-MHKS and MultiV-MHKS are both the special cases of the function g', where r_q^* in RMultiV-MHKS is optimized through RST and r_q in MultiV-MHKS is set to $1/M$. Therefore, the sets $\{g_{RMMHKS}\}$, $\{g_{MMHKS}\}\subseteq\{g'\}$. According to the definition of the Rademacher complexity, we can get

$$R_n(g_{RMMHKS}),R_n(g_{MMHKS}) \le R_n(g'). \tag{24}$$

On the other hand, r_q^* in RMultiV-MHKS might be optimized to $1/M$ and in this case $R_n(g_{RMMHKS})$ is the same as $R_n(g_{MMHKS})$. Otherwise, it is uncertain for the relationship between $R_n(g_{RMMHKS})$ and $R_n(g_{MMHKS})$.

Further, we give the relationship between $R_n(g')$ and $R_n(g_{MatMHKS})$. It is known that the generalization risk bound of MHKS satisfies the inequality (22). According to the equations (23) and (25)

$$g(A) = u^T A\tilde{v} + v_0, \tag{25}$$

the temp decision function g' is the convex combination of different $g_{MatMHKS}$s. It has be proven that for a class of functions $\mathbf{F}$, if conv $\mathbf{F}$ is the class of convex combinations of function from $\mathbf{F}$ and $-\mathbf{F}=\{-g: g\in\mathbf{F}\}$ [49],

$$R_n(\text{conv}\,\mathbf{F}) = R_n(\mathbf{F}). \tag{26}$$

Concretely, for the sample set $\{x_i\}_{i=1}^n$ and $\{\sigma_i\}_{i=1}^n$,

$$\begin{aligned}&\sup_{g\in\text{conv}\,\mathbf{F}}\left|\sum_{i=1}^n \sigma_i g(x_i)\right| \\ &= \max\left(\sup_{g\in\text{conv}\,\mathbf{F}}\sum_{i=1}^n \sigma_i g(x_i), \sup_{g\in\text{conv}\,\mathbf{F}} -\sum_{i=1}^n \sigma_i g(x_i)\right) \\ &= \max\left(\sup_{g\in\mathbf{F}}\sum_{i=1}^n \sigma_i g(x_i), \sup_{g\in\mathbf{F}} -\sum_{i=1}^n \sigma_i g(x_i)\right) \\ &= \sup_{g\in\mathbf{F}}\left|\sum_{i=1}^n \sigma_i g(x_i)\right|.\end{aligned}$$

Based on the definition of the Rademacher complexity, the following equation can be got:

$$R_n(g') = R_n(g_{MatMHKS}). \tag{27}$$

Finally, we analyze the relationship between $R_n(g_{MatMHKS})$ and $R_n(g_{MHKS})$. According to our previous work [9], it is known that the solution space for the weight vectors of MatMHKS is contained in that of MHKS. And MatMHKS can be viewed as the MHKS imposed by Kronecker product decomposability constraint. As a consequence, the set of functions $\{g_{MatMHKS}\}\subseteq\{g_{MHKS}\}$. According to the definition of the Rademacher complexity, i.e. (20) and (21), we can get

$$R_n(g_{MatMHKS}) \le R_n(g_{MHKS}). \tag{28}$$

Based on the formulas (24), (27) and (28), we finally have the relationship between $R_n(g_{RMMHKS})$, $R_n(g_{MMHKS})$, $R_n(g_{MatMHKS})$, $R_n(g_{MHKS})$ as follows:

$$R_n(g_{RMMHKS}),R_n(g_{MMHKS}) \le R_n(g_{MatMHKS}) \le R_n(g_{MHKS}). \tag{29}$$

In order to clearly show the relationship between $R_n(g_{RMMHKS})$, $R_n(g_{MMHKS})$, $R_n(g_{MatMHKS})$, $R_n(g_{MHKS})$, we further give the empirical Rademacher complexity values according to Eq. (20) with

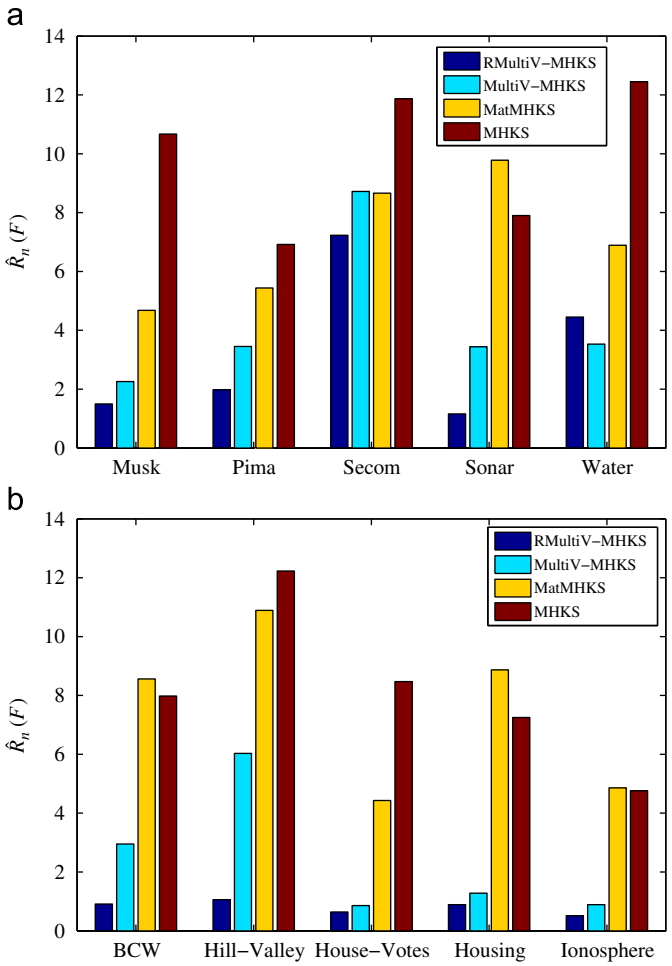


Fig. 5. The empirical Rademacher complexity for RMultiVMHKS, MultiV-MHKS, MatMHKS, and MHKS on the datasets Musk, Pima, Secom, Sonar, Water, BCW, Hill-Valley, House-votes, Housing, and Ionosphere.

experiments. We use the 10 binary-class datasets shown in Table 2. The parameters $\{\sigma_i\}_{i=1}^{n}$ in the (20) are independent uniform $\{\pm 1\}$-valued random variables based on Definition 1. The $\mathbf{F}$ in the (20) can correspond to the class of the decision functions of RMultiV-MHKS, MultiV-MHKS, MatMHKS, and MHKS. For each dataset, we compute the (20) for 10 times and report the average values in Fig. 5. From this figure, it can be found that both the $\hat{R}_n(\mathbf{F})$ of RMultiV-MHKS and MultiV-MHKS are smaller than that of MatMHKS or MHKS. The $\hat{R}_n(\mathbf{F})$ of RMultiV-MHKS gets the smallest values on most of the used datasets. Although MatMHKS yields unstable capability and its $\hat{R}_n(\mathbf{F})$ even achieves the largest ones on Sonar, BCW, Housing, and Ionosphere, it has a lower $\hat{R}_n(\mathbf{F})$ than MHKS on more than half of all the used datasets. Therefore, these experimental results in Fig. 5 are consistent with the theoretical analysis above.

6. Conclusion and future work

In this paper, we change a base classifier into M different sub-classifiers (views), and then implement one joint learning process for the generated M sub-ones, which is named RMultiV-MHKS. Differently from our previous work MultiV-MHKS [8] that treats each view equally, the proposed RMultiV-MHKS adopts the RST to optimize the weight of each view. In doing so, RMultiV-MHKS can distribute the heavier weight to the favored view which can bring more classification information. Simultaneously, it is theoretically and experimentally demonstrated that RMultiV-MHKS has a tighter generalization risk bound than its single-view learning machine MHKS in terms of the Rademacher complexity. The experimental results also validate that the proposed algorithm owns a statistically superior classification accuracy to the original MultiV-MHKS. But on the other hand, we find that RMultiV-MHKS would take a bigger computational cost since it is made up of the two cycles. Thus our future work is to design a more efficient technique so as to decrease the computational cost.

Acknowledgments

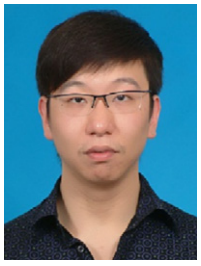
The authors thank Natural Science Foundations of China under Grant no. 60903091, 21176077, and 61170151, the Specialized Research Fund for the Doctoral Program of Higher Education under Grant no. 20090074120003, and Natural Science Foundations of Jiangsu under Grant no. BK2011728 for partial support.

References

[1] V.R. de Sa, Learning classification with unlabeled data, in: Neural Information Processing Systems, 1994.
[2] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: The 33rd Annual Conference of the Association for Computational Linguistics, 1995.
[3] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: The Conference on Computational Learning Theory, 1998.
[4] S. Dasgupta, M. Littman, D. McAllester, Pac generalization bounds for co-training, in: Neural Information Processing Systems, 2001.
[5] M. Collins, Y. Singer, Unsupervised models for named entity classification, in: EMNLP, 1999.
[6] S. Abney, Bootstrapping, in: The 40th Annual Conference of the Association for Computational Linguistics, 2002.
[7] Z. Wang, S.C. Chen, Multi-view kernel machine on single-view data, Neurocomputing 72 (2009) 2444–2449.
[8] Z. Wang, S.C. Chen, D. Gao, A novel multi-view learning developed from single-view patterns, Pattern Recognition 44 (2011) 2395–2413.
[9] S. Chen, Z. Wang, Y. Tian, Matrix-pattern-oriented Ho-Kashyap classifier with regularization learning, Pattern Recognition 40 (2007) 1533–1543.
[10] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
[11] R. Brylla, G.O. Ricardo, F. Queka, Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets, Pattern Recognition 36 (2003) 1291–1302.
[12] R.E. Schapire, The boosting approach to machine learning: an overview, in: D.D. Denison, M.H. Hansen, C. Holmes, B. Mallick, B. Yu (Eds.), Nonlinear Estimation and Classification, Springer, 2003.
[13] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University, 2004.
[14] R. Myers, D. Montgomery, Response Surface Methodology: Process and Product Optimization using Designed Experiments, Wiley-Interscience, 2002.
[15] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, vol. 2, Citeseer, 2001.
[16] J. Leski, Ho-Kashyap classifier with generalization control, Pattern Recognition Lett. 24 (14) (2003) 2281–2290.
[17] V.R. Sa, Spectral clustering with two views, in: The ICML Workshop on Learning with Multiple Views, 2005.
[18] V. Sindhwani, P. Niyogi, M. Belkin, A co-regularization approach to semi-supervised learning with multiple views, in: The ICML Workshop on Learning with Multiple Views, 2005.
[19] M. Becker, B. Hachey, B. Alex, C. Grover, Optimising selective sampling for bootstrapping named entity recognition, in: The ICML Workshop on Learning with Multiple Views, 2005.
[20] S.J. Kim, A. Magnani, S. Boyd, Optimal kernel selection in kernel fisher discriminant analysis, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 465–472.
[21] Fei Yan, J. Kittler, K. Mikolajczyk, A. Tahir, Non-sparse multiple kernel learning for fisher discriminant analysis, in: IEEE International Conference on Data Mining, 2009.
[22] Fei Yan, K. Mikolajczyk, M. Barnard, Hongping Cai, J. Kittler, ℓ_p norm multiple kernel fisher discriminant analysis for object and image categorisation, in: International Conference on Computer Vision and Pattern Recognition, 2010.
[23] F.R. Bach, Consistency of the group Lasso and multiple kernel learning, J. Mach. Learn. Res. 9 (2008) 1179–1225.
[24] Y. Tang, L. Li, X. Li, Learning similarity with multikernel method, IEEE Trans. Syst. Man Cybern. B Cybern. 41 (1) (2011) 131–138.
[25] Yu Zhang, Dit-Yan Yeung, A convex formulation for learning task relationships in multi-task learning, in: Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10), Corvallis, Oregon, 2010, AUAI Press, pp. 733–442.
[26] S. Gaïffas, G. Lecué, Hyper-sparse optimal aggregation Arxiv preprint arXiv:0912.1618, 2009.
[27] C. Cortes, Support vector machine, Learning 20 (3) (1995) 273–297.
[28] T. Evgeniou, M. Pontil, T. Poggio, Regularization networks and support vector machines, Adv. Comput. Math. 13 (1) (2000) 1–50.
[29] P. Zhang, J. Peng, SVM vs regularized least squares classification, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004, vol. 1, IEEE, 2004, pp. 176–179.
[30] L. Wang, X. Wang, J. Feng, On image matrix based feature extraction algorithms, IEEE Trans. Syst. Man Cybern. B Cybern. 36 (1) (2006) 194–197.
[31] Q. Gao, L. Zhang, D. Zhang, J. Yang, Comments on 'on image matrix based feature extraction algorithms', IEEE Trans. Syst. Man Cybern. B Cybern. 37 (5) (2007) 1373.
[32] T. Zhang, B. Fang, Y. Tang, Z. Shang, B. Xu, Generalized discriminant analysis: a matrix exponential approach, IEEE Trans. Syst. Man Cybern. B Cybern. 40 (1) (2010) 186–197.
[33] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Mach. Learn. 46 (1) (2002) 131–159.
[34] M. Momma, K.P. Bennett, A pattern search method for model selection of support vector regression, in: Proceedings of the SIAM International Conference on Data Mining, Citeseer, 2002, p. 50.
[35] M. Gönen, E. Alpaydln, Regularizing multiple kernel learning using response surface methodology, Pattern Recognition 44 (1) (2011) 159–171.
[36] Ben Blum, Michael I. Jordan, David Kim, Rhiju Das, Philip Bradley, David Baker, Feature selection methods for improving protein structure prediction with Rosetta, In: John Platt, Daphne Koller, Yoram Singer, Andrew McCallum (Eds.), *Advances in Neural Information Processing Systems (NIPS) 20*, 2008.
[37] J.D.R. Farquhar, D.R. Hardoon, H. Meng, J. Shawe-Taylor, S. Szedmak, Two view learning: SVM-2K, theory and practice, in: Neural Information Processing Systems, 2005.
[38] S. Sonnenburg, G. Ratsch, C. Schafer, A general and efficient multiple kernel learning algorithm, Adv. Neural Inf. Process. Syst. 18 (2006) 1273.
[39] I. Tsang, A. Kocsor, J. Kwok, Efficient kernel feature extraction for massive data sets, in: International Conference on Knowledge Discovery and Data Mining, 2006.
[40] Q.S. Xu, Y.Z. Liang, Monte Carlo cross validation, Chemometrics Intell. Lab. Syst. 56 (2006) 1–11.
[41] D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, Uci repository of machine learning databases. Available from: 〈http://www.ics.uci.edu/mlearn/MLRepository.html〉, 1998.
[42] U. Krebel. Pairwise classification and support vector machines, in: Advances in Kernel Methods: Support Vector Learning, 1999, pp. 255–268.
[43] T.M. Mitchell, Machine Learning, McGraw-Hill, Boston, 1997.
[44] P. Bartlett, S. Boucheron, G. Lugosi, Model selection and error estimation, Mach. Learn. 48 (2002) 85–113.
[45] V. Koltchinskii, Rademacher penalties and structural risk minimization, IEEE Trans. Inf. Theory 47 (5) (2001) 1902–1914.

[46] V. Koltchinskii, D. Panchenko, Rademacher processes and bounding the risk of function learning, in: E. Gine, D. Mason, J. Wellner (Eds.), High Dimensional Probability, vol. II, 2000, pp. 443–459.
[47] S. Mendelson, Rademacher averages and phase transitions in Glivenko–Cantelli classes, IEEE Trans. Inf. Theory 48 (1) (2002) 251–263.
[48] V. Vapnik, A. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, Theory Probab. Appl. 2 (1971) 264–280.
[49] P. Bartlett, S. Mendelson, Rademacher and Gaussian complexities: risk bounds and structural results, J. Mach. Learn. Res. 3 (2002) 463–482.

Zhe Wang received the B.Sc. and Ph.D. degrees in Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2003 and 2008, respectively. He is now an Associate Professor in Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai, China. His research interests include feature extraction, kernel-based methods, image processing, and pattern recognition. At present, he has several papers with the first author published on some international journals including IEEE Trans. Pattern Anal. and Mach. Intell, IEEE Trans. Neural Networks, Pattern Recognition, etc.



Jin Xu is a graduate student at the Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai, China. His research interests focus on neural computing and pattern recognition.

Songcan Chen received his B.S. degree in mathematics from Hangzhou University (now merged into Zhejiang University) in 1983. In 1985, he completed his M.S. degree in computer applications at Shanghai Jiaotong University and then worked at Nanjing University of Aeronautics and Astronautics in January 1986. There he received a Ph.D. degree in communication and information systems in 1997. Since 1998, as a full-time professor, he has been with the Department of Computer Science & Engineering at Nanjing University of Aeronautics and Astronautics. His research interests include pattern recognition, machine learning and neural computing. In these fields, he has authored or coauthored over 160 scientific peer-reviewed papers.

Daqi Gao received the Ph.D. degree from Zhejiang University, China, in 1996. Currently, he is a Professor in East China University of Science and Technology. He is a member of the International Neural Network Society. He has published over 50 scientific papers. His research interests are pattern recognition, neural networks, and machine olfactory.