Neurocomputing 72 (2009) 2444–2449

Contents lists available at ScienceDirect

Neurocomputing

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

Multi-view kernel machine on single-view data

Zhe Wang [a,b], Songcan Chen [a,*]

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

ARTICLE INFO

Article history:
Received 7 July 2008
Received in revised form
21 October 2008
Accepted 18 November 2008
Communicated by T. Heskes
Available online 3 December 2008

Keywords:
Multi-view learning
Single-view data
Modified Ho–Kashyap algorithm
Kernel alignment
Pattern recognition

ABSTRACT

Existing multi-view learning focuses on the problem of how to learn from data represented by *multiple* independent sets of attributes (termed as multi-view data), and has been proved to bring an excellent performance. However, in general, we have only a *single* set of attributes (termed as single-view data) available. The goal of this paper is to employ the multi-view viewpoint to develop a multi-view kernel machine for such a single-view data. The key of doing so is to associate each learning machine with one kernel, take it as one view and thus form a set of learning machines from their corresponding kernels, as a result, a multi-view kernel machine can be developed by synthesizing them into a single learning framework. Further, in the two-view (two-kernel) case, we explore the relationship between the generalization ability of the proposed kernel machine and its associated kernels, in which with the kernel alignment (KA) as a correlation measure between kernels, it is found that superior performance of the proposed machine results from a weaker correlation between the constitutive kernels. To the best of our knowledge, both the multi-view learning on single-view data and the KA measure used here have not appeared in any literature. In practice, we take the kernel modified Ho–Kashyap with squared (KMHKS) approximation of the misclassification errors as a learning machine to develop a multi-view KMHKS (MultiV-KMHKS) on single-view data.

1. Introduction

The multi-view learning discusses how to well learn the data represented by *multiple* independent sets of attributes (termed as multi-view data). A typical example is to classify web pages, each of which can be represented by either the words on itself (view one) or the words contained in anchor texts of inbound hyperlinks (view two).

The multi-view learning is first proposed by de Sa [7]. De Sa [7] thinks that when labels are not available or too expensive, different sensory modalities can be used to substitute for the labels, where each sensory modality is taken as one view. Then, minimizing the disagreement between the outputs of dealt patterns from the different views is shown to be a sensible approximation to minimizing the classification error in each view. Subsequently, Yarowsky [22] and Blum and Mitchell [3] also show that the multi-view learning can bring better classification performance than the single view. Yarowsky [22] applies multiple views to word sense disambiguation, which bases on the two senses: collocation and discourse. Blum and Mitchell [3] boost the performance of a learning algorithm by co-training distinct views and take the web page classification as an instance. Further, Collins and Singer [4], Dasgupta et al. [6], and Abney [1] develop the multi-view learning, respectively. Collins and Singer [4] construct an objective function, which can explicitly measure the degree of agreement between the different views. Dasgupta et al. [6] give an upper bound on the generalization error of multiple views, which bases on maximizing the agreement-based objective function suggested by Collins and Singer [4]. Abney [1] gives the reason why the multi-view learning successes, i.e., the fact that the disagreement between the outputs from the different views of certain independence can be minimized by using the unlabeled data. Recently, the Workshop on Learning with Multiple Views held at International Conference on Machine Learning (2005) shows a strong interest in learning the multi-view-representation instances at the fields of machine learning including unsupervised learning [16], semi-supervised learning [18], and supervised learning [2]. Especially in [18], it synoptically gives that the framework of the multi-view learning requires two assumptions about the multi-view data: (1) the compatibility assumption that the base algorithms in each view farthest agree on labels of samples and (2) the independence assumption that the views are independent from each other.

It has been proved that the multi-view learning [1,3,4,6,7,13,15,18,22] can bring excellent performance in practice. Simultaneously, it is also found that almost all the existing multi-view learning starts from that data are represented by multiple independent sets of attributes, i.e., requires the multi-view data.

* Corresponding author.
E-mail address: s.chen@nuaa.edu.cn (S. Chen).

0925-2312/$ - see front matter Crown Copyright © 2009 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.11.017

However, in most applications, data are represented by only one set of attributes and not properly separated into several distinct sets of attributes. In other words, from the multi-view viewpoint, the data in such cases are just a single view, which are termed as single-view data. Relative to the multi-view data, the single-view data are more easily and cheaper acquired and also require less storage. Consequently, in this paper, we expect to employ the multi-view technique on the single-view data to design a multi-view learning machine. To the best of our knowledge, such viewpoint has not appeared in any literatures. Concretely, we introduce the multi-view technique to kernel-based learning machines [12,17,20] on the single-view data.

Kernel-based learning machines have succeeded in classification, regression and other pattern recognition applications [8,10,12,17,20]. These machines map the input data $\mathscr{X}$ into a feature space $\mathscr{F}$ by $\Phi:\mathscr{X}\to\mathscr{F}$, and work in the feature space such as constructing linear learning machines in $\mathscr{F}$ instead of the nonlinear counterparts in $\mathscr{X}$. The mapping Φ is introduced by a kernel, which determines the geometrical structure of the mapped data in the feature space and plays a crucial role. In our method, given the single-view data, we associate each kernel-based learning machine with one kernel, take each such kernel machine as one view, and form a set of views with multiple kernels. Then, by using a single synthetical rather than separate learning framework for each view, a multi-view kernel machine for these individual views is developed, which on the one hand minimizes the empirical risks in each view, and on the other hand minimizes the disagreement between the outputs from the derived views.

Further, we analyze the effectiveness of the proposed multi-view kernel machine on the single-view data. In the multi-view-data case, the effectiveness of the multi-view learning is proved under the two assumptions (compatibility and independence) [13,15,18]. The compatibility assumption guarantees that the difficulty of the problem may be relaxed by the constraint of learning on compatible views, and the independence assumption guarantees that it is unlikely that the algorithms in independent views agree on an incorrect result [18]. But, due to the single-view data used here, the independence assumption on the multi-view data cannot be directly moved to our method. At the same time, it is clear that multiple kernels used in our method are the key that results in different views. So, we focus on kernels and explore how the given multiple kernels influence the performance of our method. Here, we adopt the kernel alignment (KA) [5] as a correlation measure between kernels to analyze the effectiveness of the multi-view kernel machine, and finally find that the superior performance results from a weaker correlation between the used kernels, which is experimentally demonstrated in the two-view (two-kernel) case.

In our implementation, we take the kernel modified Ho–Kashyap with squared (KMHKS) approximation of the misclassification errors [9] as the paradigm, and develop the multi-view KMHKS (MultiV-KMHKS). The experimental results demonstrate its promising performance.

The rest of this paper is organized as follows. Section 2 introduces the multi-view viewpoint to kernel-based learning machines in detail, and then develops a multi-view kernel machine (MultiV-KMHKS) based on the KMHKS. Experiments in Section 3 validate the superior performance of MultiV-KMHKS to KMHKS, and gives the experimental-based convergence proof of MultiV-KMHKS. Following that, we further analyze the effectiveness of the proposed machine, and give the condition under which the multi-view kernel machine has better performance. Finally, both conclusion and future work are given in Section 5.

2. Multi-view kernel machine

According to so-acquired data for an object, they can be sorted into: single-view data and multi-view data. Correspondingly, in our opinion, learning machines can also be sorted into: the single-view machines with only one machine architecture and the multi-view machines with multiple architectures. Naturally, there are four combinations: the single-view machines on the single-view data, the single-view machines on the multi-view data, the multi-view machines on the single-view data, and the multi-view machines on the multi-view data. In traditional machine learning [12,17,20], the single-view machines on the single-view data are widespread. In the multi-view learning [1,3,4,6,7,13,15,18,22] as discussed in Section 1, it is shown the superiority of the single-view machines on the multi-view data to the single-view machines learnt from any individual view. Then, due to the advantages of the single-view data in acquisition cost and storage compared with the multi-view data, in this paper, we develop a multi-view kernel machine especially for the single-view data. To the best of our knowledge, the idea has not appeared in any literature. Such a development can in principle apply to any existing kernel machines, thus without loss of generality, we just select the learning machine KMHKS [9] as the paradigm mainly due to its similar principle to support vector machine (SVM) [20] of maximizing the separating margin and superior classification performance.

2.1. KMHKS approximation of the misclassification errors

Suppose that there are N training samples $(\mathbf{x_i},y_i)$, $i=1,\ldots,N$, where $\mathbf{x_i}\in\mathbb{R}^d$ and the corresponding class label $y_i\in\{+1,-1\}$. For a sample $\mathbf{x}\in\mathbb{R}^d$ to be classified, the decision function of KMHKS is given as

$$f(\mathbf{x})=\operatorname{sign}\left(\sum_{i=1}^{N}y_i\alpha_i k(\mathbf{x},\mathbf{x_i})+w_0\right), \tag{1}$$

where $\alpha_i,w_0\in\mathbb{R}$, and $k(\mathbf{x},\mathbf{x_i})$ is an arbitrary kernel function. By defining the vectors $\mathbf{\Gamma}=[\alpha_i]_{i=1}^N\in\mathbb{R}^{N\times1}$, $\mathbf{Y}=[y_i]_{i=1}^N\in\mathbb{R}^{N\times1}$ and the kernel matrix $K=[y_iy_jk(x_i,x_j)]_{i,j=1}^N\in\mathbb{R}^{N\times N}$, KMHKS tries to obtain $\alpha_i,i=1,\ldots,N$ and the bias w_0 such that

$$K\mathbf{\Gamma}+w_0\mathbf{Y}-\mathbf{1}\geqslant\mathbf{0}, \tag{2}$$

where $\mathbf{1}$ and $\mathbf{0}$ denote the vectors of dimension $N\times1$ with all entries equal to 1 and 0, respectively. To solve the inequalities system of (2), (2) is replaced by the linear equation system

$$K\mathbf{\Gamma}+w_0\mathbf{Y}-\mathbf{1}=\mathbf{b}, \tag{3}$$

where $\mathbf{b}\in\mathbb{R}^{N\times1}\geqslant\mathbf{0}$. Consequently, KMHKS gives the following minimization:

$$\min_{\substack{\mathbf{\Gamma}\in\mathbb{R}^N\\ \mathbf{b}\geqslant\mathbf{0}\\ w_0\in\mathbb{R}}} J(\mathbf{\Gamma},\mathbf{b},w_0)=(K\mathbf{\Gamma}+w_0\mathbf{Y}-\mathbf{1}-\mathbf{b})^{\mathrm{T}}(K\mathbf{\Gamma}+w_0\mathbf{Y}-\mathbf{1}-\mathbf{b})+c\mathbf{\Gamma}^{\mathrm{T}}K\mathbf{\Gamma}, \tag{4}$$

where the second term of the right-handed side of (4) is a regularization term, and the regularization parameter $c\geqslant0$. By differentiating (4) with respect to $\mathbf{\Gamma},w_0$ and equating the results to zero, KMHKS uses the gradient descent method to get $\mathbf{\Gamma},w_0$. Due to the condition $\mathbf{b}\geqslant\mathbf{0}$ in each iteration, KMHKS sets

$$\mathbf{b}_{k+1}=\mathbf{b}_k+\rho(\mathbf{e}_k+|\mathbf{e}_k|), \tag{5}$$

where k denotes the iteration index, $0<\rho<1$ is a parameter, $\mathbf{e}_k=K\mathbf{\Gamma}_k+w_{0k}\mathbf{Y}-\mathbf{1}-\mathbf{b}_k$ is called the error vector, and $|\mathbf{e}_k|$ returns the

vector such that each element of the vector is the absolute value of the corresponding element of $\mathbf{e}_k$. The detailed description about KMHKS is in [9].

2.2. Multi-view KMHKS (MultiV-KMHKS)

In the multi-view learning machine on the single-view data, suppose that there are N labeled training samples $\{(\mathbf{x_i}, y_i)\}_{i=1}^N$ available, and M kernel functions $\{k^p(\mathbf{x_i}, \mathbf{x_j})\}_{p=1}^M$ taken. Each kernel function $k^p(\mathbf{x_i}, \mathbf{x_j})$ is associated with one KMHKS, which owns one classifier architecture (view). Consequently, multiple kernel functions $\{k^p(\mathbf{x_i}, \mathbf{x_j})\}_{p=1}^M$ correspond to multiple kernel matrices $\{K^p\}_{p=1}^M$ with respect to $\{(\mathbf{x_i}, y_i)\}_{i=1}^N$, and further induce multiple views KMHKSs. In such a set of views, there are a set of solutions $\{\mathbf{\Gamma}^p, w_0^p\}_{p=1}^M$, correspondingly. A natural idea is to learn the solution set $\{\mathbf{\Gamma}^p, w_0^p\}_{p=1}^M$ such that for a given sample, each individual KMHKS, respectively, based on the corresponding kernel matrix correctly classifies it, and the disagreement between the outputs of all KMHKSs is farthest minimized. This suggests the following objective function of MultiV-KMHKS:

$$\begin{aligned}
&\min_{\substack{\mathbf{\Gamma}^p\in\mathbb{R}^N\\ \mathbf{b}^p\geqslant\mathbf{0}\\ w_0^p\in\mathbb{R}}} J'(\mathbf{\Gamma}^p, \mathbf{b}^p, w_0^p)\\
&\quad=\sum_{p=1}^{M}((K^p\mathbf{\Gamma}^p + w_0^p\mathbf{Y} - \mathbf{1} - \mathbf{b}^p)^{\mathrm{T}}(K^p\mathbf{\Gamma}^p + w_0^p\mathbf{Y} - \mathbf{1} - \mathbf{b}^p)\\
&\qquad + c^p\mathbf{\Gamma}^{p\mathrm{T}}K^p\mathbf{\Gamma}^p) + \gamma\sum_{p=1}^{M}\left(K^p\mathbf{\Gamma}^p + w_0^p\mathbf{Y} - \sum_{j=1}^{M}\mu_j(K^j\mathbf{\Gamma}^j + w_0^j\mathbf{Y})\right)^{\mathrm{T}}\\
&\qquad\times\left(K^p\mathbf{\Gamma}^p + w_0^p\mathbf{Y} - \sum_{j=1}^{M}\mu_j(K^j\mathbf{\Gamma}^j + w_0^j\mathbf{Y})\right),
\end{aligned}\tag{6}$$

where $\mathbf{b}^p$, c^p are, respectively, the error vector and the regularization parameter of each view, γ is the coupling parameter that regularizes multiple views towards the compatibility using the multiple kernel matrices $\{K^p\}_{p=1}^M$ on a given single-view data set, $\mu_j \geqslant 0$, $\sum_{j=1}^M \mu_j = 1$, μ_j denotes the importance of the corresponding view and the bigger the μ_j is, the more important the corresponding view is. The first term of the right side of (6) is to guarantee each view can correctly classify the samples, and the second one is to minimize the disagreement between each view by making the output of each view be maximally close to the weight average output of all views.

By differentiating (6) with respect to $\mathbf{\Gamma}^p, w_0^p$ and equating the results to zero, we obtain

$$\begin{aligned}
&((1+\gamma)K^{p\mathrm{T}}K^p + c^pK^p)\mathbf{\Gamma}^p + (1+\gamma)K^{p\mathrm{T}}\mathbf{Y}w_0^p\\
&\quad= K^{p\mathrm{T}}\left(\mathbf{1} + \mathbf{b}^p + \gamma\sum_{j=1}^{M}\mu_j(K^j\mathbf{\Gamma}^j + w_0^j\mathbf{Y})\right),
\end{aligned}\tag{7}$$

$$(1+\gamma)\mathbf{Y}^{\mathrm{T}}K^p\mathbf{\Gamma}^p + (1+\gamma)\mathbf{Y}^{\mathrm{T}}\mathbf{Y}w_0^p = \mathbf{Y}^{\mathrm{T}}\left(\mathbf{1} + \mathbf{b}^p + \gamma\sum_{j=1}^{M}\mu_j(K^j\mathbf{\Gamma}^j + w_0^j\mathbf{Y})\right).\tag{8}$$

Then, considering that K^p is positive semi-definite and defining the matrix

$$\Lambda = \begin{bmatrix} (1+\gamma)K^p + c^pI & (1+\gamma)\mathbf{Y} \\ (1+\gamma)\mathbf{Y}^{\mathrm{T}}K^p & (1+\gamma)\mathbf{Y}^{\mathrm{T}}\mathbf{Y} \end{bmatrix},$$

Table 1
Algorithm **MultiV-KMHKS**.

Input: $\{(\mathbf{x_i}, y_i)\}_{i=1}^N$; M kernel matrices $\{K^p\}_{p=1}^M$.
OutPut: $\{\mathbf{\Gamma}^p, w_0^p\}_{p=1}^M$.
1. Initialize $\mathbf{\Gamma}_1^p, w_{01}^p, \mathbf{b}_1^p \geqslant 0,\ p = 1, \ldots, M$ at random;
let $k = 1$;
2. Do until the termination criterion (12) is satisfied:
—(a) For $p = 1 \ldots M$:
——i. Compute $\mathbf{\Gamma}_{k+1}^p, w_{0k+1}^p$ with (9);
——ii. Set $\mathbf{b}_{k+1}^p$ with (11);
—(b) Compute J'_{k+1} with (6);
—(c) Increment k.
3. Return the final $\{\mathbf{\Gamma}^p, w_0^p\}_{p=1}^M$.

(7) and (8) are converted into

$$\begin{bmatrix} \mathbf{\Gamma}_{k+1}^p \\ w_{0k+1}^p \end{bmatrix} = \Lambda^{-1}\begin{bmatrix} \mathbf{1} + \mathbf{b}_k^p + \gamma\left(\sum_{j=1}^{p-1}\mu_j(K^j\mathbf{\Gamma}_{k+1}^j + w_{0k+1}^j\mathbf{Y}) + \sum_{j=p}^{M}\mu_j(K^j\mathbf{\Gamma}_k^j + w_{0k}^j\mathbf{Y})\right) \\ \mathbf{Y}^{\mathrm{T}}\left(\mathbf{1} + \mathbf{b}_k^p + \gamma\left(\sum_{j=1}^{p-1}\mu_j(K^j\mathbf{\Gamma}_{k+1}^j + w_{0k+1}^j\mathbf{Y}) + \sum_{j=p}^{M}\mu_j(K^j\mathbf{\Gamma}_k^j + w_{0k}^j\mathbf{Y})\right)\right) \end{bmatrix},\tag{9}$$

where the subscript k denotes the iteration index. The gradient of (6) with respect to $\mathbf{b}^p$ is given as follows:

$$\nabla_{\mathbf{b}^p}J' = -2(K^p\mathbf{\Gamma}^p + w_0^p\mathbf{Y} - \mathbf{1} - \mathbf{b}^p).\tag{10}$$

In order to keep the condition $\mathbf{b}^p \geqslant 0$ in each view, we start with $\mathbf{b}_1^p \geqslant 0$, refuse to decrease any of its components like KMHKS [9], and give the update of $\mathbf{b}^p$ as follows:

$$\begin{cases} \mathbf{b}_1^p > 0, \\ \mathbf{b}_{k+1}^p = \mathbf{b}_k^p + \rho^p(\mathbf{e}_k^p + |\mathbf{e}_k^p|), \end{cases}\tag{11}$$

where at the kth iteration, the error vector of the pth view $\mathbf{e}_k^p = K^p\mathbf{\Gamma}_k^p + w_{0k}^p\mathbf{Y} - \mathbf{1} - \mathbf{b}_k^p$, and the learning rate of the pth view $0 < \rho^p < 1$. In practice, the termination criterion can be designed as

$$\frac{\|J'_{k+1} - J'_k\|}{\|J'_k\|} \leqslant \xi,\tag{12}$$

where $\xi \in \mathbb{R}$ is a small positive value, and $\|\cdot\|$ is chosen to be L_2 norm throughout the paper. Such designed procedure is just the MultiV-KMHKS and summarized in Table 1.

The decision function of MultiV-KMHKS for the sample $x \in \mathbb{R}^d$ is given as follows:

$$g(x) = \mathrm{sign}\left(\sum_{p=1}^{M}\sum_{i=1}^{N}\mu_p(y_i\alpha_i^p k^p(\mathbf{x}, \mathbf{x_i}) + w_0^p)\right),\tag{13}$$

where $\mathbf{\Gamma}^p = [\alpha_i^p]_{i=1}^N$.

Remark. First, it can be found that in Algorithm *MultiV-KMHKS*, the update of $\mathbf{\Gamma}_{k+1}^p$, w_{0k+1}^p is determined by $\{\mathbf{\Gamma}_{k+1}^j, w_{0k+1}^j\}_{j=1}^{p-1}$ and $\{\mathbf{\Gamma}_k^j, w_{0k}^j\}_{j=p}^M$ as in (9), which reflects that each view *cooperates*. Second, it can be further discovered that if $M = 1, \gamma = 0$ of (6), MultiV-KMHKS is degenerated to KMHKS and so KMHKS is the special instances of MultiV-KMHKS.

3. Experiments

In this section, the multi-view kernel machine MultiV-KMHKS is compared with KMHKS, where the associated kernel functions are Gaussian RBF (RBF) $k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\|\mathbf{x_i} - \mathbf{x_j}\|^2/2\sigma^2)$ and Polynomial (POLY) (POLY) $k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i^T x_j} + 1)^d$, respectively. The d of POLY is set to the prior value 3, and the σ of RBF is set to $\tau\bar{\sigma}$, where the $\bar{\sigma}$ is set to the average of all the pairwise distances $\|\mathbf{x_i} - \mathbf{x_j}\|, i,j = 1,\ldots,N$ between patterns as used in [19] and the τ values are given in the corresponding methods (Tables 2 and 3). Without any prior knowledge, the parameter $\mu_i, i = 1,\ldots,M$ of MultiV-KMHKS is set to $1/M$, i.e., each view owns the same importance. The coupling parameter of MultiV-KMHKS γ is determined by searching from 10^{-5} to 10^4 with each step by multiplying 10. The classification performances of all the classifiers here are reported by Monte Carlo cross validation (MCCV) [21], i.e., randomly split the samples into two parts (the training and validation sets), and repeat the procedure N times. In our experiments, N is set to 10. Benchmark data sets used here are obtained from [14]. All computations are run on Pentium IV 2.80 GHz processor running Windows 2000 Terminal and MATLAB environment.

3.1. Classification performance comparison

To investigate the classification performance of the proposed MultiV-KMHKS, it is compared with its single-view classifier KMHKS. Two cases are designed here: the first one is MultiV-KMHKS with the two *heterogeneous* kernels, i.e., RBF and POLY; the second one with the two *homogeneous* RBF kernels with different spread parameter σ's. For each data set, the accuracies of the two classification strategies on the validation sets generated by the 10-folds MCCV are averaged and reported in Tables 2 and 3, respectively, where for the different methods, the best results are in bold. In addition to reporting the average accuracies, we perform the paired t-test [11] by comparing MulitV-KMHKS with KMHKS. The null hypothesis H_0 demonstrates that there is no significant difference between the mean number of samples correctly classified by MulitV-KMHKS and KMHKS. Under this assumption, the p-value of each test is the probability of a significant difference in correctness values occurring between two validation sets. Thus, the smaller the p-value, the less likely that the observed difference results from identical validation set correctness distributions. The threshold for p-value is set to 0.05. Consequently, from this table, it can be found that the average classification accuracy of MultiV-KMHKS is superior to that of KMHKS on all data sets only except House-votes with the first case. Further, based on the p-value, MultiV-KMHKS has significantly different accuracies from KMHKS on Pima-diabetes and Balance.

Table 2
Average validation accuracy (%) and p-values comparison between MultiV-KMHKS and KMHKS.

Data set	MultiV-KMHKS (RBF + POLY) Accuracy	KMHKS (RBF) Accuracy p-Value	KMHKS (POLY) Accuracy p-Value
Pima-diabetes	**65.97**	50.36* 8.3022e−6	50.67* 8.5446e−06
Sonar	**80.74**	77.78 0.1268	79.54 0.444
Balance	**96.31**	91.12* 6.5558e−10	95.32 0.1333
House-votes	92.53	**92.94** 0.469	91.31 0.1472

Notes: The p-values are from a t-test comparing each classifier to MultiV-KMHKS. The best accuracy results of each data set are in bold. An asterisk * denotes that the difference from MultiV-KMHKS is significant at 5% significance level, i.e. p-value less than 0.05. The σ of RBF is set to $\tau\bar{\sigma}$, where $\tau = 1$ as used in [19].

Table 3
Average validation accuracy (%) and p-values comparison between MultiV-KMHKS and KMHKS.

Data set	MultiV-KMHKS (RBF1 + RBF2) Accuracy	KMHKS (RBF1) Accuracy p-Value	KMHKS (RBF2) Accuracy p-Value
Pima-diabetes	**71.37**	51.69* 2.9472e−13	50.91* 1.2033e−13
Sonar	**71.57**	65.56 0.0697	64.44 0.0517
Balance	**89.29**	87.50* 0.0023	87.79* 0.0044
House-votes	**88.01**	75.11* 1.4177e−11	67.10* 5.8673e−13

Notes: The p-values are from a t-test comparing each classifier to MultiV-KMHKS. The best accuracy results of each data set are in bold. An asterisk * denotes that the difference from MultiV-KMHKS is significant at 5% significance level, i.e. p-value less than 0.05. The parameters σs of RBF1 and RBF2 are set to $\tau\bar{\sigma}$, where $\tau_1 = 0.1$ and $\tau_2 = 10$ respectively.

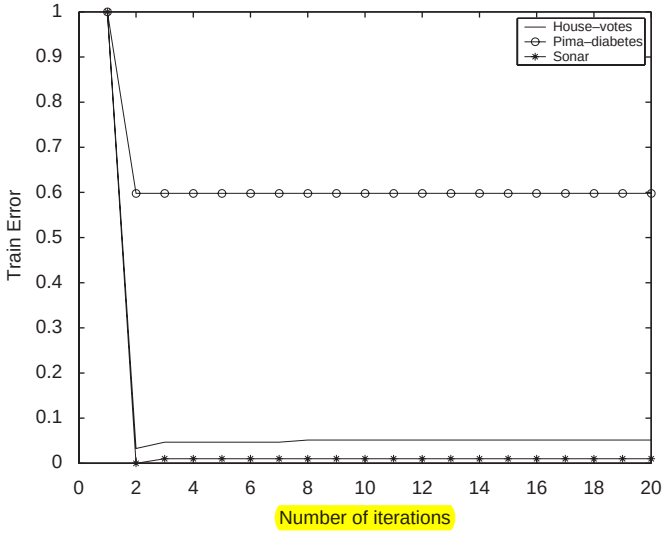

Fig. 1. Convergence of MultiV-KMHKS on training data.

3.2. Experimental-based convergence

We give a discussion on the convergence of the proposed MultiV-KMHKS. Here, due to some difficulty in theoretical proof [9], we adopt an empirical validation to demonstrate that MultiV-KMHKS can converge in the limited iterations. Fig. 1 shows the changes of the training error with the iteration number of MultiV-KMHKS on the binary-class data sets: House-votes, Pima-diabetes, and Sonar, respectively. From the figure, it can be found that all the training errors on these data sets can obviously converge to stable values. Especially, less than 10 iterations are usually enough to achieve convergence.

4. Performance of MultiV-KMKHS depending on its constitutive kernels (views)

As discussed above, the multi-view learning on the multi-view data [13,15,18] does well under the two assumptions (compatibility and independence) satisfied well. In the proposed

multi-view kernel machine here, the compatibility between multiple kernels (views) on the single-view data can be guaranteed by the objective function (6), where the second term is to minimize the disagreement between each view by making the output of each view maximally close to the weight average output of all the views. Simultaneously, due to only the single-view data available in our method, the independence on the multi-view data [13,15,18] cannot be utilized. On the other hand, in our method, although any given multiple kernels including different types or parameters can be used to gives birth to a set of views, it does not necessarily guarantee for the corresponding multi-view kernel machine to give better performance, such as in Table 2, compared to KMHKS with only one RBF kernel, MultiV-KMHKS with the two heterogeneous kernels (views) has the slightly worse classification performance on the data set House-votes, which may boil down to the correlation between the so-adopted two kernels (views) and thus urges us to analyze the influence of such correlation on performance of our method. Here, the KA [5] is a good correlation measure between kernels. Its definition is given as follows:

Definition. [*Kernel Alignment*, *Cristianini* [5]]. The alignment or correlation between the kernels k_1 and k_2 with respect to the samples is

$$A(K_1,K_2)=\frac{\mathrm{tr}(K_1^{\mathrm{T}}K_2)}{\sqrt{\mathrm{tr}(K_1^{\mathrm{T}}K_1)\mathrm{tr}(K_2^{\mathrm{T}}K_2)}}, \tag{14}$$

where K_i is the corresponding kernel matrix for the samples with k_i, and $\mathrm{tr}(\cdot)$ is a matrix trace operation.

The KA can be taken as the cosine of the angle between the kernel matrices, it satisfies $-1\leqslant A(K_1,K_2)\leqslant 1$. In our method, due to the positive semi-definite property of K_i, $0\leqslant A(K_1,K_2)\leqslant 1$. Intuitively, the bigger the value of $A(K_1,K_2)$ is, the more correlated the kernels are. When $A(K_1,K_2)=1$, $K_1=\alpha K_2, \alpha\in\mathbb{R}$. Consequently, according to A, we define three kinds of correlations between kernels: uncorrelation ($A=0$), weaker correlation ($0<A<0.5$), and correlation ($0.5\leqslant A\leqslant 1$).

Further, we take MultiV-KMHKS using the two homogeneous RBF kernels with two different spread parameter σ's as an example. Here $\sigma=\tau\bar{\sigma}$, where $\bar{\sigma}$ is the average of all the pairwise distances $\|\mathbf{x_i}-\mathbf{x_j}\|$, $i,j=1,\ldots,N$ on the training on the training set. In such a case, the one τ is set to 0.1, and the other τ varies from the set $\{0.1,0.2,\ldots,1,2,\ldots,10\}$. The average accuracies of MultiV-KMHKS with the corresponding A are plotted in Fig. 2. From the figure, although the corresponding curves of all the used data sets are not identical, there seems a common trend. For Balance, House-votes, and Sonar, the performance curves of MultiV-KMHKS all exhibit a single peak, i.e., the curves rapidly climb at the beginning, then to the peak point, and finally slowly descend with A approaching to 1, where the A's corresponding to the peak points all fall into $(0,0.5)$. For Pima-diabetes, its performance curve begins with a peak point, and then also descends with A approaching to 1, where its A of the peak point also falls into $(0,0.5)$ similarly. Consequently, in our experiments, it can be found that a weaker correlation between the constitutive kernels (views) can result in superior performance, but the correlation or uncorrelation cannot. Intuitively, it cannot do well when the constitutive kernels (views) degenerate to the same one, which has been demonstrated in Tables 2 and 3. However, here the uncorrelation between the kernels (views) does not bring any performance gain, which seems not to quite accord with the expected benefit from the independence assumption in the usual multi-view learning, thus deserves a further study.

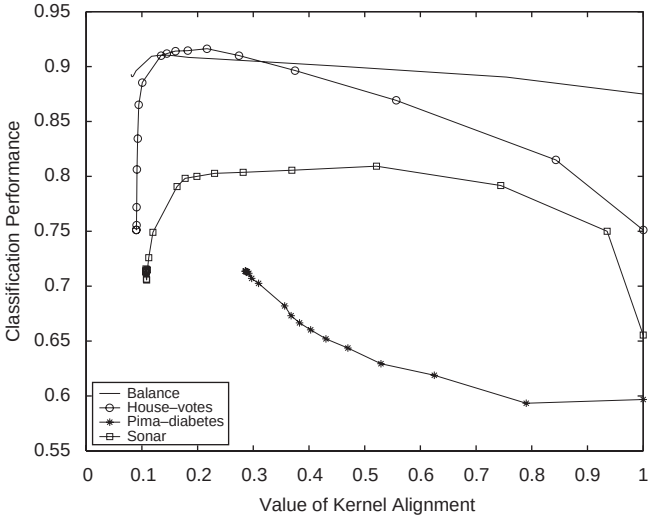

Fig. 2. Average classification accuracy of MultiV-KMHKS with value of KA under homogeneous RBF kernels.

5. Conclusion and future work

In this paper, our contributions are: (1) we develop a multi-view kernel machine, which is established through utilizing multiple kernels on the single-view data; (2) we employ the KA as a correlation measure between the constitutive kernels (views) of our method, and give that the superior performance results from a weaker correlation between the constitutive kernels (views), which is experimentally demonstrated in the two-view case. In the future, we will further explore (1) how the three kernels or more influence the proposed multi-view kernel machine; (2) how the multi-view learning machines work on the multi-view data.

Acknowledgments

The authors thank Natural Science Foundations of China under Grant Nos. 60773061 and 60603029, Jiangsu Natural Science Foundation under Grant No. BK2008381 for partial supports, respectively.

References

[1] S. Abney, Bootstrapping, in: The 40th Annual Conference of the Association for Computational Linguistics, 2002.
[2] 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.
[3] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: The Conference on Computational Learning Theory, 1998.
[4] M. Collins, Y. Singer, Unsupervised models for named entity classification, in: EMNLP, 1999.
[5] N. Cristianini, A. Elisseef, J. Shawe-Taylor, On kernel-target alignment, in: Advances in Neural Information Processing Systems, 2001.
[6] S. Dasgupta, M. Littman, D. McAllester, Pac generalization bounds for co-training, in: Neural Information Processing Systems, 2001.
[7] V.R. de Sa, Learning classification with unlabeled data, in: Neural Information Processing Systems, 1994.
[8] F. Lauer, G. Bloch, Incorporating prior knowledge in support vector machines for classification: a review, Neurocomputing 71 (7–9) (2008) 1578–1594.
[9] J. Leski, Kernel Ho–Kashyap classifier with generalization control, Int. J. Appl. Math. Comput. Sci. 14 (1) (2004) 53–61.
[10] Q. Liu, H. Jin, X. Tang, H. Lu, S. Ma, A new extension of kernel feature and its application for visual recognition, Neurocomputing 71 (10–12) (2008) 1850–1856.
[11] T.M. Mitchell, Machine Learning, McGraw-Hill, Boston, 1997.

[12] K.R. Muller, S. Mika, G. Ratsch, K. Tsuda, B. Scholkopf, An introduction to kernel-based learning algorithms, IEEE Trans. Neural Networks 12 (2) (2001) 181–202.
[13] I. Muslea, C. Kloblock, S. Minton, Active + semi-supervised learning = robust multi-view learning, in: The International Conference on Machine Learning, 2002.
[14] 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.
[15] K. Nigam, R. Ghani, Analyzing the effectiveness and applicability of co-training, in: Information and Knowledge Management, 2000.
[16] V.R. Sa, Spectral clustering with two views, in: The ICML Workshop on Learning with Multiple Views, 2005.
[17] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, Cambridge, 2004.
[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] I. Tsang, A. Kocsor, J. Kwok, Efficient kernel feature extraction for massive data sets, in: International Conference on Knowledge Discovery and Data Mining, 2006.
[20] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[21] Q.S. Xu, Y.Z. Liang, Monte Carlo cross validation, Chemometrics Intell. Laboratory Syst. 56 (2001) 1–11.
[22] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: The 33rd Annual Conference of the Association for Computational Linguistics, 1995.

Zhe Wang received the B.Sc. and Ph.D. degrees in Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2003 and 2008, respectively.

He is now a Lecturer in Department of Computer Science and Engineering, East China University of Science and Technology (ECUST), Shanghai, China. His research interests include machine learning, pattern recognition, and image processing.

Songcan Chen received his B.Sc. degree in mathematics from Hangzhou University (now merged into Zhejiang University), Hangzhou, China, in 1983, M.Sc. degree in computer applications from Shanghai Jiaotong University, Shanghai, China, in 1985, and Ph.D. degree in communication and information systems from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1997.

Since 1998, he has been a Full Professor at the Department of Computer Science and Engineering. He has authored or coauthored over 140 peer-reviewed scientific journal papers. His research interests include pattern recognition, machine learning, and neural computing.