Pattern Recognition 44 (2011) 2395–2413

Contents lists available at ScienceDirect

Pattern Recognition

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

A novel multi-view learning developed from single-view patterns

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

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

ARTICLE INFO

Article history:
Received 3 September 2009
Received in revised form
6 January 2011
Accepted 5 April 2011
Available online 14 April 2011

Keywords:
Multi-view learning
Classifier design
Rademacher complexity
Ensemble learning
Ho–Kashyap classifier
Regularization learning
Pattern recognition

ABSTRACT

The existing multi-view learning (MVL) learns how to process patterns with multiple information sources. In generalization this MVL is proven to have a significant advantage over the usual single-view learning (SVL). However, in most real-world cases we only have single source patterns to which the existing MVL is unable to be directly applied. This paper aims to develop a new MVL technique for single source patterns. To this end, we first reshape the original vector representation of single source patterns into multiple matrix representations. In doing so, we can change the original architecture of a given base classifier into different sub-ones. Each newly generated sub-classifier can classify the patterns represented with the matrix. Here each sub-classifier is taken as one view of the original base classifier. As a result, a set of sub-classifiers with different views are come into being. Then, one joint rather than separated learning process for the multi-view sub-classifiers is developed. In practice, the original base classifier employs the vector-pattern-oriented Ho–Kashyap classifier with regularization learning (called MHKS) as a paradigm which is not limited to MHKS. Thus, the proposed joint multi-view learning is named as MultiV-MHKS. Finally, the feasibility and effectiveness of the proposed MultiV-MHKS is demonstrated by the experimental results on benchmark data sets. More importantly, we have demonstrated that the proposed multi-view approach generally has a tighter generalization risk bound than its single-view one in terms of the Rademacher complexity analysis.

1. Introduction

It is well-known that it is important to integrate the prior knowledge of dealt patterns in designing classifiers [8]. In practice, patterns can generally be obtained from single or multiple information sources. If each information source is taken as one view, accordingly there are two kinds of patterns, i.e. single-view patterns and multi-view patterns.[1] Correspondingly, the learning based on single-view and multi-view patterns can be called as single-view learning (SVL) and multi-view learning (MVL), respectively. It has been proven that co-training as one typical MVL approach has a superior generalization ability to SVL [9]. Co-training learns on both labeled and unlabeled pattern sets. Both labeled and unlabeled patterns are composed of two naturally split attribute sets. Each attribute set is called one view of the patterns. In implementation, co-training algorithm requires that the two views given the class labels are conditionally independent. The independence assumption is guaranteed by the patterns composed of two naturally split attribute sets.

In this paper, we expand the existing MVL to single-view patterns and thus develop a novel MVL framework, whose underlying motivations are:

- It is known that patterns can be sorted into single-view patterns and multi-view patterns according to the number M of information sources [9–11]. However, in most real-world applications there are usually only single-view patterns available since the M has to be one. In that case, the existing MVL framework cannot effectively work since there is not any natural way to partition the attribute space [8,10–12]. Therefore, this fact motivates us to develop a new MVL framework. The new MVL is expected to create multiple different views from single-view patterns and then to learn on the generated views simultaneously.
- In the existing MVL framework, multi-view patterns are represented by multiple independent sets of attributes. Its base algorithms have the same architecture in each view so as to iteratively bootstrap each other. Here, we expect to utilize the multi-view technique due to its superior generalization to the SVL. However, different from the exist MVL on multi-view patterns, we give a new multi-view viewpoint for a given base classifier on single-view patterns. Concretely, we change the original architecture of the given base classifier and thus obtain a set of sub-classifiers with different architectures from

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

[1] Each information source can induce one attribute set for patterns. Thus, single-view patterns are generally composed of one single attribute set and multi-view patterns are generally composed of multiple attribute sets.

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.04.002

each other. Each derived sub-classifier can be taken as one view of the original base classifier, which forms a set of sub-classifiers with multiple views. For all the derived sub-classifiers, we further adopt a joint rather than separated learning process. Therefore, one new learning algorithm is developed for these multi-view sub-classifiers. It is minimized for the disagreement between the outputs of each derived classifier on the same patterns.

In practice, we select the vector-pattern-oriented linear classifier as the so-discussed base classifier. Before being classified, any pattern whatever form it originally is, should be transformed into a vector representation in the vectorial case [33]. However, it is not always efficient to construct a vector-pattern-oriented classifier since the vectorization for patterns such as images might lead to a high computation and a loss of spatial information [21,23,26,34,40]. For overcoming the disadvantage, we proposed a matrix-pattern-oriented Ho–Kashyap classifier named MatMHKS [21,40] in the previous work. MatMHKS is a matrixized version of the vectorial Ho–Kashyap classifier with regularization learning (namely MHKS) [20]. The literature [21,23,34,40] has demonstrated the significant advantages of the matrixized classifier design in terms of both classification and computational performance.

The discriminant function of the vectorial MHKS is given as

$$g(x)=\tilde{\omega}^T x+\omega_0, \tag{1}$$

where $x\in\mathbb{R}^d$ is a vector pattern, $\tilde{\omega}\in\mathbb{R}^d$ is a weight vector, and $\omega_0\in\mathbb{R}$ is a bias. Correspondingly, the discriminant function of MatMHKS is given as

$$g(A)=u^T A\tilde{v}+v_0, \tag{2}$$

where $A\in\mathbb{R}^{m\times n}$ is a matrix pattern, $u\in\mathbb{R}^m$ and $\tilde{v}\in\mathbb{R}^n$ are the two weight vectors, and $v_0\in\mathbb{R}$ is a bias. It is found that for a given pattern, there can be one vector-form representation in the formulation (1) but multiple matrix-form representations with different dimensional sizes for the m and n in the formulation (33). In other words, there are multiple ways for reshaping the vector to the matrix. For instance, a vector $x=[1,2,3,4,5,6,7,8]^T$ could be assembled into two different matrices:

$$\begin{bmatrix}1&3&5&7\\2&4&6&8\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}1&2&3&4\\5&6&7&8\end{bmatrix}^T.$$

Consequently, only one MHKS can be created for classifying the given pattern x. In contrast, multiple MatMHKSs can be created for the same task due to multiple reshaping ways from a vector to a matrix. Therefore, for the same classification problem, the solution set $\{\tilde{\omega},\omega_0\}$ of single MHKS corresponds to the solution sets $\{u^p,\tilde{v}^p,v_0^p\}_{p=1}^M$ of multiple MatMHKSs, where the weight vector sets $\{u^p,\tilde{v}^p\}_{p=1}^M$ are different from each other in terms of the dimensional size but can share a common discriminant function form $g(A)=u^T A\tilde{v}+v_0$. Here, MHKS is viewed as the base classifier. Each MatMHKS is taken as one view of the base MHKS. Our previous work [21] has validated that each MatMHKS provides one hypothesis and exhibits one representation of the original pattern. Thus multiple MatMHKSs can provide a complementarity for each other in classification due to their different representations for patterns. In order to achieve the complementarity, we syncretize the learning processing of multiple MatMHKSs into one single processing. In this case, each MatMHKS is expected to correctly classify one given pattern with the same attributes. Meanwhile it should be minimized for the disagreement between the outputs of all MatMHKSs. As a result, the single learning process is produced and one multi-view-combined classifier named MultiV-MHKS is proposed. Through the Rademacher complexity analysis, we demonstrate that the proposed multi-view MultiV-MHKS has a tighter generalization risk bound compared with the single-view MHKS.

- The proposed MultiV-MHKS algorithm is a nice way to solve the view selection problem of MatMHKS [21]. In MatMHKS, it is always a problem to select the best right matrix-form reshaped from a given vector pattern. This paper suggests one way to bypass it through choosing all the relevant ones and optimizing over them jointly. It is known that from a vector pattern as the input of MHKS to a matrix as the input of MatMHKS, the classification performance of MatMHKS relies on the different reshaping or matrixization ways [21,40]. In the processing of matrixizing a vector, different reshaping ways can induce multiple matrix patterns with different dimensional sizes of the row and column. Consequently, different reshaping ways result in different classification performances of MatMHKSs on the same vector patterns. Then for the best performance, we have to make a choice in multiple reshaping ways with the cross-validation technique at the cost of high computation [21]. Since the proposed MultiV-MHKS here simultaneously considers multiple MatMHKSs with multiple matrices, the choice of matrixizing ways could be avoided to great extent.
- The proposed MultiV-MHKS algorithm adopts the data representation in multiple views different from the other main strategies for creating good ensembles of classifiers: sampling either pattern sets or attribute (interchangeably feature) sets [13,14,48]. Compared with sampling pattern sets or feature sets, the proposed multi-view classifier design provides an alternative novel approach of producing multiple data sets for base learners, i.e. reshaping a vector pattern to different matrix ones with the same full features. In this case, the proposed multi-view classifier has the advantages in terms of the actual number of the unique samples, the size of the feature set and the representations, which brings up the superior performance of the proposed MVL here. In addition, different from the strategy of sampling pattern sets or feature sets, the proposed MVL employs a joint optimization rather than a separate learning in the training processing. To the best of our knowledge, it is novel for the proposed strategy of generating multiple training data sets on the base classifier. The implemented experimental results here have also shown that the proposed classifier MultiV-MHKS algorithm has a superior classification performance to the other strategies of ensembles.

We highlight the contributions of this paper as follows:

- Significance: This paper introduces the creation of multiple views from a single view for multi-view learning. It is known that though the existing MVL has been shown effective in the literature [8,10–12], it still relies heavily on the naturally separating the feature set into two independent components. In many settings, there might not be any natural way to partition the feature space, and thus the existing MVL framework might not be applicable. In such a scenario, the proposed approach suggested in this paper can potentially create multiple independent or at least weaker correlated views from a single view and then learn from the generated multiple views simultaneously.
- Novelty in the two aspects: In the first aspect, the learning approach proposed in this paper is different from the existing multi-view learning approach. Instead of the classifiers trained on two views iteratively boot-strapping each other, this paper proposes a joint learning approach that minimizes the

disagreement across the classifications with multiple views. There might be some similarities with ensemble learning. In ensemble learning, the predictions from different sub-classifiers over a single view are combined. But, in contrast, the critical difference of the proposed MVL here is in the joint optimization. In the second aspect, compared with the typical ensemble models: Bagging or Boosting based on pattern sampling [13,48] and Attribute Bagging based on attribute sampling [14], our strategy is neither sampling patterns nor sampling attributes, instead reshaping the original pattern set to the matrix pattern set in multiple times. Each reshaping can develop a corresponding sub-classifier and then can be syncretized together, which leads to a performance gain.

- Generalization: The proposed MVL is a wrapper technique and is not restricted to the MHKS classifier. It acts as the state-of-the-art *kernelization* technique applied to linear algorithms. The proposed multi-view-combined learning can fall into the framework as follows:

$$\min L = J_{ind} + \gamma J_{com} \tag{3}$$

where $J_{ind} = \sum_{p=1}^{M} I_p(f_p)$, $J_{com} = \sum_{p=1}^{M}(f_p - \sum_{q=1}^{M} r_q f_q)$, $\sum_{q=1}^{M} r_q = 1$. J_{ind} denotes that M learning machines $f_p, p = 1, \ldots, M$ train according to the criterion I_p, respectively. J_{com} makes M machines f_p corresponding to M views of the common labels achieve as much agreement on their outputs as possible. J_{com} tries to achieve the complementarity between M learning machines f_p's. When the individual machine f_p adopts the classifier g of Eq. (33) in practice, the learning framework becomes the proposed algorithm MultiV-MHKS.

The rest of this paper is organized as follows. Section 2 discusses the related work on the multi-view learning. Section 3 gives how to create multiple pattern representations from single-view patterns. Section 4 introduces the multi-view viewpoint into MHKS and MatMHKS, and further gives the description about the structure of the proposed multi-view-combined classifier MultiV-MHKS. The experiments in Section 5 have demonstrated the feasibility and effectiveness of the proposed MVL. Following that, both conclusion and future work are given in Section 6.

2. Related work

One typical example of the existing MVL is web-page classification [9], where each web page can be represented by either the words on itself (view one) or the words contained in anchor texts of inbound hyperlinks (view two). In [9], Blum and Mitchell design a co-training algorithm for the labeled and unlabeled web pattern sets composed of two naturally split views. On labeled web set, two sub-classifiers of co-training algorithm are incrementally built with their corresponding views, and thus on each cycle each sub-classifier labels the unlabeled webs and picks those with the highest confidence into the labeled set. The processing repeats until the terminated condition is satisfied. The co-training algorithm requires the assumptions: (1) the sufficiency that each base classifier should be sufficient to classify the data correctly, (2) the independence assumption that the different views given the class label are conditionally independent, (3) the compatibility assumption that the base classifiers in each view farthest agree on labels of web patterns. But in most cases, it is hard to satisfy the independence assumption due to the non-existence of naturally split attribute sets (i.e. naturally split views) such as only single-view patterns available. Nigam and Ghani [10] experimentally explore the co-training algorithm with or without the independence assumption, demonstrate that the co-training algorithm with a natural split of the attributes outperforms the ones without, and further propose a probabilistic multi-view algorithm co-EM. Moreover, Muslea et al. [11] incorporate active learning into co-EM and present a new method co-EMT. Co-EMT outperforms both co-training and co-EM and has a robustness in view-correlation cases to some extent. Although both co-EMT and co-EM have a superior generalization to co-training, all these algorithms cannot effectively work on the patterns with the non-naturally split attributes, especially the single-view patterns.

It should be stated that the existing MVL [9–11] focuses on semi-supervised learning and works on both labeled and unlabeled patterns. It is known that unlabeled patterns are much easier to obtain in real-world learning applications than labeled ones. Thus, several researches are done to exploit the role of unlabeled patterns. Brefeld et al. [50] show that unlabeled data can significantly improve the predictive performance of classification algorithms and further propose an efficient semi-supervised least squares regression algorithm that scales linearly in the number of unlabeled patterns through a semi-parametric variant. Zhou [51] gives a more general discussion on unlabeled patterns and shows the reasons that unlabeled patterns can effectively work with either few or many labeled patterns. In contrast, the proposed MVL here is supervised rather than semi-supervised and thus pays more attention to labeled patterns available. Meanwhile, paying more attention on agreement of the data is intuitively understood. For example, the same face from the two different views (i.e. different cameras) should share the same label, implying such two-view input face must have the same identity of that person. In the proposed MVL framework, we still employ the assumptions of the existing MVL framework onto labeled patterns. Firstly, the proposed MVL adopts MatMHKS as the base classifier. MatMHKS has been demonstrated to classify patterns correctly with labeled training set [21]. Thus, the sufficiency assumption can be guaranteed. Secondly, the proposed MVL reshapes the same vector patterns into different matrix representations which are different from each other in the representation level. But different matrix representations all correspond to one unique pattern and thus are independently given the class label on the labeled pattern set. Thirdly, the proposed MVL requires not only the compatibility between the sub-classifier of each view and a labeled pattern set, but also the minimization of the disagreement among all views. In a word, the proposed MVL falls into supervised learning framework.

Further, since the proposed MVL on single-view data sets here generates different sub-classifiers from multiple matrix representations, it could be naturally associated with ensemble learning [13–18,43,47,48]. The MVL usually combines the generated sub-classifiers into one learning process in which all the outputs of the sub-classifiers are expected to maximally agree with each other. In contrast, an ensemble of sub-classifiers works by separably running a base sub-classifier multiple times, and forming a final decision with a combination of the outputs of the individual sub-classifiers, where all the outputs are expected to achieve a large diversity of prediction errors. Ensemble learning is known as an effective method of boosting classification performance on single-view patterns. It also combines multiple sub-classifiers separately trained on a given data set. Valentini and Masulli [15] give an overview of ensemble algorithms so as to distinguish generative and non-generative algorithms. The generative ensemble generates a set of base sub-learners, whereas the non-generative one confines themselves and combines a set of existing base sub-learners. But in [15], the authors do not give an unified theory on ensembles. Thus, Seewald [16] proposes a theoretical framework for the field of ensemble learning, where some common ensemble learning schemes can be reduced into the Stacking strategy. Further, Kuncheva [47] gives how to combine sub-classifiers

together in order to achieve an improved classification performance across-the-board. Here, we pay attention to the special ensemble scheme that generates a set of base sub-learners acting on a set of given patterns since our proposed multi-view-combined classifier also works on a set of given patterns. The ensemble method acting on the original patterns is categorized into two main schemes: sampling patterns and features. Bagging [13], AdaBoosting [48] and Attribute Bagging [14] are their typical instances. For the sampling patterns technique, this paper focuses on Bagging that works by randomly sampling M times from the original training set with replacement and generating M new training sets. As the instance of the sampling features technique, Attribute Bagging works by randomly selecting multiple subsets of features from the original feature set without replacement.

There are two differences between our method and Bagging. First, the proposed approach adopts the so-called *multiviewization* that reshapes the original training set represented with single vector into the sets represented with multiple matrices. Thus, the original single-view training set can induce multiple training sets represented with different matrices. In this case, we not only keep the size of the original training set, but also hope to induce some representation information. In contrast, Bagging randomly samples the original training set multiple times with replacement. In this sense, some examples from the original training set are repeated in the newly generated training set. Thus, the actual number of unique pattern decreases though the size remains the same. Secondly, the proposed approach adopts the joint learning on the generated training sets. Bagging adopts the separate learning, i.e. the majority voting technique. In a word, our proposed method works by producing multiple different matrix representations from the original vector representation and joint learning over the representations. This is an entirely different but novel approach of producing multiple data sets for base classifiers. In order to explore the differences between them, we have made an experimental comparison in Section 5.

3. Creating multiple pattern representations from single-view patterns

3.1. The way of multiviewization

This section gives the way to generate multiple pattern representations from the single-view patterns. Firstly, according to the size of sources for patterns, we sort patterns into single-view and multi-view patterns. In our opinion, each source of a pattern can form a set of attributes for the pattern. Thus, each set of attributes can be taken as one view of the pattern. Then suppose that there are patterns $\{z_i\}_{i=1}^N$, where each pattern z_i has M views denoted as $\{z_i^p\}_{p=1}^M$ and each view of all the patterns denoted as $\{z_i^p\}_{i=1}^N$ is independent from each other. If $M=1$, the patterns $\{z_i^p\}_{p=1}^M$ is called the single-view patterns. If $M \geq 2$, the patterns $\{z_i^p\}_{p=1}^M$ is called the multi-view patterns.

As Section 2 stated, in traditional machine learning [45,46], the learning machines on the single-view patterns is widespread and there are usually only single-view patterns available in most cases. Correspondingly, in the multi-view learning framework [9–11], the superiority of the machines has been shown on the multi-view patterns to the machines learnt from any individual view. Thus, due to the superior performance of the learning on multi-view patterns to single-view patterns, we develop a so-called *multiviewization* technique for the single-view patterns.

In practice, we consider that the single-view patterns $\{z_i\}_{i=1}^N$ are originally represented by vector i.e. $z_i \in \mathbb{R}^d$, which is the most familiar case. In order to create multiple views from $\{z_i\}_{i=1}^N$, we define a simple reshaping way for the given single-view patterns $\{z_i\}_{i=1}^N$. Through the defined reshaping way, each pattern z_i can be represented in multiple matrices. In other words, the original-vector patterns $z_i \in \mathbb{R}^d$ can be reshaped into multiple matrices in different reshaping ways. For convenience, the defined reshaping way is without overlapping among the components of the pattern, i.e. the original-vector z_i is partitioned into many equal-size sub-vectors, and then arranged column-by-column into the corresponding matrix as shown in Fig. 1. In this case, different sizes of the sub-vector generate different matrix representations of z_i. Therefore in mathematics, each pattern $z_i \in \mathbb{R}^d$ can have multiple matrix representations denoted as $A_i^p \in \mathbb{R}^{m_p \times n_p}$, $p=1, \ldots, M$, where the value of d is equal to that of $m_p \times n_p$. We call the reshaping processing as *multiviewization*.

3.2. Advantages of multiviewization

In the proposed *multiviewization* processing, the original vector pattern z_i is reshaped into the matrix $A_i^p \in \mathbb{R}^{m_p \times n_p}$ in each view. The literatures [26,23,21,40,36] have given the advantages of using the matrix rather than vector representation for patterns. Firstly, the two weight vectors $u, \tilde{v}$ of Eq. (33) respectively acting on the two sides of the matrix pattern A_p replace the original single weight vector $\tilde{\omega}$ of Eq. (1) on z_i. Thus, the memory required for the weight vectors of the linear classifier is reduced from $d = m_p \times n_p$ to $m_p + n_p$. Secondly, Chen et al. [36] have demonstrated that reshaping the vector pattern into the matrix one can introduce some new implicit information through the new constraint in structure. Thirdly, the discriminant function (33) of each matrix view is the discriminant function (1) of the original vector case imposed with Kronecker product [41] decomposability constraint [40]. The searching for its optimal weight vectors $u, \tilde{v}$ of each matrix view might be guided by some prior information such as the structural or locally spatial information which is reflected in the representation of the Kronecker production of u and $\tilde{v}$. That is the reason why using the matrix rather than vector representation for patterns can improve in the classification performance especially for image patterns. Since the proposed *multiviewization* adopts matrix representation for patterns, it inherits all the listed advantages above.

Moveover, the proposed *multiviewization* is a wrapper technique, since the *multiviewization* as Fig. 1 shows is not limited into

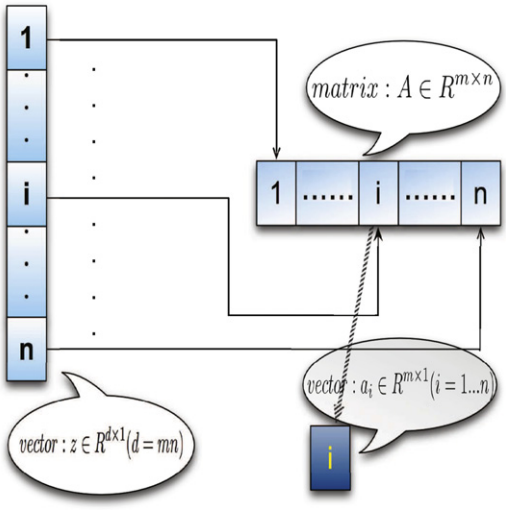

Fig. 1. Reshape vector $z \in \mathbb{R}^{d \times 1}$ to a matrix $A \in \mathbb{R}^{m \times n}$, where $d=mn$ and the i-th sub-vector $a_i \in \mathbb{R}^{m \times 1}, i=1, \ldots, n$.

one special learning algorithm and can be extended to the learning framework with the formulation (1). The proposed *multiviewization* processing creates multiple projections of the original training set. Thus, the size of the original training set is still kept the same. Comparison with the two current typical combined classifiers Bagging based on pattern sampling [13] and Attribute Bagging based on attribute sampling [14] both for generation of classifier diversity, our strategy is neither sampling patterns nor sampling attributes. Instead, the proposed *multiviewization* reshapes a whole pattern set to matrix pattern set and works with the representation level of the original single-view patterns. Therefore, the proposed *multiviewization* is an alternative approach for improving classification like Bagging [13] and Attribute Bagging [14].

4. Joint learning on multiple views

This section describes how to learn on the multiple pattern representations generated from the single-view patterns. We first review the base classifier MHKS [20] and its corresponding matrixization version MatMHKS, respectively. Based on both MHKS and MatMHKS, we further propose a multi-view-combined classifier namely MultiV-MHKS.

4.1. MHKS classifier

The vector linear classifiers [25] own the discriminant function written as the formulation (1), and have attracted more and more attentions due to their ease of mathematical tractability. Among these classifiers, the Ho–Kashyap (HK) classifier [19] is well-known as its simplicity and fast gradient descent optimization with a heuristic update-rule. Subsequently, some modifications have been done in the HK classifier. MHKS [20] is a regularized least squares (RLS) [42,44] classifier. It adopts a similar principle to support vector machine (SVM) [22], maximizes the separating margin without solving the quadratic programming (QP) problem, and gets better generalization performance than the original HK classifier in their experiments. This subsection reviews the architecture of MHKS.

Suppose that there are N samples $(x_i,\varphi_i), i=1,\ldots,N$, where $x_i \in \mathbb{R}^d$ and the corresponding class label $\varphi_i \in \{+1,-1\}$. The discriminant function of the HK classifier for the binary classification problem is denoted as the formulation (1). If the binary classification problem is linearly separable, we can use (1) to get the formulation in the following form:

$$\varphi_i g(x_i) = \varphi_i(\tilde{\omega}^T x_i + \omega_0) > 0, \quad i=1,\ldots,N. \tag{4}$$

Further, by defining the augmented pattern vector $y_i = [x_i^T, 1]^T$, the corresponding augmented weight vector can be given by $w = [\tilde{\omega}^T, \omega_0]^T \in \mathbb{R}^{d+1}$. Thus (4) can be rewritten as

$$\varphi_i g(y_i) = \varphi_i \omega^T y_i > 0, \quad i=1,\ldots,N. \tag{5}$$

Let $y_i = \varphi_i y_i$, $Y = [y_1,\ldots,y_N]^T$, then, (5) can be denoted in matrix form

$$Y\omega > 0_{N\times 1}. \tag{6}$$

The criterion function of the HK classifier is to minimize the quadratic loss function as follows:

$$J_s(\omega,b) = \|Y\omega - b\|_2^2 = (Y\omega - b)^T(Y\omega - b), \tag{7}$$

where b is the margin vector, and $b \geq 0_{N\times 1}$. The gradients of J_s with respect to ω and b are respectively given by

$$\nabla_\omega J_s = 2Y^T(Y\omega - b), \tag{8}$$

$$\nabla_b J_s = -2(Y\omega - b). \tag{9}$$

Then, the margin vector b is first initialized to $b_1 \geq 0_{N\times 1}$ with all components set to a non-negative value. At each iteration k, the weight vector ω_k is deduced from b_k by

$$\omega_k = Y^\dagger b_k, \tag{10}$$

where $Y^\dagger$ stands for the pseudo-inverse of Y. By the gradient descent method, the new estimate of the margin vector b can be computed. But it is not free to compute b since the constraint $b \geq 0_{N\times 1}$. For this, by starting with $b \geq 0_{N\times 1}$ and preventing any of its components from reducing, we obtain the HK rule for minimizing $J_s(\omega,b)$ based on (9) and (10) as follows:

$$\begin{cases} b_1 \geq 0_{N\times 1}, \\ b_{k+1} = b_k + \rho(e_k + |e_k|), \end{cases} \tag{11}$$

where the error vector $e_k = Y\omega_k - b_k$, and the learning rate $0 < \rho < 1$. In practice, we will define a termination criterion $\|b_{k+1} - b_k\|_2 \leq \xi$.

Although the HK classifier can find a separable vector in the linearly separable case and provide evidence in the linearly inseparable case [25], it is sensitive to outliers [20] and cannot guarantee good classification performance. Thus, Leski [20] proposes MHKS to remedy the shortcoming and defines a canonical hyperplane as follows:

$$Y\omega \geq 1_{N\times 1}. \tag{12}$$

Then, a hyperparameter c is introduced to tune the tradeoff between the model complexity and the training error. Consequently, the criterion of MHKS is defined as follows:

$$\min_{\omega\in\mathbb{R}^{d+1}, b\geq 0} I(\omega,b) = (Y\omega - 1_{N\times 1} - b)^T(Y\omega - 1_{N\times 1} - b) + c\tilde{\omega}^T\tilde{\omega}, \tag{13}$$

where the second term of the right-handed side of (13) is a regularization one, and the regularization parameter $c \geq 0$. The procedure of MHKS remains the same as the HK classifier except (10), which becomes

$$\omega_k = (Y^TY + c\tilde{I})^{-1}Y^T(b_k + 1_{N\times 1}), \tag{14}$$

where $\tilde{I}$ is an identity matrix with the last element on the main diagonal set to zero, and the error vector is now $e_k = (Y\omega_k - b_k - 1_{N\times 1})$. On the whole, MHKS adopts the similar principle to SVM, and maximizes the separating margin without solving the QP problem.

4.2. MatMHKS classifier

This subsection reviews our previous work MatMHKS [21] that can classify a matrix pattern reshaped from images or original one-dimensional vectors and shows a superior classification performance. Suppose that for the binary classification problem in the matrix case, there are matrix samples $Tr2D^{(N)} = \{(A_1,\varphi_1),\ldots,(A_N,\varphi_N)\}$, where N is the sample number, $A_i \in \mathbb{R}^{m\times n}$, and the corresponding class label $\varphi_i \in \{+1,-1\}$. With the formulation (33), the discriminant function of MatMHKS for the binary classification problem can be given in terms of

$$g(A_i) = u^TA_i\tilde{v} + v_0 \begin{cases} > 0, & \text{if } \varphi_i = = +1, \\ < 0, & \text{if } \varphi_i = = -1, \end{cases} \quad i=1,\ldots,N. \tag{15}$$

Similarly to MHKS, we define the equalities $\varphi_i(u^TA_i\tilde{v} + v_0) - 1 = b_i, i=1,\ldots,N$, where $b = [b_1,\ldots,b_N]^T$ is an arbitrary non-negative vector, $b_i \geq 0$, and the error vector $e = [e_1,\ldots,e_N]^T$ with all components $e_i = \varphi_i(u^TA_i\tilde{v} + v_0) - 1 - b_i, i=1,\ldots,N$. The p-th component of the e named e_p, is taken as a measure of the distance of the p-th pattern to the separation hyperplane (the distance is called margin). If the margin is positive, $e_p \geq 0$, and in this case, the

pattern A_p is correctly classified and thus e_p can be set to zero by increasing the corresponding b_p. On the other hand, if the margin of the p-th pattern is negative, $e_p<0$, but due to the constraint $b_p\geq 0$, it is impossible to prevent condition $b_p<0$ by decreasing b_p to set e_p to zero. Thus, the misclassification error in the matrix case, can be written in the form

$$I(u,\tilde{v},v_0,b)=\sum_{i=1}^{N}\hbar(-e_i), \tag{16}$$

where $\hbar(z)=1$ for $z>0$, and $\hbar(z)=0$ for $z\leq 0$. But this optimization problem is NP-complete [35] since the criterion is not a convex function. Instead of (16), the alternative criterion is defined as below

$$\min_{u\in\mathbb{R}^m,\tilde{v}\in\mathbb{R}^n,v_0,b\geq 0} I(u,\tilde{v},v_0,b)=\sum_{i=1}^{N}(\varphi_i(u^TA_i\tilde{v}+v_0)-1-b_i)^2 +c(u^TS_1u+\tilde{v}^TS_2\tilde{v}), \tag{17}$$

where $S_1=mI_{m\times m}, S_2=nI_{n\times n}$ are the two regularization matrices respectively corresponding to the u and $\tilde{v}$, the regularization parameter c ($c\in\mathbb{R}, c\geq 0$) controls the generalization ability of the classifier designed by making a tradeoff between the complexity of the classifier and the training errors. In order to express simply, we set $Y=[y_1,\ldots,y_N]^T, y_i=\varphi_i[u^TA_i,1]^T, i=1,\ldots,N, v=[\tilde{v}^T,v_0]^T$, and (17) can be simplified in matrix form as follows:

$$\min_{u\in\mathbb{R}^m,v\in\mathbb{R}^{n+1},b\geq 0} I(u,v,b)=(Yv-1_{N\times 1}-b)^T(Yv-1_{N\times 1}-b) +c(u^TS_1u+v^T\tilde{S_2}v), \tag{18}$$

where $\tilde{S_2}$ is a matrix with dimensionality of $(n+1)\times(n+1)$ and $\tilde{S_2}=\begin{pmatrix}S_2 & 0\\ 0 & 0\end{pmatrix}$. Then, the weight vector $u,\tilde{v}$ and the bias v_0 can be obtained by the gradients of the objective functions (17) and (18) with respect to u, v and b. The design procedure for MatMHKS is shown in Table 1.

4.3. Proposed multi-view-joint classifier (MultiV-MHKS)

This subsection introduces the multi-view viewpoint into the classifier design, and gives a detailed description about the proposed multi-view-combined classifier on single-view patterns. Suppose that there are N labeled samples $\{(x_i,\varphi_i)\}_{i=1}^N$ in a single-view sense, where $x_i\in\mathbb{R}^d$ and the corresponding class label $\varphi_i\in\{+1,-1\}$. The discriminant function of MHKS is (1) as follows:

$$g(x)=\tilde{\omega}^Tx+\omega_0.$$

The discriminant function of MatMHKS is (33) as follows:

$$g(A)=u^TA\tilde{v}+v_0.$$

$\{\omega,\omega_0\}$ is denoted as one MHKS-solution set and correspondingly, $\{u,\tilde{v},v_0\}$ is denoted as one MatMHKS-solution set. For a given sample with the same attributes to be classified, its vector form is x_i, and its multiple matrix forms are denoted as $\{A_i^p\}_{p=1}^M$ which are reshaped from x_i in the pre-defined M ways. Consequently, for the given sample, only one MHKS can classify it, whereas multiple MatMHKSs can classify it. The solution set $\{\tilde{\omega},\omega_0\}$ of one MHKS corresponds to the solution sets $\{u^p,\tilde{v}^p,v_0^p\}_{p=1}^M$ of multiple MatMHKSs which are different from each other in the dimensionality size of their weight vectors $u^p,\tilde{v}^p$ and have a common expression. The corresponding discriminant function set of multiple MatMHKSs is denoted as $G(A)=\{g^p(A_i^p)\}_{p=1}^M$. A natural idea is to learn the set $G(A)=\{g^p(A_i^p)\}_{p=1}^M$ such that each function correctly classifies its corresponding matrix form A_i^p of the sample x_i, and the disagreement between the outputs of the function set is farthest minimized. We fuse the M MatMHKSs into one single learning process in the framework (3).Concretely, we can get the following optimization problem:

$$\min_{u^p\in\mathbb{R}^m,\tilde{v}^p\in\mathbb{R}^n,v_0^p\in\mathbb{R}\atop p=1,\ldots,M} L=\sum_{p=1}^{M}\left(\sum_{i=1}^{N}(\varphi_ig^p(A_i^p)-1-b_i^p)^2+c^p(u^{pT}S_1u^p+\tilde{v}^{pT}S_2\tilde{v}^p)\right) +\gamma\sum_{i=1}^{N}\sum_{p=1}^{M}\left(\varphi_ig^p(A_i^p)-\sum_{q=1}^{M}r_q\varphi_ig^q(A_i^q)\right)^2, \tag{19}$$

where b_i^p is an arbitrary scalar quantity; c^p is the regularization parameter of each view; $S_1=mI_{m\times m}, S_2=nI_{n\times n}$; γ is the coupling parameter that regularizes the set $G(A)$ towards compatibility using the multiple A_i's of a given sample x_i; $r_q\geq 0, \sum_{q=1}^M r_q=1$, r_q denotes the importance of the corresponding view and the bigger the r_q is, the more important the corresponding view is. The first term of the right side of (19) is to guarantee that each view can correctly classify samples, and the second one is to minimize the disagreement between each view by making the outputs of each view be maximally close to the weight average outputs of all views.

It is known that the outputs of the sub-classifiers should disagree on labeled data in order to get a diversity in ensemble learning [53]. The diversity is supposed to improve performance in ensemble learning [47,51,53]. However, as the above statement, the proposed MVL requires an agreement among the outputs of multiple views as shown in Eq. (19). There are three reasons. Firstly, in our method, the original pattern set is reshaped into multiple different matrix representation sets, which has supplied a diversity in representation level. As Wang and Zhou [52] have stated, the key for the success of disagreement-based approaches is the existence of a diversity, and it is unimportant how the diversity is obtained. Actually, the diversity of our method is achieved through the proposed *multiviewization*. Secondly, the literatures [49,50,54] state that the disagreement of multiple views acts as an upper bound on the generalization error. Therefore, although minimizing the rate of disagreement increases the dependency between the hypotheses and the original motivation for co-training no longer holds, there is still an improved predictive performance of these co-training approaches through minimizing the disagreement. Thirdly, ensemble learning such as Bagging or Boosting [13,48] does not change the original patterns themselves. It just changes the size of the training set and thus needs an additional way to generate diversity. In contrast, our method adopts the matrixized reshaping way.

In this paper, we adopt $r_q=1/M, q=1,\ldots,M$ for simplicity. Thus (19) can be converted into (20)

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

Table 1
Algorithm **MatMHKS**.

Input: $Tr2D^{(N)}=\{(A_1,\varphi_1),\ldots,(A_N,\varphi_N)\}$
OutPut: the weight vectors $u,\tilde{v}$, and the bias v_0

1. fix $c\geq 0, 0<\rho<1$; initialize $b(1)\geq 0$ and $u(1)$; set the iteration index $k=1$;
2. $Y=[y_1,\ldots,y_N]^T$, where $y_i=\varphi_i[u(k)^TA_i,1]^T$;
3. $v(k)=(Y^TY+c\tilde{S_2})^{-1}Y^T(1_{N\times 1}+b(k))$;
4. $e(k)=Yv(k)-1_{N\times 1}-b(k)$;
5. $b(k+1)=b(k)+\rho(e(k)+|e(k)|)$;
6. if $\|b(k+1)-b(k)\|>\xi$, then go to Step 7, else stop;
7. $u(k+1)=(\sum_{i=1}^N A_i\tilde{v}(k)\tilde{v}(k)^TA_i^T+cS_1)^{-1}(\sum_{i=1}^N\varphi_i(1+b_i(k)-\varphi_iv_0)A_i\tilde{v}(k))$, $k=k+1$, go to Step 2.

$$+\gamma\sum_{i=1}^{N}\sum_{p=1}^{M}\left(\varphi_i g^p(A_i^p)-\frac{1}{M}\sum_{q=1}^{M}\varphi_i g^q(A_i^q)\right)^2. \tag{20}$$

Then, similar to MatMHKS, we set $Y^p=[y_1^p,\ldots,y_N^p]^T, y_i^p=\varphi_i[u^{pT}A_i^p,1]^T, i=1,\ldots,N, b^p=[b_1^p,\ldots,b_N^p]^T, v^p=[\tilde{v}^{pT},v_0^p]^T$, and (20) can be simplified in matrix form as follows:

$$\begin{aligned}\min_{\substack{u^p\in\mathbb{R}^m,v^p\in\mathbb{R}^{n+1}\\ p=1,\ldots,M}} L=&\sum_{p=1}^{M}((Y^pv^p-1_{N\times1}-b^p)^T(Y^pv^p-1_{N\times1}-b^p)\\ &+c^p(u^{pT}S_1u^p+v^{pT}\tilde{S_2}v^p))\\ &+\gamma\sum_{p=1}^{M}\left(Y^pv^p-\frac{1}{M}\sum_{q=1}^{M}(Y^qv^q)\right)^T\left(Y^pv^p-\frac{1}{M}\sum_{q=1}^{M}(Y^qv^q)\right),\end{aligned} \tag{21}$$

where $\tilde{S_2}$ is a matrix with dimensionality of $(n+1)\times(n+1)$ and $\tilde{S_2}=\begin{pmatrix}S_2 & 0\\ 0 & 0\end{pmatrix}$.

Now taking the gradient of (20) and (21) with respect to u^p and v^p to be zero respectively, we can obtain

$$\begin{aligned}u^p=&\left(\left(1+\gamma\left(\frac{M-1}{M}\right)^2\right)\sum_{i=1}^{N}A_i^p\tilde{v}^p(A_i^p\tilde{v}^p)^T+c^pS_1\right)^{-1}\\ &\sum_{i=1}^{N}\left(A_i^p\tilde{v}^p\left(\varphi_i(b_i^p+1)-\left(1+\gamma\left(\frac{M-1}{M}\right)^2\right)v_0^p\cdot\right.\right.\\ &\left.\left.+\gamma\frac{M-1}{M^2}\sum_{q=1,q\neq p}(u^{qT}A_i^q\tilde{v}^q+v_0^q)\right)\right)\end{aligned} \tag{22}$$

$$\begin{aligned}v^p=&\left(\left(1+\gamma\left(\frac{M-1}{M}\right)^2\right)Y^{pT}Y^p+c^p\tilde{S_2}\right)^{-1}\\ &Y^{pT}\left(1_{N\times1}+b^p+\gamma\frac{M-1}{M^2}\sum_{q=1,q\neq p}^{M}Y^qv^q\right)\end{aligned} \tag{23}$$

The gradient of (21) with respect to b^p is given as follows:

$$\nabla_{b^p}L'=-2(Y^pv^p-1_{N\times1}-b^p). \tag{24}$$

Then by denoting the vector b of the p-th view at the k-th iteration by b_k^p and with (24), we obtain

$$\begin{cases}b_1^p\geq0,\\ b_{k+1}^p=b_k^p+\rho^p(e_k^p+|e_k^p|),\end{cases} \tag{25}$$

where at the k-th iteration, the error vector of the p-th view $e_k^p=Y_k^pv_k^p-1_{N\times1}-b_k^p$, and the learning rate of the p-th view $0<\rho^p<1$. In practice, the termination criterion can be designed as

$$\frac{\|L'_{k+1}-L'_k\|_2}{\|L'_k\|_2}\leq\xi. \tag{26}$$

Such a designed procedure is termed as MultiV-MHKS and summarized in Table 2.

The discriminant function of MultiV-MHKS for the sample $z\in\mathbb{R}^d$ with the M reshaped matrices $\{Z^p\in\mathbb{R}^{m^p\times n^p}\}_{p=1}^M$ is given as follows:

$$g(z)=\frac{1}{M}\sum_{p=1}^{M}(u^{pT}Z^p\tilde{v}^p+v_0^p)\begin{cases}>0 & \text{then } z\in class+1,\\ <0 & \text{then } z\in class-1.\end{cases} \tag{27}$$

Finally, it can be found that if $M=1,\gamma=0$ of (19), MultiV-MHKS is degenerated into MatMHKS. Meanwhile, further setting $m=1$, $u=1$, the procedure is degenerated to MHKS. Thus both MHKS and MatMHKS classifiers can be reviewed as two special instances of MultiV-MHKS.

Table 2
Algorithm **MultiV-MHKS**.

Input: Labeled data $\{(x_i,\varphi_i)\}_{i=1}^N$, and the pre-defined M ways that satisfy the condition $mn=d$.
OutPut: The solution to MultiV-MHKS $\{u^p,\tilde{v}^p,v_0^p\}_{p=1}^M$.

1. Reshape x_i to $\{A_i^p\}_{p=1}^M$ with the pre-defined M ways, where the value of mn equals to the value of d; initialize $u_1^p,v_1^p,p=1,\ldots,M$ at random; set the initial value of Y by $Y_1^p=[y_1^p,\ldots,y_N^p]^T, y_i^p=\varphi_i[u_1^{pT}A_i^p,1]^T$, $i=1,\ldots,N, p=1,\ldots,M$; let $k=1$;
2. Do until the termination criterion (26) is satisfied:
   (a) For $p=1\ldots M$:
      (i) Compute $v_{k+1}^p,u_{k+1}^p,b_{k+1}^p$ with (23), (22), and (25), respectively;
      (ii) Set $Y_{k+1}^p=[y_1^p,\ldots,y_N^p]^T, y_i^p=\varphi_i[u_{k+1}^{pT}A_i^p,1]^T, i=1,\ldots,N$;
   (b) Compute L'_{k+1} with (20) or (21);
   (c) Increment k;
3. Return the final $\{u^p,\tilde{v}^p,v_0^p\}_{p=1}^M$.

5. Experiments

This section gives the demonstration on the effectiveness of the proposed MVL. The proposed MVL can be composed of two components: (1) pattern representation in multiple views (multi-viewization); (2) the joint learning processing with the generated views. Thus, in order to demonstrate the proposed approach, we have to demonstrate the effectiveness of the two components, respectively. For the first component, we give a comparison between the proposed multiviewization processing and Bagging [13], Attribute Bagging [14]. For the second component, with the multiple pattern representations generated from the single-view patterns, the joint learning is compared with the separate one with majority voting and co-training [9]. We also compare the proposed MultiV-MHKS with its corresponding single-view version MHKS and SVM in terms of both classification and running time. Simultaneously, we give the running time analysis of the proposed MVL. Finally, we discuss both the complementarity of the generated views and the multiviewization processing with random attribute permutation.

5.1. Experimental setting

In order to evaluate the feasibility and effectiveness of the multi-view classifier MultiV-MHKS, it is compared with its base classifier MHKS [20], Bagging [13] and Attribute Bagging [14]. The benchmark data sets used here are obtained from UCI Machine Learning Repository [24]. All computations are run on Windows 2000 Terminal and MATLAB environment. The involved parameters of all the classifiers here are given as follows. In the proposed approach MultiV-MHKS, $b_1^p=10^{-6},\rho^p=0.99$, $\xi=10^{-4},p=1,\ldots,M$. In MHKS, $b_1=10^{-6},\rho=0.99,\xi=10^{-4}$. The coupling parameter γ in MultiV-MHKS and the regularization parameter c in MultiV-MHKS and MHKS are both selected from the set $\{2^{-4},2^{-3},\ldots,2^3,2^4\}$ by N-fold cross-validation [28–30], i.e. randomly split the samples into two parts (the training and testing sets), and repeat the procedure N times. Both the γ and c are finally determined by the best average accuracy on the N testing sets. In our experiments, N is set to 10.

5.2. Multi-view learning vs. single-view learning

This subsection examines whether the multi-view learning MultiV-MHKS is superior to its corresponding single-view learning MHKS and support vector machines (SVM) with the linear kernel in terms of classification performance. In practise, a kernelized classifier is different from the non-kernelized one in terms of performance. The MultiV-MHKS is a linear version. Thus

here, we only select the linear SVM as the compared single-view classifiers for fairness. Table 3 gives the compared results and shows the effectiveness of the MultiV-MHKS, where the best accuracies of the data sets are in bold. The used data sets are shown in Table 4, where Shuttle-landing-control and Echocardiogram are denoted as SLC and Echo. for short, respectively. Two kinds of matrix representations (i.e. $M=2$ in MultiV-MHKS) for each data set are generated as shown in Section 3. Ten-fold cross-validation is adopted and their classification accuracies on their corresponding testing sets are averaged and reported in Table 3. Here, the best classification performance of the MultiV-MHKS corresponds to the best M ($M=2$) matrix representations that are determined by an internal cross-validation on each training fold. In each training fold, the best subset of matrix representations for MultiV-MHKS are selected from the set that is shown in the fourth column of Table 4. In addition to reporting the average accuracies across the 10 folds, we also perform the paired t-test [32] comparing MulitV-MHKS with MHKS. The null hypothesis H_0 demonstrates that there is no significant difference between the mean number of samples correctly classified by MulitV-MHKS and MHKS. Under this assumption, the p-value for each test is the probability of a significant difference in correctness values occurring between two testing sets. Thus the smaller the p-value, the less likely that the observed difference results from identical

Table 3
Average testing accuracy (%) and p-values of MultiV-MHKS and MHKS, Linear SVM.

Data sets	MultiV-MHKS accuracy	MHKS accuracy p-value	SVM accuracy p-value	Data sets	MultiV-MHKS accuracy	MHKS accuracy p-value	SVM accuracy p-value
Hepatitis	**80.63 ± 1.81**	77.60 * ± 4.04 0.0290	79.75 ± 0 0.2846	Cmc	50.01 ± 1.41	48.62 ± 1.19 0.0786	**51.42 ± 1.35** 0.1367
Glass	**99.24 ± 0.60**	87.43* ± 3.69 9.3441e−009	99.14 ± 0.94 0.7730	Echocardiogram	**89.40 ± 2.58**	87.76 ± 3.20 0.2235	88.36 ± 2.31 0.4992
Water	**96.97 ± 2.71**	95.15 ± 2.45 0.1720	95.76 ± 3.83 0.1092	House-votes	**92.81 ± 1.45**	**92.81 ± 1.69** 0.1132	90.45 ± 1.31 0.0573
Sonar	**76.67 ± 3.23**	75.65 ± 1.95 0.4049	73.70 ± 3.44 0.1200	Horse-colic	**77.85 ± 1.89**	76.96 ± 1.72 0.1206	48.17* ± 0 1.0159e−012
Pima-diabetes	**71.14 ± 0.27**	69.31* ± 2.57 0.0336	52.60* ± 0.16 1.5543e−015	Housing	**92.91 ± 0**	92.68 ± 0.74 0.3306	90.91 ± 0.23 0.3306
Optdigits	**95.72 ± 1.23**	94.82 ± 2.05 0.2796	92.26* ± 0.81 8.1710e−007	Ionosphere	**89.33 ± 1.50**	88.27 ± 2.18 0.2652	86.67* ± 3.09 0.0044
Dermatology	97.17 ± 0.96	**97.28 ± 0.84** 0.9569	96.56 ± 1.77 0.1383	Wine	**94.43 ± 1.74**	94.34 ± 2.66 0.9265	92.92 ± 2.64 0.0686
Lenses	53.85 ± 10.90	43.85 ± 12.58 0.0906	**61.54 ± 0** 0.1358	Iris	**97.73 ± 1.09**	93.60* ± 1.63 3.2090e−006	96.40* ± 1.29 0.0490
Balance	**88.85 ± 1.02**	87.37* ± 0.90 7.0498e−005	88.27 ± 2.27 0.1055	Shuttle-landing-control	**72.86 ± 10.54**	68.57 ± 13.12 0.4313	67.14 ± 10.54 0.3065
Breast-cancer-wisconsin	**97.32 ± 0.78**	96.42 ± 1.18 0.1253	78.07* ± 0 0.0069	Lung-cancer	**53.33 ± 8.46**	50.00 ± 10.99 0.3506	44.00* ± 13.12 0.0056

Note: The p-values are from a t-test comparing each classifier with MultiV-MHKS. The best accuracy results are in bold. An asterisk * denotes that the difference from MultiV-MHKS is significant at 5% significance level, i.e. the p-value less than 0.05.

Table 4
Information of data sets.

Data sets	Number of attributes	Maximum number of views	Generated views (the size of matrix)
Wine	12	6	1×12; 2×6; 3×4; 4×3; 6×2; 12×1
SLC	6	4	1×6; 2×3; 3×2; 6×1
Echo.	12	6	1×12; 2×6; 3×4; 4×3; 6×2; 12×1
Sonar	60	10	2×30; 3×20; 4×15; 5×12; 6×10; 10×6; 12×5; 15×4; 20×3; 30×2
Iris	4	3	1×4; 2×2; 4×1
Glass	10	4	1×10; 2×5; 5×2; 10×1
Water	38	4	1×38; 2×19; 19×2; 38×1
Pima-diabetes	8	4	1×8 2×4 4×2 8×1
Dermatology	34	4	1×34; 2×17; 17×2; 34×1
Lenses	4	3	1×4; 2×2; 4×1
Balance	4	3	1×4; 2×2; 4×1
Breast-cancer-wisconsin	10	4	1×10; 2×5; 5×2; 10×1
Cmc	9	3	1×9; 3×3; 9×1
Hepatitis	18	6	1×18; 2×9; 3×6; 6×3; 9×2; 18×1
Horse-colic	27	4	1×27; 3×9; 9×3; 27×1
Housing	12	6	1×12; 2×6; 3×4; 4×3; 6×2; 12×1
Ionosphere	34	4	1×34; 2×17; 17×2; 34×1
House-votes	16	5	1×16; 2×8; 4×4; 8×2; 16×1
Lung-cancer	56	8	1×56; 2×28; 4×14; 7×8; 8×7; 14×4; 28×2; 56×1
Optdigits	64	5	2×32; 4×16; 8×8; 16×4; 32×2

testing 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-MHKS is superior to that of MHKS on most data sets except Dermatology and House-votes. Even on such two data sets, the MultiV-MHKS also has a competitive performance to MHKS. Further, the p-values also show that MultiV-MHKS has a different significance from MHKS on Iris, Glass, Pima-diabetes, Hepatitis and Balance in terms of classification performance. Compared with the linear SVM, the MultiV-MHKS also shows its superior performance in terms of classification.

5.3. Multiviewization vs. Bagging and Attribute Bagging

For the given single-view patterns, the proposed MVL reshapes the original vector representation into M matrix ones, each of which is taken as one view of the original patterns. The proposed multiviewization can be regarded to work with the different representation level of the original patterns. At present there are two typical ensemble schemes on single-view patterns: Bagging [13], AdaBoosting [48], and Attribute Bagging [14]. Since both Bagging and AdaBoosting belong to the sampling scheme, we take Bagging as the discussed paradigm. Bagging generates M training sets by randomly sampling M times with replacement, then develops M classifiers on the generated training sets respectively, and finally classifies each pattern by equal weight majority-voting on all the M classifiers. Bagging can be regarded to work with the size level of the original single-view patterns. Different from Bagging that generates different training sets by randomly sampling, Attribute Bagging produces different training sets by randomly selecting subsets of attributes M times, then develops M classifiers on the generated training sets respectively, and finally classifies each pattern by equal weight majority-voting. Attribute Bagging can be regarded to work with the feature level of the original single-view pattern. It can be found that the proposed multiviewization, Bagging, Attribute Bagging work with representation, sample, and attribute levels of the original single-view patterns, respectively. Both Bagging and Attribute Bagging have been demonstrated to be effective in improving classification performance [13,14]. Here, we compare the proposed multiviewization with both Bagging and Attribute Bagging so as to validate whether it is also effective or better for the work with representation level of the original patterns.

We implemented the proposed MVL, Bagging, and Attribute Bagging on some UCI benchmark data sets that are shown in Table 4. Table 4 gives all the possible matrix representations from their corresponding vector. Taking "Sonar" for example, the number of its attributes is 60. Thus, there are 10 kinds of matrix representations (views) for "Sonar" as shown in Table 4. Fig. 7 shows the classification performance on the 20 data sets, where the x-axis denotes the number of views on the given data set and the y-axis denotes the classification accuracies. Since it is different for the number of the possible matrix representations for each data set as shown in Table 4, it is also different for the range of the number of the views on each data set. In order to select the views of each data set of Fig. 7, we first carry out the sub-classifier MatMHKS in each view. Then, for each joint combination with M views, the selected M views correspond to the best M classification accuracies of MatMHKS in single view. From the figure, it can be clearly found that (1) the proposed multi-view strategy learning on multiple matrix representations takes the first place on all the data sets only but Sonar Lenses and Horse-colic; (2) Attribute Bagging and Bagging take the second and third places, respectively. In order to analyze the experimental phenomenon, we give the comparison among the proposed MVL, Bagging and Attribute Bagging tabulated in Table 5, where the second column "Sample" represents whether the algorithm keeps the number of the training set, the third column "Attribute" represents whether the algorithm keeps the size of the feature set, and the fourth column "Representation" represents how many kinds of representations the algorithm adopts. From Table 5, we find that compared with the other two algorithms, the proposed MVL has the advantages in terms of the number of samples, the size of the feature set and the representations, which brings up the superior performance of the proposed MVL. Further, it is known that the number of samples plays an important role on algorithms including Bagging. Thus, Bagging might not work well in the small-scale sample case. Similarly, Attribute Bagging might not work well on the data with a small-size-set of features. But the proposed MVL can still work well in the above two cases due to that it utilizes all given samples with full features. Fig. 7 validates the above statement.

Table 5
Comparison among the proposed MVL, Bagging and Attribute Bagging.

Algorithm	Sample	Attribute	Representation
Bagging	No	Yes	1
Attribute Bagging	Yes	No	1
MultiV-MHKS	Yes	Yes	$M(\geq 1)$

It can be found that the performance of MHKS with Bagging on these data sets such as Wine, SLC and Echo. is much lower than that of MHKS. The phenomenon mainly attributes to that in our experimental setting for Bagging, the size of the training set for each sampling is kept to the same size as that of the original training set. Since the training set is randomly sampled with replacement, the actual size of the sampled training set is smaller than the size of the original one [14]. The experimental results here show that the decrease of the size of the training data set plays an important role on the classification performance here. It accords with the results of the literature [14] that Attribute Bagging is better than Bagging in terms of classification.

Fig. 7 also shows that the performance of the MultiV-MHKS achieves its best accuracy with the number of views $M=2$ on the data sets (Wine, SLC, Echo., Iris, Glass, Dermatology, Lenses, Balance, Breast-cancer-wisconsin, Cmc, Optdigits, Horse-colic, Housing, Ionosphere, and House-votes). This phenomenon seems to show that the proposed multi-view learning can farthest increase the single-view learning only on the case of $M=2$ in terms of classification performance. And we need not select the value of the number of the views M since the MultiV-MHKS with $M=2$ can success, which can avoid a large computation in terms of searching parameters. On the other side, this phenomenon also shows that the performance of the MultiV-MHKS seems to decrease with the number of views growing larger than two on some data sets, which might attribute to the over-fitting or the increase of correlation among views. The experimental results have told that the proposed multi-view learning with $M=2$ is enough to give a superior performance. This phenomenon may be just for the proposed multi-view matrixization pattern representation here. However, for other cases, more weaker-correlated views should be favorable for boosting performance [38]. As we have known, the more the weaker-correlated views, the more variance is reduced. Thus more likely, the more generalization is boosted. Therefore, it is a future work for us to discuss why this phenomenon appears.

5.4. Joint vs. separate learning on multiple views

This subsection gives the reasons why a joint rather than separate learning is adopted for the different matrix representations generated from the original single-view patterns. Here, we have implemented the voting learning that separably carries out MatMHKS

on different matrix representations from the original single-view patterns and classifies each data instance by equal weight voting on different MatMHKS. Tables 6 and 7 give the average classification performance and the training time of MultiV-MHKS and MatMHKS with voting on all the range of the views for the six data sets: Sonar, Wine, SLC, Iris, Glass, Echo., respectively. Taking "Sonar" for example, the maximum number of its views is 10 as shown in Table 4. Thus, the corresponding results of Sonar in Tables 6 and 7 denote the average classification accuracies and training time on the range of the generated views from 2 to 10, respectively. From Tables 6 and 7, it can be found that the joint learning MultiV-MHKS can significantly improve classification performance on some data sets such as Glass and Iris. Simultaneously, the proposed joint learning takes a comparable training time to that of the separate learning.

5.5. Joint vs. co-training learning on multiple views

This subsection further gives a comparison between the proposed joint learning (MultiV-MHKS) and the co-training learning [9] on multiple matrix representation views generated from a given single-view patterns. From the literature [9,10], it can be found that the co-training learning is only fit for two views, i.e. $M=2$. Thus, each data set used here first is multiviewized into two kinds of matrix representations, each of which is one view of the original data set denoted as V_1 and V_2, respectively. Then, each data set is partitioned into three equal-size parts: training example set L, unlabeled example set U, and test example set T, where each set has two views. The co-training learning loops until the set U is null. In each iteration, (1) use L to train two classifiers MatMHKS H_i that work in V_i, $i=1,2$, respectively; (2) let H_i label p examples from U in the two views, respectively; (3) move the $2p$ self-labeled examples from U to L. The final classifier of co-training is $H=\frac{1}{2}(H_1+H_2)$. Fig. 2 gives the classification performance comparison between MultiV-MHKS and the co-training learning on the 20 data sets shown in Table 4. Since co-training is described for two views in the literatures [9,10], the number of the generated kinds of matrix representations is two and the results of Fig. 2 correspond to the optimal combinations in two kinds of matrix representations through an internal cross-validation on each training fold. From this figure, it can be clearly found that the proposed joint learning strategy is superior to the co-training learning in terms of classification. In other words, the co-training learning does not work well in this case.

5.6. Analysis of computational complexity in MultiV-MHKS

In this subsection, we give a discussion on the computational complexity of the joint learning MultiV-MHKS. Since MultiV-MHKS is taken as a multi-view version of MHKS, we first analyze both of them in terms of computational complexity. Both MultiV-MHKS and MHKS are iterative algorithms. In each iteration, MHKS mainly takes time computing on the formulation (14) that needs $o(d^3)$ where d is the number of the original attributes. If $d=m\times n$, MultiV-MHKS needs $o(M(m^3+n^3))$ mainly for both (22) and (23) in each iteration as shown in Table 2. Since $M\leq 10$ in our experiments, both MultiV-MHKS and MHKS should take a comparable computational cost. Table 8 gives the average training time of both MultiV-MHKS and MHKS under the environment as shown in the section of Experimental setting. From this table, it can be found that the average running times of MultiV-MHKS are actually comparable to those of MHKS on most of the used data sets.

Table 7
Training time comparison (in s) between MultiV-MHKS and MatMHKS with voting.

Data sets	Glass	Wine	Iris	SLC	Echo.	Sonar
MultiV-MHKS	27.73	0.50	17.05	0.02	0.46	7.58
MatMHKS with voting	21.25	25.24	3.59	0.02	7.11	4.69

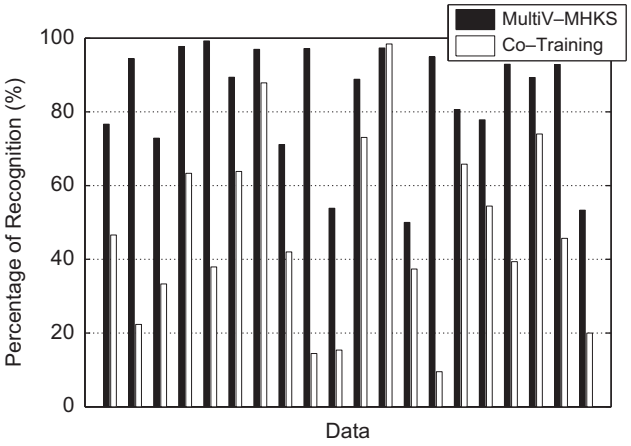

Fig. 2. Classification performance (%) comparison between MultiV-MHKS and Co-Training on data sets (from left to right): Sonar, Wine, SLC, Iris, Glass, Echo., Water, Pima-diabetes, Dermatology, Lenses, Balance, Breast-cancer-wisconsin, Cmc, Optdigits, Hepatitis, Horse-colic, Housing, Ionosphere, House-votes, Lung-cancer.

Table 8
Training time (in 's') comparison between MultiV-MHKS and MHKS.

Data sets	MultiV-MHKS	MHKS
Hepatitis	1.56	4.39
Shuttle-landing	0.04	0.11
Water	2.19	13.29
Wine	0.47	4.76
Pima-diabetes	1.35	0.05
Optdigits	6433	3995
Dermatology	125.73	63.35
Lenses	0.06	0.21
Balance	10.36	6.67
Breast-cancer-wisconsin	7.63	5.13
Cmc	2.74	0.55
Echocardiogram	1.04	0.78
House-votes	1.17	1.34
Horse-colic	5.44	0.38
Housing	0.61	1.80
Ionosphere	5.03	1.33
Sonar	8.87	0.18
Iris	0.35	4.20
Glass	85.48	48.10
Lung-cancer	4.23	1.65

Table 6
Classification performance (%) comparison between MultiV-MHKS and MatMHKS with voting.

Data sets	Glass	Wine	Iris	SLC	Echo.	Sonar
MultiV-MHKS	$\mathbf{98.80\pm0.20}$	93.46 ± 1.22	$\mathbf{97.10\pm0.42}$	69.53 ± 3.01	88.16 ± 0.70	76.92 ± 0.53
MatMHKS with voting	88.17 ± 0.90	93.42 ± 0.73	93.20 ± 1.13	70.47 ± 0.80	88.10 ± 0.82	75.74 ± 0.93

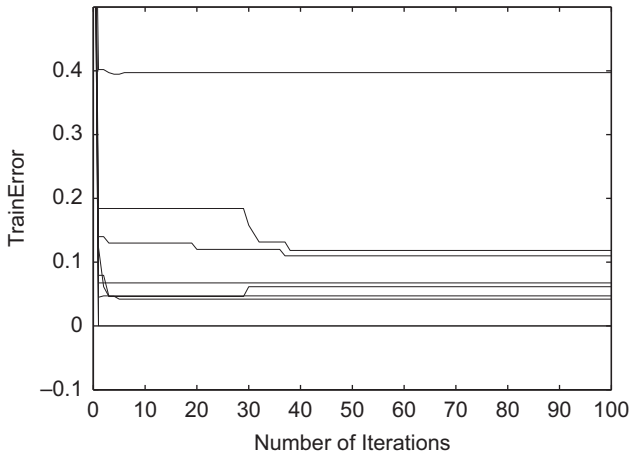

Fig. 3. Convergence of MultiV-MHKS on the training data.

Further, we have also experimentally demonstrated that MultiV-MHKS can converge well. Fig. 3 shows the training errors change with the iteration number of MultiV-MHKS on the binary-class data sets: Water, Sonar, Pima-diabetes, Breast-cancer-wisconsin, Echocardiogram, Hepatitis, Housing, House-votes, and Shuttle-landing-control, respectively. From the figure, it can be found that all the training errors of the used data sets can obviously converge to stable values. Especially for Water, Sonar, Pima-diabetes, Housing, and Shuttle-landing-control, the iterations less than 7 are usually enough to achieve convergence.

5.7. Analysis on large dimensional patterns

As Table 4 shows, the maximal dimensionality of the used data is 64. In order to further explore the effect of the proposed method, we carry out MultiV-MHKS, MHKS, MatMHKS and SVM on the large dimensional database MNIST.[2] The MNIST consists of the handwritten digits from 0 to 9. It has a training set of 60 000 samples and a test set of 10 000 samples. The size of each sample of the MNIST is 28×28. Here, we adopt the whole test set of 10 000 samples and the subset of the training set with 50 000 samples due to the experimental condition.

Fig. 4 shows the classification accuracies of MultiV-MHKS, MHKS, MatMHKS and SVM as a function of the training size on the MNIST, where the sizes of the training sets are respectively taken as 1000, 3000, 5000, 10 000, 15 000, 20 000, 30 000, 40 000, 50 000 samples and the size of the test set is fixed to 10 000 samples. In this figure, the SVM adopts the Linear kernel $k(x_i,x_j)=x_i^T x_j$ and the Poly kernel $k(x_i,x_j)=(x_i^T x_j)^d$ with the parameter $d=2$, where the regularization parameter c is selected from the set $\{2^{-4},2^{-3},\ldots,2^3,2^4\}$ by N-fold cross-validation and Table 9 lists the corresponding optimal c. Both MHKS and MatMHKS use the same parameters as the description of Section 5.1. In MatMHKS, each sample is respectively reshaped into different matrices with 2×392, 392×2, 4×196, 196×4, 7×112, 112×7, 8×98, 98×8, 14×56, 56×14, 16×49, 49×16, 28×28. Fig. 4 shows the optimal results of MatMHKS with respect to the above different sizes. In MultiV-MHKS, we empirically adopts the two, three and four kinds of the matrix sizes corresponding to the first two, three and four best results of MatMHKS, respectively. From this figure, it can be found that (1) MultiV-MHKS always takes the first place in terms of classification performance; (2) MHKS has the worst classification accuracy in all the cases; (3) the proposed multi-view method is superior to the single-view one in the linear learning framework. Meanwhile, Fig. 5 shows the classification accuracies of MultiV-MHKS with the number of the views $M=2,3,4$ as a function of the training size on the MNIST, where the size of the test set is fixed to 10 000 samples. From this figure, it can be clearly found that (1) the classification accuracy of MultiV-MHKS increases with the growing number of the views M in all the cases with different training set sizes; (2) the proposed MultiV-MHKS with different M has a comparable performance on the larger training samples.

It should be stated that the literature [56] proposes a fast SVM that can get a much effective and efficient performance on MNIST. In their work [56], the fast SVM can get a 99.79% classification accuracy on the whole training set of MNIST since it removes

[2] Available at http://yann.lecun.com/exdb/mnist/.

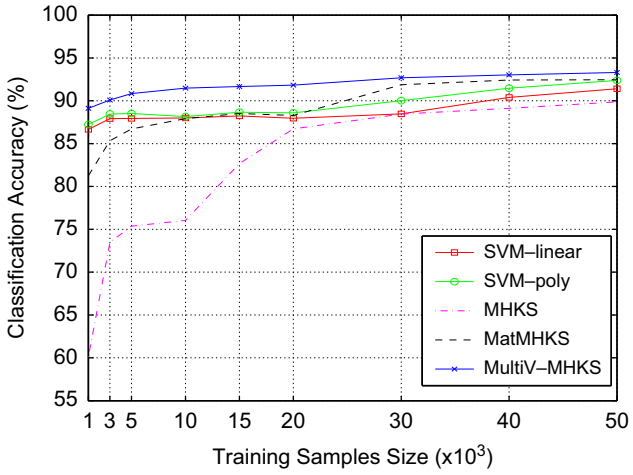

Fig. 4. Classification accuracies of MultiV-MHKS, MHKS, MatMHKS and SVM as a function of the training size on the MNIST, where the size of the test set is fixed to 10 000 samples.

Table 9
The optimal regularization parameter c of SVM in Fig. 4.

Kernel	The size of training samples								
	1000	3000	5000	10 000	15 000	20 000	30 000	40 000	50 000
Linear	2^{-2}	2^{-2}	2^{-2}	2^2	2^4	2^{-2}	2^{-2}	2^{-2}	2^{-2}
Poly	2^{-4}	2^{-4}	2^{-4}	2^2	2^2	2^2	2^2	2^2	2^2

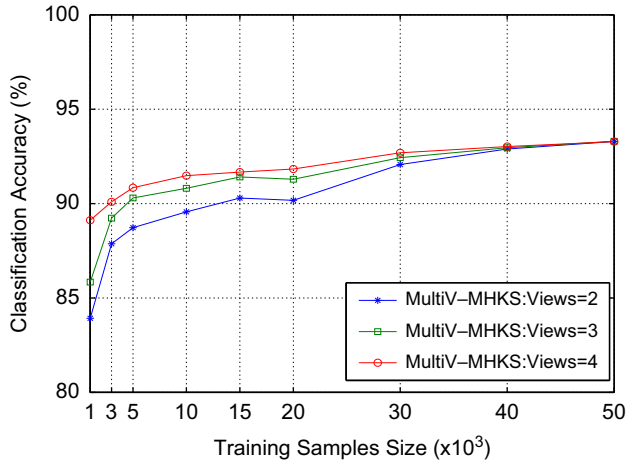

Fig. 5. Classification accuracies of MultiV-MHKS with the number of views $M=2,3,4$ as a function of the training size on the MNIST, where the size of the test set is fixed to 10 000 samples.

most non-support vectors quickly and adopts some effective strategies such as kernel caching and efficient computation of kernel matrix. In contrast, the proposed MultiV-MHKS has some difficulty in outperforming the fast SVM since it is only a linear machine. On the other hand, we here focus on the proposed multiviewization that has been demonstrated more effectiveness than the single-view MHKS. Thus we adopt the original SVM without efficient optimization for comparison. But the effective and efficient work of [56] makes us explore the efficiency and kernelization of our proposed work in future.

5.8. Further discussion

5.8.1. MHKS vs. MatMHKS

Since MHKS is the baseline of the MultiV-MHKS and MatMHKS is taken one view of MutliV-MHKS. Here, we give the relationship between MHKS and MatMHKS. First, we give the following lemma.

Lemma 1 (*Graham [41]*). *Let* $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$ *and* $C \in \mathbb{R}^{p\times q}$, *then*

$$vec(ABC) = (C^T \otimes A)vec(B), \tag{28}$$

where $vec(X)$ *denotes an operator that vectorizes the matrix* X *into the corresponding vector. For example, let* $X = (x_{ij}) \in \mathbb{R}^{p\times q}$ *and* $x_i = (x_{1i}, \ldots, x_{pi})^T$ *is the* i*-th column of* X, *and thus* $vec(X) = (x_1^T, \ldots, x_i^T, \ldots, x_q^T)^T$ *is a vector with* $p \times q$ *dimensionality.* " $\otimes$ " *denotes Kronecker product operation.*

Then, since we revealed the relationship between MatMHKS and MHKS in the literature [21], here we give the conclusion as stated in the following theorem.

Theorem 1. *Let the discriminant functions of MHKS and MatMHKS respectively be the function*: (i) $g(x) = \tilde{\omega}^T x + \omega_0$ *and* (ii) $g(A) = u^T A\tilde{v} + v_0$, *then*

(a) *both* (i) *and* (ii) *may have the same form*;
(b) *the solution space for the weights in MatMHKS is contained in that of MHKS, and MatMHKS is a MHKS imposed with Kronecker product decomposability constraint.*

Further, in the literature [40], we have also experimentally demonstrated that in searching for the optimal weight vectors $u,\tilde{v}$ of MatMHKS, MatMHKS could be guided by some prior information such as the structural or locally spatial information which is reflected in the representation of the Kronecker production of $u,\tilde{v}$. That is the reason why MatMHKS outperforms MHKS on image data sets [40]. More importantly, MatMHKS can avoid over-fitting since it makes implicitly a tradeoff between a less constrained model with more parameters (MHKS) and a more constrained model with fewer parameters (MatMHKS).

5.8.2. The Rademacher complexity of MultiV-MHKS, MatMHKS and MHKS

The motivation of the proposed method is that it might obtain some useful information for classification to reshape the features of one pattern into different matrices and jointly learn on these matrices. In detail, for some patterns which inherently possess the two-dimensionality structure like images, the matrix form of classifier could be advantageous than the vector format as the former preserves the location information in the pattern representation. In other words, it could depict both global and local information to reshape 2D structural patterns into new matrices according to Figs. 1 and 6, which is also demonstrated in the literature [57]. Thus, learning on different matrices can increase performance in this case.

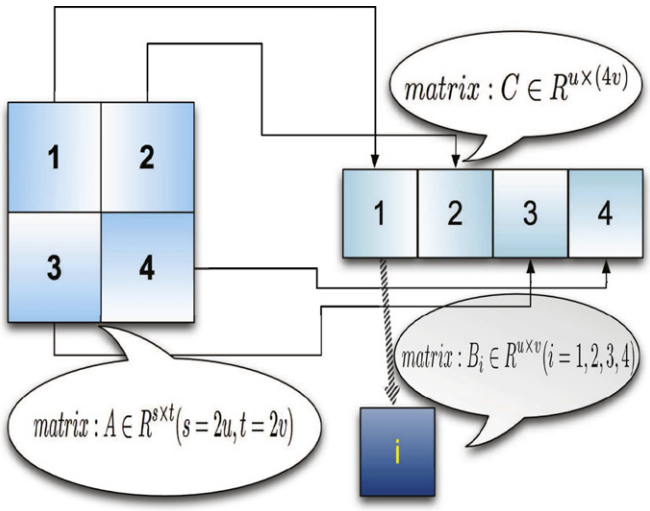

Fig. 6. Reshape matrix $A \in \mathbb{R}^{s\times t}$ to another matrix $C \in \mathbb{R}^{u\times 4v}$, where $s=2u$, $t=2v$ and the i-th sub-matrix $B_i \in \mathbb{R}^{u\times v}, i = 1 \ldots 4$.

On the other hand, it cannot always be guaranteed that arbitrarily reshaping a vector into the corresponding matrix format could bring location information of the pattern. However, the experimental results in Table 3 show the better performance of the proposed approach than the single-view algorithm on some non-image data sets. In order to explain this phenomenon, we analysis the proposed multi-view method with the generalization risk bound.

It is well-known that the analysis of the generalization risk bound is important for theoretically interpreting performance behavior of a learning algorithm. Here, we give the discussion for the MultiV-MHKS, MatMHKS and MHKS in terms of the generalization risk bound with the Rademacher complexity. It is known that the classical risk bound theory was proposed by Vapnik and Chervonenkis [4] and can be described through Theorem 2.

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* 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* F *satisfies*

$$P(y \neq g(x)) \leq \hat{P}_n(y \neq g(x)) + c\sqrt{\frac{VC(\mathsf{F})}{n}}, \tag{29}$$

where $VC(\mathsf{F})$ *denotes the Vapnik–Chervonenkis dimension of* 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(F) dimension measures the complexity of the class function F. Further, the Rademacher complexity was proposed as an alternative notion of the complexity of a function class F [2]. Here, the Rademacher complexity is used to measure the complexity of the proposed MultiV-MHKS. Definition 1 gives a definition of the Rademacher complexity [2].

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 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 random variable

$$\hat{R}_n(\mathsf{F}) = \mathbf{E}\left[\sup_{g\in\mathsf{F}}\left|\frac{2}{n}\sum_{i=1}^n \sigma_i g(x_i)\right| \middle| x_1, \ldots, x_n\right], \tag{30}$$

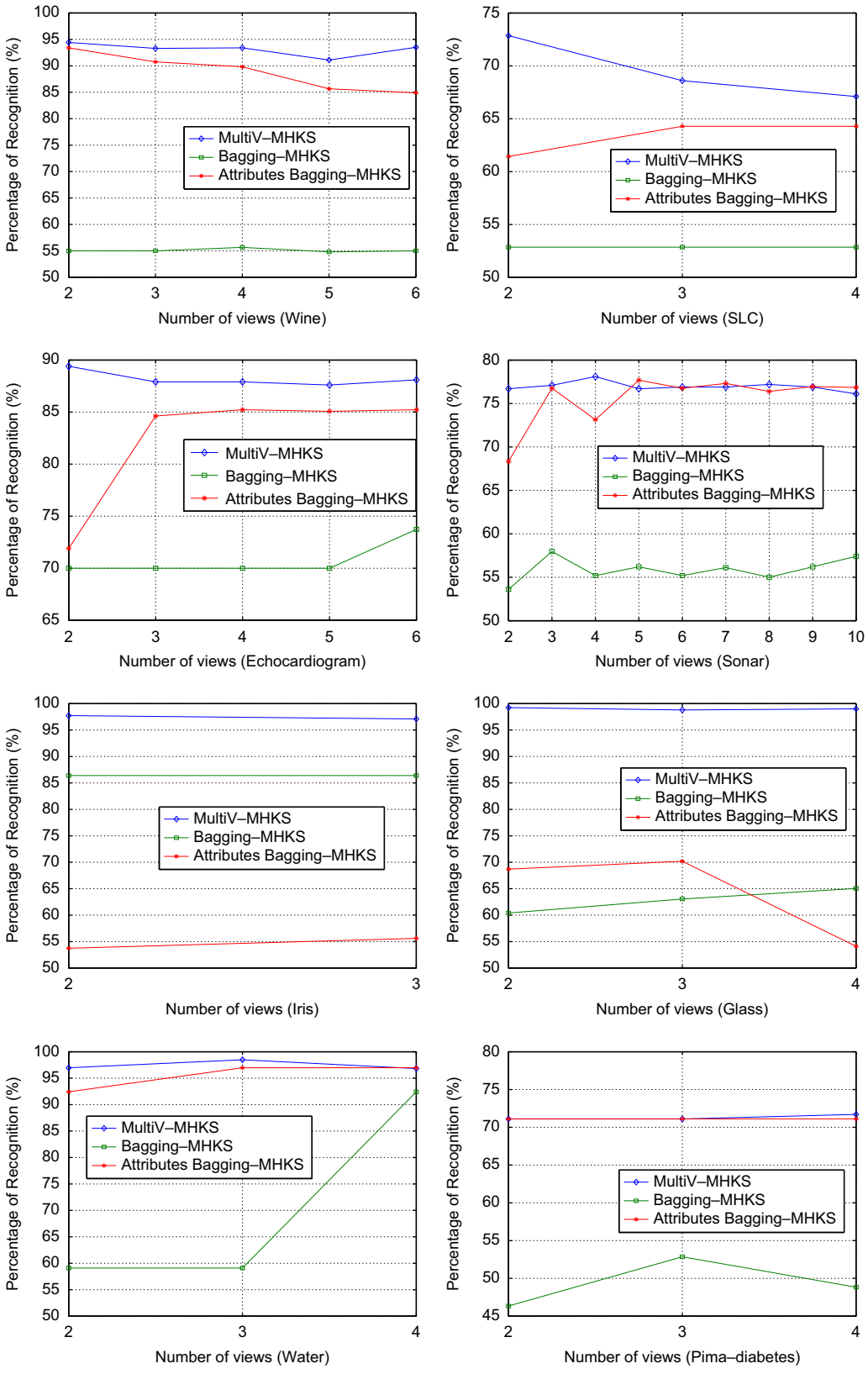

Fig. 7. The classification accuracies of MultiV-MHKS, Bagging and Attribute Bagging as a function of the number of the views on the given data sets (Wine, SLC, Echo., Sonar, Iris, Glass, Water, Pima-diabetes, Dermatology, Lenses, Balance, Breast-cancer-wisconsin, Cmc, Optdigits, Hepatitis, Horse-colic, Housing, Ionosphere, House-votes, and Lung-cancer).

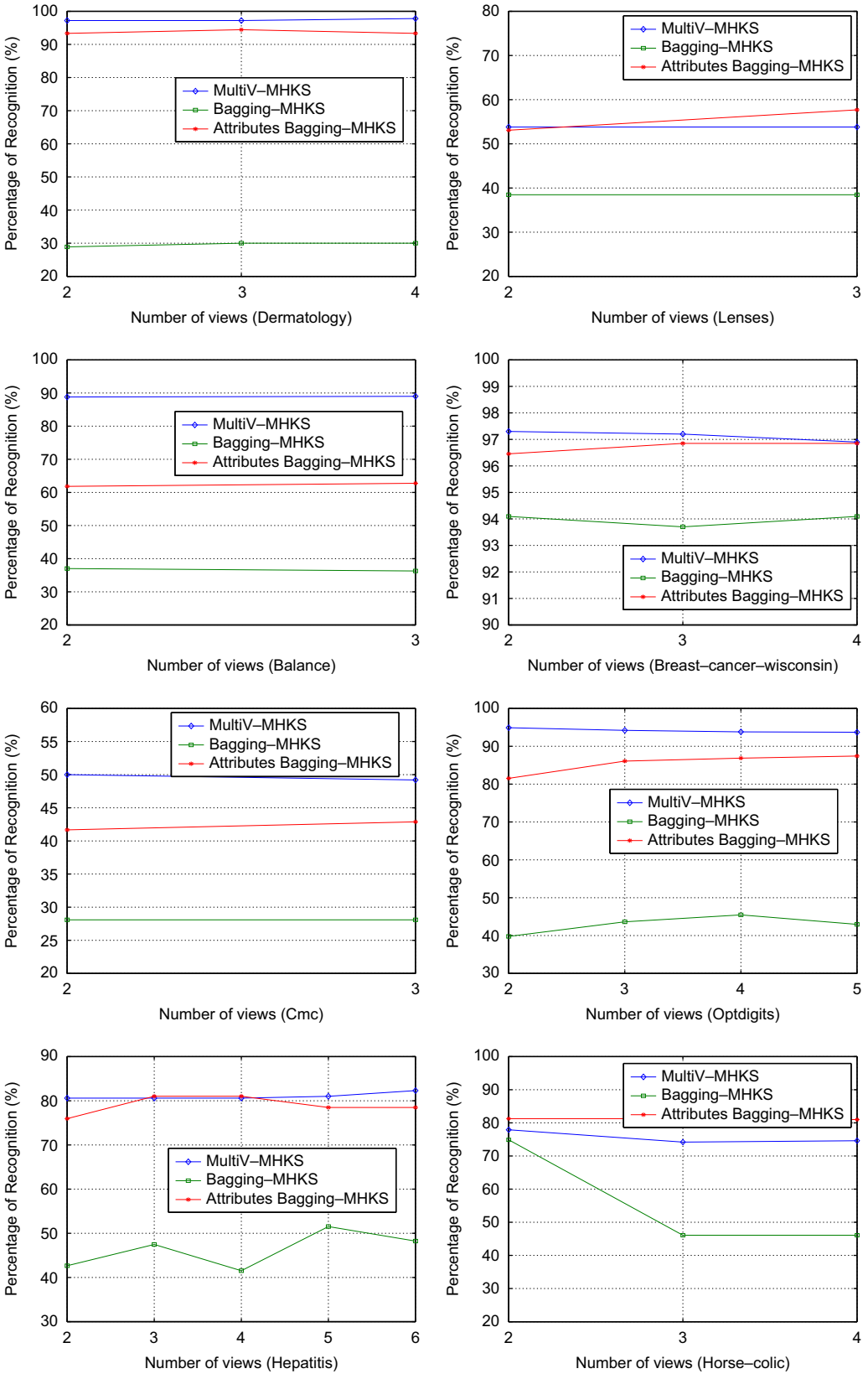

Fig. 7. (*continued*)

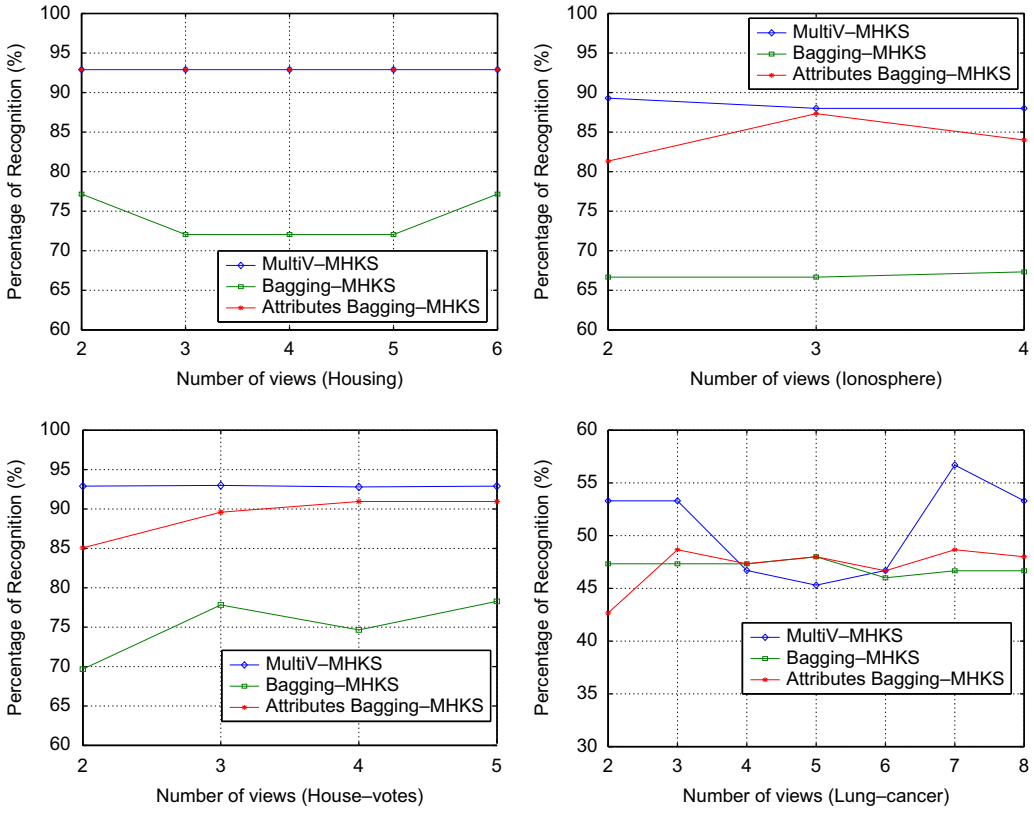

Fig. 7. (*continued*)

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

$$R_n(\mathbf{F}) = \mathbf{E}\hat{R}_n(\mathbf{F}). \tag{31}$$

Theorem 3 [5] gives the generalization risk bound of $\mathbf{F}$ with the Rademacher complexity $R_n(\mathbf{F})$.

Theorem 3. 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

$$P(y \neq g(x)) \leq \hat{P}_n(y \neq g(x)) + \frac{R_n(\mathbf{F})}{2} + \sqrt{\frac{\ln(1/\delta)}{2n}}. \tag{32}$$

We use the $R_n(g_{MultiVMHKS})$,$R_n(g_{MatMHKS})$ and $R_n(g_{MHKS})$ to denote the Rademacher complexities of the MultiV-MHKS, MatMHKS and MHKS, respectively. First, we give the relationship between $R_n(g_{MultiVMHKS})$ and $R_n(g_{MatMHKS})$. It is known that the generalization risk bound of the single-view MHKS satisfies the inequality (32). According to Eqs. (33) and (34),

$$gA = u^T A\tilde{v} + v_0, \tag{33}$$

$$g(z) = \frac{1}{M}\sum_{p=1}^{M}(u^{pT}Z^p\tilde{v}^p + v_0^p)\begin{cases} >0 & \text{then } z \in class+1, \\ <0 & \text{then } z \in class-1 \end{cases} \tag{34}$$

the decision function $g_{MultiVMHKS}$ of the proposed multi-view MultiV-MHKS is the convex combination of the decision functions $g_{MatMHKS}$. It has been proven that for a class of functions $\mathbf{F}$, if $conv\,\mathbf{F}$ is the class of convex combinations of function from $\mathbf{F}$, $-\mathbf{F} = \{-g : g \in \mathbf{F}\}$ [5], then

$$R_n(conv\,\mathbf{F}) = R_n(\mathbf{F}). \tag{35}$$

That is

$$R_n(g_{MultiVMHKS}) = R_n(g_{MatMHKS}) \tag{36}$$

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

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

Further, according to the definition of the Rademacher complexity, Eq. (36) is proven [5].

Secondly, we give the relationship between $R_n(g_{MatMHKS})$ and $R_n(g_{MHKS})$. According to our previous work [21] (Theorem 1), it is known that the solution space for the weights in MatMHKS is contained in that of MHKS, and MatMHKS is a MHKS imposed by Kronecker product decomposability constraint. Therefore, the set of functions $\{g_{MatMHKS}\} \subseteq \{g_{MHKS}\}$. According to the definition of the Rademacher complexity, i.e. (30) and (31), we could get

$$R_n(g_{MatMHKS}) \leq R_n(g_{MHKS}). \tag{37}$$

With (36) and (37), we finally have the relationship among $R_n(g_{MultiVMHKS})$, $R_n(g_{MatMHKS})$, $R_n(g_{MHKS})$ as follows:

$$R_n(g_{MultiVMHKS}) = R_n(g_{MatMHKS}) \leq R_n(g_{MHKS}). \tag{38}$$

Therefore, it can be found that the proposed multi-view method generally has a tighter generalization risk bound than the single-view MHKS.

5.8.3. Complementarity analysis of MultiV-MHKS

This subsection first gives a discussion on multiple solutions $\{u^p,\tilde{v}^p\}_{p=1}^M$ in the MultiV-MHKS. The previous description has stated that the effectiveness of the MultiV-MHKS can attribute to the complementarity between multiple matrix representations of a given single-view pattern. It is known that combining multiple learners can mitigate the limitation of single learner. Here, the unknown function to be approximated might be not present in one vector (MHKS) or matrix (MatMHKS) representation space. But a joint learning of multiple MatMHKS from multiple matrix representation spaces can expand the space of representable functions, likely also embracing the true one. For validating the statement, we explore multiple solutions $\{u^p,\tilde{v}^p\}_{p=1}^M$ in the MultiV-MHKS. For the solution $u^p,\tilde{v}^p$ of each view, we define a solution vector w^p by $w^p=u^p\otimes\tilde{v}^p$. w^p is used to represent the whole solution space of the p-th view in MultiV-MHKS since the solution of each matrix representation is guided by the Kronecker production of u and $\tilde{v}$, which is stated in the last subsection. Thus, we adopt the following formulation (39) for measuring the difference between the solutions of the i-th and j-th views in MultiV-MHKS:

$$D_{ij}=\frac{|w^{iT}w^j|}{\|w^i\|\|w^j\|}, \tag{39}$$

where $0\le D_{ij}\le 1$. Further, the differences among the solutions of the M views in MultiV-MHKS can be defined as $D=(2/M(M-1))\sum_{i=1}^{M}\sum_{j=i+1}^{M}D_{ij}$, where $0\le D\le 1$. It can be found that the larger the D is, the larger difference among all the solutions of the M views in MultiV-MHKS. Table 10 gives the classification performance of MultiV-MHKS, MHKS, the p-value and the D values of the used data sets: Sonar, Wine, SLC, Iris, Echo. and Glass, where the values of $M=2$ for each data set. From this table, it can be found that the data sets such as Iris and Glass which have a clear performance increase correspond to those values of D that are neither too small nor too large. In the case that the value of D is too large, the correlation among the views of MultiV-MHKS is so strong that the solutions of the views are overlapping. But in the case that the value of D is too small, the correlation among the views of MultiV-MHKS is so weak that the solution space expanded by multiple views is too large, where MultiV-MHKS cannot easily find the true solution. Consequently, the weaker correlation between the views of MultiV-MHKS is enough to lead to a performance improvement. The similar phenomenon has been witnessed in another work of us [38].

We further analyses the initializations for $u_1^p,v_1^p,p=1,\ldots,M$ in each view of the MultiV-MHKS. In our experiments, $u_1^p,v_1^p,p=1,\ldots,M$ are initialized with full 1 vectors, i.e. $u_1^p=[1,\ldots,1]^T,v_1^p=[1,\ldots,1]^T,p=1,\ldots,M$. Here, it should be noted that although the elements of $u_1^p,v_1^p,p=1,\ldots,M$ are all the same, the vector sizes of each view u_1^p,v_1^p are different from each other. As shown in Table 4, each view of each data set has different dimensionalities in u_1^p and v_1^p. Taking Wine for example, the vector u_1^p of MultiV-MHKS has the dimension size 1, 2, 3, 4, 6, 12 in each view, respectively and the vector v_1^p has the size 12, 6, 4, 3, 2, 1, respectively. It guarantees that the solutions for each view of MultiV-MHKS would be different.

5.8.4. Reshaping analysis of MultiV-MHKS

In order to more deeply investigate the effect of reshaping, we here design another two reshaping ways besides the original one as shown in Fig. 1. The second reshaping way is to first randomly arrange the feature array of vector or matrix patterns and then again reshape them like Fig. 1. The third reshaping way is defined in such a way that we first convert a given pattern z into one matrix $A\in\mathbb{R}^{s\times t}$, then partition the matrix A into different small matrices $B_i\in\mathbb{R}^{u\times v},i=1,\ldots,l$, and finally arrange the small matrices $B_i,i=1,\ldots,l$ into one new matrix C. This reshaping way is illustrated in Fig. 6. The first, second and third reshaping ways are denoted respectively as "Reshape1(Original)", "Reshape2(Random)" and "Reshape3(Blocking)" in Tables 11 and 12. Both Tables 11 and 12 show the classification accuracies of the MultiV-MHKS with respect to the three different reshaping ways on the data Glass, Wine, Iris, SLC, Echo., Sonar, and the other four data sets with more than 100 attributes: Arrhythmia (452 examples/279 attributes/13 classes), Hill-Valley (606 examples/100 attributes/2 classes), SECOM (1567 examples/590 attributes/2 classes) and Musk (476 examples/166 attributes/2 classes) [24]. Since the dimensionalities of the data sets Glass, Wine, Iris, SLC, Echo. and Sonar are all small, we only apply the third reshaping way to the

Table 10
Complementary analysis of multiple views in MultiV-MHKS.

Data sets	MultiV-MHKS (%)	MHKS (%)	p-value	D value
Sonar	76.67 ± 3.23	75.65 ± 1.95	0.4049	0.1378
Wine	94.43 ± 1.74	94.34 ± 2.66	0.9265	0.1403
SLC	72.86 ± 10.54	68.57 ± 13.12	0.4313	0.4091
Iris	97.73 ± 1.09	93.60 ± 1.63	3.2090e−006	**0.3685**
Echo.	89.40 ± 2.58	87.76 ± 3.20	0.2235	0.5552
Glass	99.24 ± 0.60	87.43 ± 3.69	9.3441e−009	**0.2373**

Table 12
Classification performance (%) comparison for the three reshaping ways of MultiV-MHKS on the non-image data, where the number of views $M=2$.

Different data set	Reshape1 (Original)	Reshape2 (Random)	Reshape3 (Blocking)
Glass	98.80	98.67	–
Wine	93.46	93.42	–
Iris	97.10	95.05	–
SLC	69.53	66.17	–
Echo.	88.16	88.16	–
Sonar	76.92	74.80	–
Arrhythmia	59.92	59.92	59.92
Hill-Valley	75.74	75.74	75.74
SECOM	92.95	92.95	92.95
Musk	78.51	78.51	78.51

Table 11
Classification performance (%) comparison for the three reshaping ways of MultiV-MHKS on MNIST.

The number of views (M)	Reshape1(Original)		Reshape2(Random)		Reshape3(Blocking)	
Size (training, test)	(1000,10 000)	(3000,10 000)	(1000,10 000)	(3000,10 000)	(1000,10 000)	(3000,10 000)
$M=2$	83.92	87.87	83.92	87.87	83.92	87.87
$M=3$	85.84	89.23	85.84	89.23	85.84	89.23
$M=4$	89.12	90.09	89.12	90.09	89.12	90.09

large dimensional data MNIST, Arrhythmia, Hill-Valley, SECOM and Musk. From Tables 11 and 12, it can be found that (i) the three different reshaping ways for MultiV-MHKS yield similar results on most of the used data sets; (ii) MultiV-MHKS with the original feature array shows its advantage over MultiV-MHKS with the random feature array in terms of classification performance on the data sets Iris, SLC and Sonar; (iii) it seems that for these data, different reshaping ways do not result in a significant difference in the performance of MultiV-MHKS here.

6. Conclusions and future work

In this paper, we have developed a new multi-view classifier MultiV-MHKS that is composed of *multiviewization* and a *joint* learning process. It takes MHKS as the base classifier and each corresponding MatMHKS generated from MHKS as one view, and combines all the views into one single learning process. Different from the existing multi-view viewpoint that patterns are represented by multiple independent attribute sets, the proposed multi-view viewpoint is to reshape the original vector representation of the single-view patterns into multiple matrix representations, select one classifier as the base classifier, change the architecture of the base classifier into different ones so as to deal with the corresponding matrix patterns, then take each newly generated one as a view, and finally form a set of classifiers with different views.

It can be found that MultiV-MHKS can well solve the matrixization-dependent problem of MatMHKS [21]. The experiments are done to illustrate the feasibility and effectiveness of the MultiV-MHKS. Firstly, a comparison of MultiV-MHKS with its corresponding single-view classifier MHKS has shown that the proposed classifier has a relatively superior classification performance but just needs a comparable running time. Moreover, we give the theoretical reason why the MultiV-MHKS is better than the single-view MHKS. Secondly, compared with the other ensemble schemes on single-view patterns: sampling pattern and sampling features, the multiviewization in MulitV-MHKS also has an advantage in terms of classification performance. Thirdly, the joint learning of MultiV-MHKS is demonstrated to be better than both the separate and co-training learning on multiple generated matrix views. It should be stated that the compared co-training algorithm with two matrix representations is actually new since it uses the single-view patterns instead of the conventional multi-view patterns. Fourthly, MultiV-MHKS is experimentally illustrated to converge within a few training iterations. Finally, we give a further analysis about the complementarity in the MultiV-MHKS and conclude that the weaker correlation between the solution spaces of all the views in MultiV-MHKS leads to the classification performance improvement.

It is well-known that the outputs of the sub-classifiers should disagree on labeled data in order to get a diversity in ensemble learning [53]. The diversity is supposed to improve performance in ensemble learning [47,51,53]. However, as the above statement, the proposed MVL requires an agreement among the outputs of multiple views as shown in Eq. (19). There are three reasons. Firstly, in our method, the original pattern set is reshaped into multiple different matrix representation sets, which has supplied a diversity in representation level. As Wang and Zhou [52] has stated, the key for the success of disagreement-based approaches is the existence of a diversity, and it is unimportant how the diversity is obtained. Actually, the diversity of our method is achieved through the proposed *multiviewization*. Secondly, the literatures [49,50,54] state that the disagreement of multiple views acts as an upper bound on the generalization error. Therefore, although minimizing the rate of disagreement increases the dependency between the hypotheses and the original motivation for co-training no longer holds, there is still an improved predictive performance of these co-training approaches through minimizing the disagreement, which induces Eq. (3). Thirdly, ensemble learning such Bagging or Boosting [13,48] does not change the original patterns themselves. It just changes the size of the training set and thus needs an additional way to generate diversity. In contrast, our method adopts the matrixized reshaping way. In the literature [53], the ambiguity decomposition is shown as

$$(f_{ens}-t)^2=\sum_t c_i(f_i-t)^2-\sum_t c_i(f_i-f_{ens})^2, \tag{40}$$

where t is the target value of an arbitrary datapoint, $c_i=1, c_i\geq 1$, and f_{ens} is the convex combination of the M component estimators $f_{ens}=\sum_{i=1}^{M} c_i f_i$. Our proposed MVL is given as Eq. (3). They are designed from different viewpoints. Firstly, Eq. (40) falls into ensemble learning and our proposed equation (3) falls into multi-view learning. The former f_i is separately trained but the latter f_p is jointly trained. All the f_p of the proposed equation (3) are boosted each other in training processing. That is the main difference. Secondly, from the form between Eqs. (3) and (40), they are also different. The former is $\sum_t c_i(f_i-t)^2-\sum_t c_i(f_i-f_{ens})^2$, and the latter is $J_{ind}+\gamma J_{com}$.

In this manuscript, we propose the approach called as MultiV-MHKS from the multi-view learning (MVL) point of view based on the following reasons. Firstly, although the proposed method is actually different from the original MVL [9], here we abuse the word "multi-view" since the generated multiple views in this manuscript is artificially abstracted rather than those natural feature sets. Thus, the word "multi-view" of this manuscript can be regarded as the expanded conception in terms of abstract meaning. In practice, we can deal with the pattern in the proposed *multiviewization* way here if the pattern can be represented with multiple ways. It should be emphasized that multiple views of the dealt pattern can be given in either natural or artificial way. It is actually the motivation of our proposed method in this manuscript. Secondly, our previous work [55] sorts patterns into *multi-view patterns* represented by independent sets of attributes and *single-view patterns* represented by only one set of attributes and not properly separated into several distinct sets of attributes. Correspondingly, learning machines can also be sorted into: the single-view machines with only one machine architecture and the multi-view machines with multiple architectures [55]. Naturally, there are four combinations: the single-view machines on the single-view patterns, the single-view machines on the multi-view patterns, the multi-view machines on the single-view patterns, and the multi-view machines on the multi-view patterns [55]. The work proposed in this manuscript falls into the framework of the multi-view machines on the single-view patterns. Thus, we lay the proposed work here on the expanded multi-view learning. Thirdly, the classical MVL such as the co-training method [9] requires two sufficient and redundant views, i.e. two attribute sets, each of which is sufficient for learning and conditionally independent with the other given the class label. Unfortunately, such a requirement can hardly be met in most cases. The requirement of sufficient and redundant views is quite strict, which indeed motivates us to consider the way of solving the strict requirement and to design a new MVL for the single-view patterns in the MVL framework. Fourthly, the MultiV-MHKS reshapes the same vector pattern to different matrix representations which are different from each other in the representation form. But different matrix representations all correspond to the same unique pattern. Each (newly-formed) matrix can be formally viewed as one view of the original vector. Thus, the proposed method learns these artificially generated multi-view patterns, which corresponds to the new MVL of the multi-view machines on the single-view patterns presented in our previous work [55]. Fifthly, the MultiV-MHKS is

a supervised rather than semi-supervised MVL. In the proposed MVL framework, we employ the assumption of the existing MVL framework on labeled patterns. The MultiV-MHKS adopts MatMHKS as the base classifier. MatMHKS has been demonstrated to classify patterns correctly with labeled training set [21]. Thus, the sufficiency assumption can be guaranteed. Different matrices are independent on each other given the class label on the labeled pattern set in the MultiV-MHKS. Through minimizing the disagreement among all views (matrices), the MultiV-MHKS also guarantees the compatibility between the sub-classifier designed from each view.

In future, our work is to (1) further explore how to select an appropriate reshaping way in the proposed multiviewization; (2) generalize the MultiV-MHKS to a much efficient nonlinear method.

Acknowledgment

The authors thank (Key) Natural Science Foundations of China under Grant nos. 61035003, 60903091, and the Specialized Research Fund for the Doctoral Program of Higher Education under Grant no. 20090074120003 for partial support. This work is also supported by the Open Projects Program of National Laboratory of Pattern Recognition and the Fundamental Research Funds for the Central Universities.

References

[2] V. Koltchinskii, Rademacher penalties and structural risk minimization, IEEE Transactions on Information Theory 47 (5) (2001) 1902–1914.
[4] V. Vapnik, A. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, Theory of Probability and its Applications 2 (1971) 264–280.
[5] P. Bartlett, S. Mendelson, Rademacher and Gaussian complexities: risk bounds and structural results, Journal of Machine Learning Research 3 (2002) 463–482.
[8] R. Duin, E. Pekalska, Object representation, sample size and data complexity, in: M. Basu, T.K. Ho (Eds.), Data Complexity in Pattern Recognition, Springer, London, 2006, pp. 25–47.
[9] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Conference on Computational Learning Theory, 1998.
[10] K. Nigam, R. Ghani, Analyzing the effectiveness and applicability of co-training, in: Proceedings of Information and Knowledge Management, 2000.
[11] I. Muslea, C. Kloblock, S. Minton, Active+semi-supervised learning = robust multi-view learning, in: ICML, 2002.
[12] W.C. Lin, F.Y. Liao, C.K. Tsao, T. Lingutla, A hierarchical multiple-view approach to three-dimensional object recognition, IEEE Transactions on Neural Networks 2 (1) (1991) 84–92.
[13] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.
[14] R. Brylla, R. Gutierrez-Osunab, F. Queka, Attribute Bagging: improving accuracy of classifier ensembles by using random feature subsets, Pattern Recognition 36 (2003) 1291–1302.
[15] G. Valentini1, F. Masulli, Ensembles of learning machines, in: M. Marinaro, R. Tagliaferri (Eds.), WIRN VIETRI, Lecture Notes in Computer Science, vol. 2486, 2002, pp. 3–20.
[16] A.K. Seewald, Towards a theoretical framework for ensemble classification, in: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Morgan Kaufmann, 2003, pp. 1443–1444.
[17] T. Windeatt, Accuracy/diversity and ensemble MLP classifier design, IEEE Transactions on Neural Networks 17 (5) (2006) 1194–1211.
[18] B. Igelnik, Y.-H. Pao, S.R. LeClair, C.Y. Shen, The ensemble approach to neural-network learning and generalization, IEEE Transactions on Neural Networks 10 (1) (1999) 19–30.
[19] E. Ho, R.L. Kashyap, An algorithm for linear inequalities and its applications, IEEE Transactions on Electronics Computers 14 (1965) 683–688.
[20] J. Leski, Ho-Kashyap classifier with generalization control, Pattern Recognition Letters 24 (14) (2003) 2281–2290.
[21] S. Chen, Z. Wang, Y. Tian, Matrix-pattern-oriented Ho–Kashyap classifier with regularization learning, Pattern Recognition 40 (5) (2007) 1533–1543.
[22] V. Vapnik, Statistical Learning Theory, John Wiley, New York, 1998.
[23] Z. Wang, S. Chen, New least squares support vector machines based on matrix patterns, Neural Processing Letters 26 (2007) 41–56.
[24] 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.
[25] R.O. Duda, P.E. Hart, D.G. Stock, Pattern Classification, second ed., John Wiley and Sons, Inc., New York, 2001.
[26] Z. Wang, S. Chen, Matrix-pattern-oriented least squares support vector classifier with AdaBoost, Pattern Recognition Letters 29 (6) (2008) 745–753.
[28] Q.S. Xu, Y.Z. Liang, Monte Carlo cross validation, Chemometrics and Intelligent Laboratory Systems 56 (2001) 1–11.
[29] M. Hubert, S. Engelen, Fast cross-validation of high-breakdown resampling methods for PCA, Computational Statistics & Data Analysis 51 (10) (2007) 5013–5024.
[30] X. Geng, D.-C. Zhan, Z.-H. Zhou, Supervised nonlinear dimensionality reduction for visualization and classification, IEEE Transactions on Systems, Man and Cybernetics, Part B 35 (6) (2005) 1098–1107.
[32] T.M. Mitchell, Machine Learning, McGraw-Hill, Boston, 1997.
[33] D. Beymer, T. Poggio, Image representations for visual learning, Science 272 (1996) 1905–1909.
[34] L.F. Chen, H.Y.M. Liao, M.T. Ko, J.C. Lin, G.J. Yu, A new LDA-based face recognition system which can solve the small sample size problem, Pattern Recognition 33 (2000) 1713–1726.
[35] S. Haykin, Neural Networks, A Comprehensive Foundation, second ed., Prentice-Hall, Upper Saddle River, 1999.
[36] S. Chen, Y. Zhu, D. Zhang, J. Yang, Feature extraction approaches based on matrix pattern: MatPCA and MatFLDA, Pattern Recognition Letters 26 (2005) 1157–1167.
[38] Z. Wang, S. Chen, H. Xue, Z.S. Pan, A novel regularization learning for single-view patterns: multi-view discriminative regularization, Neural Processing Letters 31 (2010) 159–175.
[40] Z. Wang, S. Chen, J. Liu, D. Zhang, Pattern representation in feature extraction and classifier design: matrix versus vector, IEEE Transactions on Neural Networks 19 (5) (2008) 758–769.
[41] A. Graham, Kronecker Products and Matrix Calculus: with Applications, Halsted Press, John Wiley and Sons, NY, 1981.
[42] P. Zhang, J. Peng, SVM vs regularized least squares classification, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004.
[43] J. Basak, R. Kothari, Classification paradigm for distributed vertically partitioned data, Neural Computation 16 (7) (2004) 1525–1544.
[44] T. Evgeniou, M. Pontil, T. Poggio, Regularization networks and support vector machines, Advances in Computational Mathematics 13 (1) (2000) 1–50.
[45] K.R. Muller, S. Mika, G. Ratsch, K. Tsuda, B. Scholkopf, An introduction to kernel-based learning algorithms, IEEE Transactions on Neural Networks 12 (2) (2001) 181–202.
[46] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University, 2004.
[47] L.I. Kuncheva, Combining Pattern Classifiers, J. Wiley & Sons, 2004.
[48] 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.
[49] V. de Sa, Learning classification with unlabeled data, in: Proceedings of Neural Information Processing Systems, 1994.
[50] U. Brefeld, T. Gartner, T. Scheffer, S. Wrobel, Efficient co-regularised least squares regression, in: Proceedings of the 23rd International Conference on Machine Learning (ICML'06), Pittsburgh, PA, 2006.
[51] Z.H. Zhou, When semi-supervised learning meets ensemble learning, in: Proceedings of the 8th International Workshop on Multiple Classifier Systems (MCS'09), Lecture Notes in Computer Science, vol. 5519, Reykjavik, Iceland, 2009, pp. 529–538.
[52] W. Wang, Z.-H. Zhou, Analyzing co-training style algorithms, in: Proceedings of the 18th European Conference on Machine Learning (ECML'07), Warsaw, Poland, 2007, pp. 454–465.
[53] G. Brown, J.L. Wyatt, P. Tino, Managing diversity in regression ensembles, Journal of Machine Learning Research 6 (2005) 1621–1650.
[54] D. Hardoon, J.D.R. Farquhar, H. Meng, J. Shawe-Taylor, S. Szedmak, Two view learning: SVM-2K, theory and practice, in: Advances in Neural Information Processing Systems (NIPS'06), 2006.
[55] Z. Wang, S. Chen, Multi-view kernel machine on single-view data, Neurocomputing 72 (2009) 2444–2449.
[56] J.X. Dong, A. Krzyzak, C.Y. Suen, Fast SVM training algorithm with decomposition on very large data sets, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (4) (2005) 603–618.
[57] H.T. Chen, T.L. Liu, C.S. Fuh, Learning effective image metrics from few pairwise examples, in: Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV'05), 2005.

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 an Associate Professor 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 the B.Sc. degree in mathematics from Hangzhou University (now merged into Zhejiang University), Hangzhou, China, in 1983, the M.Sc. degree in computer applications from Shanghai Jiaotong University, Shanghai, China, in 1985, and the Ph.D. degree in communication and information systems from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1997. He was an Assistant Lecturer at NUAA, where since 1998, he has been a Full Professor at the Department of Computer Science and Engineering. He has authored or coauthored over 130 scientific journal papers. His research interests include pattern recognition, machine learning, and neural computing.

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 (INNS). He has published over 50 scientific papers. His research interests are pattern recognition, neural networks, and machine olfactory.