Pattern Recognition 44 (2011) 1162–1171

Contents lists available at ScienceDirect

Pattern Recognition

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

A novel ensemble construction method for multi-view data using random cross-view correlation between within-class examples

Jianchun Zhang, Daoqiang Zhang *

Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

ARTICLE INFO

Article history:
Received 22 January 2010
Received in revised form
2 November 2010
Accepted 13 December 2010
Available online 19 December 2010

Keywords:
Random correlation
Canonical correlation analysis
Feature extraction
Ensemble construction

ABSTRACT

Correlated information between multiple views can provide useful information for building robust classifiers. One way to extract correlated features from different views is using canonical correlation analysis (CCA). However, CCA is an unsupervised method and can not preserve discriminant information in feature extraction. In this paper, we first incorporate discriminant information into CCA by using random cross-view correlations between within-class examples. Because of the random property, we can construct a lot of feature extractors based on CCA and random correlation. So furthermore, we fuse those feature extractors and propose a novel method called random correlation ensemble (RCE) for multi-view ensemble learning. We compare RCE with existing multi-view feature extraction methods including CCA and discriminant CCA (DCCA) which use all cross-view correlations between within-class examples, as well as the trivial ensembles of CCA and DCCA which adopt standard bagging and boosting strategies for ensemble learning. Experimental results on several multi-view data sets validate the effectiveness of the proposed method.

1. Introduction

Learning from multi-view data sets has been one of the active topics in machine learning community. In multi-view setting, data are described by several sets of features. In many real applications, we can easily obtain multiple descriptions of objects, e.g. morphological features and pixels of handwritten digits, urls and caption texts of web images, and so on. It has been shown that learning from multiple representations of data often leads to better performances than combining them into one big view [1,3]. Recently, multi-view learning techniques have also been extended for multi-view clustering [6,7] and multi-view regression [8].

Traditional multi-view methods, such as co-training [1], co-EM [2], are often associated with semi-supervised learning. Those methods train classifiers independently and iteratively on each view, and thus the complementary information between different views can be used. However, correlation information between different views, which may contain useful information, has not been explored. In [5], Zhou et al. proposed to perform semi-supervised learning with very few labeled training examples by taking advantage of correlations between different views. canonical correlation analysis (CCA) [4] is used to extract the correlation information. The authors explained that some helpful information can be obtained by exploiting the correlation between two views by CCA.

CCA is often used to reveal correlation relationships of two sets of features (or views) by projecting two-view data into respective canonical subspaces. However, standard CCA is an unsupervised method, and thus it can not preserve discriminant information in canonical subspaces. Recently, a variant of CCA called discriminant CCA (DCCA) is proposed to exploit discriminant information in feature extraction [17]. In DCCA, besides considering the correlation between two corresponding views of an example, it also uses all the cross-view correlations between within-class examples. It was reported in [17] that usually better performances can be achieved by taking within-class correlation terms into account. In this paper, following DCCA we also incorporate discriminant information into CCA. However, different from DCCA where all the cross-view correlations between within-class examples are used, in our method we use partial random cross-view correlations between within-class examples. Intuitively, there exist redundancy in the cross-view correlations of within-class examples and not all those cross-view correlations are required.

Moreover, because of the random property in choosing cross-view correlations, we can construct a lot of feature extractors based on CCA and random correlation. So naturally, we can fuse those feature extractors for multi-view ensemble learning. In fact, ensemble learning has been a very active area for decades. In ensemble paradigm, multiple learners are combined to solve a problem, where each learner is referred to as a component learner, or base learner. Compared with single learner, ensemble could improve generalization ability dramatically. A key for ensemble learning is to construct both accurate and diverse base learners,

* Corresponding author.
E-mail address: dqzhang@nuaa.edu.cn (D. Zhang).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.12.011

which has been a hot topic in ensemble learning research [9]. Over the years, many algorithms have been developed and widely used, such as *Bagging* [10], *Boosting* [11], *Random Subspace* [12]. However, most existing methods deal with single view data and typically construct diverse learners by manipulating the training examples or perturbing input attributes. Recently, Okun et al. employed multiple views in context of ensemble of nearest neighbor classifiers and trained diverse classifiers based on different subsets of features [14]. In [15], multi-view learning methods are used to mine multi-relational database, and multiple learners created on each view are then validated and combined. In those methods, base learners are trained on different views or different subsets of features, so the correlation information is still not fully used.

In this paper, we proposed a new ensemble method for multi-view data called random correlation ensemble (RCE). In RCE, groups of features are extracted from the random correlation based CCA models to train component classifiers. We compare the proposed RCE method with both CCA and DCCA, as well as the trivial ensembles of CCA and DCCA which adopt standard bagging and boosting strategies for ensemble learning. Experimental results on a series of multi-view data sets, including the Multiple Features data set and the Internet Advisement data set from UCI repository, and ORL, Yale and CMU PIE face databases validate the effectiveness of the proposed method.

The rest of the paper is organized as follows. Section 2 gives a short review of CCA and DCCA. In Section 3, we introduce the proposed RCE method in detail. Then in Section 4, experimental results on several data sets are presented. Finally, we conclude this paper in Section 5.

2. Preliminary

In this section, we give a short review of CCA and discriminant CCA (DCCA), and explain how DCCA increases class separation in multi-modal recognition.

2.1. Canonical correlation analysis

Given data observations with two views, $\mathcal{S}=\{(x_i,y_i)\}_{i=1}^{n}$, corresponding to two random vectors with zero means $x\in\mathbb{R}^p$ and $y\in\mathbb{R}^q$, respectively, where n is the size of the data set. Let $\mathcal{X}$ and $\mathcal{Y}$ denote the respective sample for the two views. CCA seeks to find two sets of directions, one set for each view, such that canonical variables, i.e. projection onto those directions of original variables, would be maximally correlated. Suppose ω_x and ω_y denote a pair of direction for the two views, the problem of CCA can be formulated as

$$\underset{\omega_x,\omega_y}{\operatorname{argmax}}\frac{\omega_x^T C_{xy}\omega_y}{\sqrt{(\omega_x^T C_{xx}\omega_x)(\omega_y^T C_{yy}\omega_y)}} \tag{1}$$

where $C_{xy}=\mathbb{E}[xy^T]$ is between-sets covariance matrix, and $C_{xx}=\mathbb{E}[xx^T]$, $C_{yy}=\mathbb{E}[yy^T]$ is within-sets covariances matrices. Since ω_x, ω_y is scale-independent, Eq. (1) is equivalent to

$$\begin{aligned}&\underset{\omega_x,\omega_y}{\operatorname{argmax}}\quad \omega_x^T C_{xy}\omega_y\\ &\text{s.t.}\quad \omega_x^T C_{xx}\omega_x=1,\omega_y^T C_{yy}\omega_y=1\end{aligned} \tag{2}$$

By applying Lagrangian equation to Eq. (2), the optimization problem of CCA can be converted to generalized eigenvalue decomposition problem, see Eq. (3):

$$\begin{bmatrix} & C_{xy}\\ C_{xy}^T & \end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\end{bmatrix}=\lambda\begin{bmatrix} C_{xx} & \\ & C_{yy}\end{bmatrix}\begin{bmatrix}\omega_x\\ \omega_y\end{bmatrix} \tag{3}$$

Correlated features can be extracted by projecting data onto ω_x and ω_y solved by Eq. (3). The classical CCA can only reveal the linear relationship between feature sets. When dealing with nonlinear problems, kernel version of CCA (KCCA) can be effective [18]. Discussion about KCCA, however, is beyond the scope of the paper.

2.2. Discriminant CCA

Considering that $\mathcal{S}$ contains c classes, $\{\omega_k\}_{k=1}^{c}$, let $\mathcal{X}_k$, $\mathcal{Y}_k$ be the kth class subset of $\mathcal{X}$ and $\mathcal{Y}$. Features extracted by CCA help revealing the hidden relationship between different data views, while it does little to increase class separation. CCA can be considered as multi-view extension of principal component analysis (PCA) [19]. CCA accomplishes dimensionality reduction without knowing the class labels of the training data. It treats all examples in $\mathcal{S}$ fairly, although they may come from different classes, so the classifiers based on the features extracted by CCA can not get exciting accuracies in most cases.

In [17], Sun et al. proposed the discriminant CCA (DCCA). Just like Fisher linear discriminant analysis (LDA) [20], the authors defined within-class correlation and between-class correlation in CCA. Through maximizing within-class correlations and minimizing between-class correlations simultaneously, DCCA improves classification accuracy of CCA effectively. The authors also showed that maximizing within-class correlations was equivalent to minimizing the between-class correlations.

DCCA can be seen as an extension of CCA and can be formulated as

$$\max_{\omega_x,\omega_y}\omega_x^T C_w\omega_y=\max_{\omega_x,\omega_y}\omega_x^T\left(\sum_{k=1}^{c}\sum_{x_i\in\mathcal{X}_k}\sum_{y_j\in\mathcal{Y}_k}x_i y_j^T\right)\omega_y=\max_{\omega_x,\omega_y}\omega_x^T XAY^T\omega_y \tag{4}$$

s.t.

$$\begin{cases}\omega_x^T XX^T\omega_x=1\\ \omega_y^T YY^T\omega_y=1\end{cases} \tag{5}$$

where C_w is called within-class correlation matrix and X and Y are corresponding data matrices for $\mathcal{X}$ and $\mathcal{Y}$. A is a n by n indication matrix, A is defined as

$$A=(a_{ij})_{n\times n}\ \text{and}\quad a_{ij}=\begin{cases}1, & x_i \text{ and } y_j \text{ come from the same class}\\ 0, & \text{otherwise}\end{cases} \tag{6}$$

DCCA can be solved similarly as CCA.

3. Random correlation ensemble

In Section 2.2, we show that DCCA leads to better classification performance through maximizing within-class correlations and minimizing between-class correlations simultaneously. The intuition behind DCCA is that correlations within the same classes should be superior to the correlations of different classes. In this section, we present our method random correlation ensemble (RCE).

Some definitions are given first. Each sum term $x_i y_j^T$ in Eq. (4) is referred to as within-class correlation term if x_i and y_j come from the same class, and between-class correlation term if they come from different classes. From Eqs. (4) and (6), it is clear that DCCA takes all possible within-class correlation terms into account to incorporate class information into CCA to obtain more discriminative features.

In this section, we first extend the classical CCA to canonical random correlation analysis, where a random subset of within-class

correlation terms is included to extract diverse correlated features. Then component classifiers are trained based on the features. Finally, the component classifiers are combined to make predictions.

3.1. Random correlation

One key of ensemble learning is to create diverse and accurate base learner. This section will focus on the topic. First we extend the optimization of classical CCA to canonical random correlation analysis. Suppose $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ are the bootstrapped samples of $\mathcal{X}$ and $\mathcal{Y}$, and $\tilde{x} \in \tilde{\mathcal{X}}$ and $\tilde{y} \in \tilde{\mathcal{Y}}$, canonical random correlation analysis can be formulated as

$$\underset{\omega_x,\omega_y}{\operatorname{argmax}} \omega_x^T \left(\sum_{i=1}^{n} \tilde{x}_i \tilde{y}_i^T \right) \omega_y \tag{7}$$

s.t.

$$\omega_x^T \left(\sum_{i=1}^{n} \tilde{x}_i \tilde{x}_i^T \right) \omega_x = 1 \quad \text{and} \quad \omega_y^T \left(\sum_{j=1}^{n} \tilde{y}_j \tilde{y}_j^T \right) \omega_y = 1$$

Because within-class correlation terms play an important role in separating different classes, here, we require that all correlation term in Eq. (7) must be within-class correlation terms. Suppose $\mathcal{X}_k$ and $\mathcal{Y}_k$ are the kth class subsets of $\mathcal{X}$ and $\mathcal{Y}$. We randomly sample $\mathcal{X}_k$ and $\mathcal{Y}_k$ with replacement, respectively, to produce corresponding bootstrapped samples $\tilde{\mathcal{X}}_k$ and $\tilde{\mathcal{Y}}_k$. In practice, the process will be repeated multiple times, say t times, and t sets of bootstrap samples are generated, i.e. $\tilde{\mathcal{X}}_k^{(l)}$, $\tilde{\mathcal{Y}}_k^{(l)}$, $l=1 \ldots t$, where the superscript l in the parentheses represents the lth bootstrap sample and

$$\tilde{\mathcal{X}}_k^{(l)} = \{\tilde{x}_i^{(l)}\}_{i=1}^{n_k}, \tilde{x}_i^{(l)} \in \mathcal{X}_k$$

$$\tilde{\mathcal{Y}}_k^{(l)} = \{\tilde{y}_j^{(l)}\}_{j=1}^{n_k}, \tilde{y}_j^{(l)} \in \mathcal{Y}_k.$$

where $k=1 \ldots c$. and n_k is the size of $\mathcal{X}_k$ and $\mathcal{Y}_k$.

The problem of canonical random correlation analysis on t sets of bootstrap samples can be formulated as follows:

$$\underset{\omega_x,\omega_y}{\operatorname{argmax}} \omega_x^T \left(\sum_{k=1}^{c} \sum_{l=1}^{t} \sum_{i=1}^{n_k} \tilde{x}_i^{(l)} \tilde{y}_i^{(l)^T} \right) \omega_y \tag{8}$$

A n by n indication matrix R_w is defined to indicate the occurrences of correlation terms, where n is size of $\mathcal{X}$ (or $\mathcal{Y}$) and the (i,j) entry of R_w corresponds to the correlation term $x_i\, y_j^T$. So R_w can be written as a block diagonal matrix, and the kth block R_{w_k} corresponds to the kth class. An example of R_{w_k} may look as follows:

$$R_{w_k} = \begin{bmatrix} 2 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 2 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

where we assume that there are five examples in the kth class and t is set to 3. Sum of all elements in R_{w_k} equals to 15. The values of entries in R_{w_k} can be regarded as the weights of corresponding correlation terms, so R_w is also called weight matrix.

Fig. 1 shows the differences between CCA, DCCA and random correlation from perspective of correlation terms. CCA (left) only consider pairwise correlation terms. DCCA (middle) take all within-class correlation terms into account. Random correlation randomly generates within-class correlation terms, so there may be points who are not contained by any correlation term (the arrow pointing).

Now with help of R_w, Eq. (8) can be rewritten as

$$\underset{\omega_x,\omega_y}{\operatorname{argmax}} \omega_x^T X R Y^T \omega_y \tag{9}$$

where $R = R_w + R_w^T$ is to guarantee symmetry of the correlated relationships, which implies that all symmetric terms, e.g. $xj\, y_i^T$ with respect to $xj\, y_i^T$, are included automatically to enhance correlated relation further.

From Eq. (9), the indication matrix in DCCA can be viewed as a special case of the weight matrix. According to Eq. (3), the above optimization problem can be solved by following generalized eigenvalue decomposition:

$$\begin{bmatrix} & XRY^T \\ YRX^T & \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} = \lambda \begin{bmatrix} XX^T & \\ & YY^T \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} \tag{10}$$

Two feature fusion strategies are available to fuse related features [21]. For any example $(x_i, y_i) \in \mathcal{S}$, we will get the fused feature by

$$\text{(i)}\ \omega_x^T x_i + \omega_y^T y_i$$

$$\text{(ii)}\ \begin{bmatrix} \omega_x^T x_i \\ \omega_y^T y_i \end{bmatrix} \tag{11}$$

Either strategy is usable. In our experiments, we choose the first strategy to fuse correlated features.

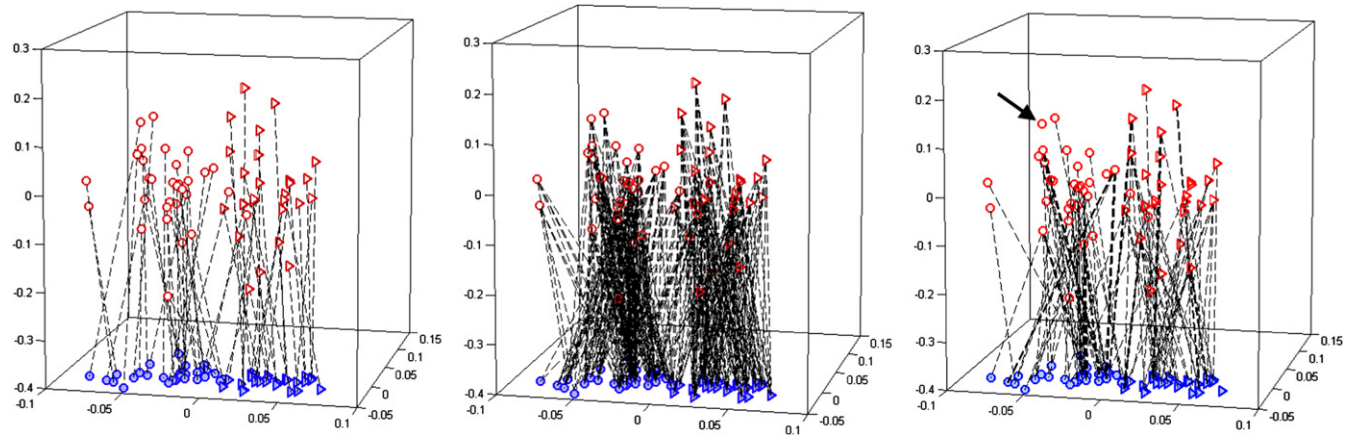

Fig. 1. Differences between CCA, DCCA and random correlation from perspective of correlation terms. The points in three-dimensional space and the points in the bottom two-dimensional plane represent two views, respectively. Different marks (circle and triangle) indicate different classes (two classes). The dashed lines represent correlation terms used in respective method.

Table 1
The RCE algorithm

Input:	Training data $\mathcal{X}=\bigcup_{k=1}^{c}\mathcal{X}_k$, $\mathcal{Y}=\bigcup_{k=1}^{c}\mathcal{Y}_k$, The number of bootstrap times t, Ensemble size h, Subspace dimensionality d,

Begin:	Initialize $\mathcal{H}=\emptyset$
	For $i=1$ **To** h **Do**
	Initialize $R_w=(0)_{n\times n}$;
	Initialize $\tilde{\mathcal{X}}=\emptyset,\tilde{\mathcal{Y}}=\emptyset$;
	For $l=1$ **To** t **Do**
	Generate lth set of bootstrapped samples $\tilde{\mathcal{X}}^{(l)}$ and $\tilde{\mathcal{Y}}^{(l)}$ over all k classes;
	Set $\tilde{\mathcal{X}}=\tilde{\mathcal{X}}\cup\tilde{\mathcal{X}}^{(l)}$, $\tilde{\mathcal{Y}}=\tilde{\mathcal{Y}}\cup\tilde{\mathcal{Y}}^{(l)}$;
	Loop
	Fill R_w according to $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$;
	Set $R=R_w+R_w^T$;
	Obtain d pairs of directions $W_x=[\omega_{x_1}\ldots\omega_{x_d}]$ and $W_y=[\omega_{y_1}\ldots\omega_{y_d}]$ by solving Eq. (10),
	Fuse features according to Eq. (11);
	Train ith component classifier H_i on the fused features;
	Set $\mathcal{H}=\mathcal{H}\cup H_i$;
	Loop

Output:	$\mathcal{H}=\{H_1\ldots H_h\}$,

3.2. Correlation ensemble

Ensemble is a powerful learning paradigm under supervised learning framework, which combines predictions made by all base classifiers to solve specific problem. Ensemble paradigm could effectively improve generalization ability of base classifier. In this subsection we present our method, called random correlation ensemble or RCE for short, where we introduce ensemble techniques in the context of multi-view setting.

As noted in Section 3.1, base classifiers are created through including random within-class correlation terms into CCA. Thus, each classifier investigate the correlated relationships between two views at different levels. Since the correlation between different views provides useful information for describing data and the intrinsic randomness of component classifiers, ensemble of them can help increasing the generalization ability of base classifiers.

The RCE algorithm is summarized in Table 1. The algorithm accept a set of two view samples as input and output an ensemble classifier of size h. For each component classifier H_i, t sets of bootstrap samples are generated over all classes firstly, according to which the indication matrix R_{corr} is filled. Then, d pairs of directions are obtained. Correlated features are extracted by W_x and W_y and fused according to Eq. (11). The component classifier H_i is trained on them. Finally, after constructing ensemble classifier $\mathcal{H}$, all component classifiers are combined by majority voting to make predictions.

4. Experiments

In this section, we evaluate our method RCE on Multiple Features data set and Internet Advisements data set picked out from UCI repository[1] as well as three face databases, YALE,[2] ORL[2] and CMU PIE.[2] We compare RCE with six related methods, i.e.

- *CCA*, *DCCA*: denote the methods which first apply CCA and DCCA, respectively, as a preprocessing step to transform original training data into lower dimensional data, and then train a *single* classifier on the lower dimensional data, without ensemble.
- *bgCCA*, *bgDCCA*: denote the methods which first apply CCA and DCCA, respectively, as a preprocessing step to transform original training data into lower dimensional data, and then perform *bagging* [10,16] on the lower dimensional data, i.e. train base classifiers from bootstrapped samples in lower dimensional space and then combine them by majority voting.
- *bsCCA*, *bsDCCA*: denote the methods which first apply CCA and DCCA, respectively, as a preprocessing step to transform original training data into lower dimensional data, and then perform *boosting* [11] on the lower dimensional data following standard boosting procedures. Also, majority voting is used to combine the outputs of base classifiers.

For all ensemble method, we choose J48 decision tree, an implementation of C4.5 in Weka library [22], Nearest Neighbor classifier and Naive Bayes classifier, respectively, as base classifiers. The implementations of J48, Naive Bayes, Bagging and Boosting are all from Weka library. We use the multi-class version of Boosting in Weka, AdaBoost.M1 [11]. The parameters of those method are kept at their default values in Weka.

For all ensemble methods, the ensemble sizes h, i.e. the number of component classifiers, are all set to 15 in experiments. Their performances with different ensemble sizes are also discussed in this section. The subspace dimensionality determines in what subspace base classifiers will be trained. Generally, performance of subspace method varies considerably with different dimension settings. For all methods, the subspace dimensionality is set automatically in each experiment, such that 95% correlation information is preserved in eigenvalue decomposition processes of those methods.

There is a free parameter in RCE, i.e. the number of bootstrap samples t. The parameter is closely related to classification performances of the base classifiers within ensemble. In each experiment, the optimal value of t is found by searching the parameter space [1,2,…,10] through fivefold cross-validation (CV) on training data. Specifically, for each value of t from [1,2,…,10], we perform a fivefold CV on training data and obtain a mean accuracy across fivefolds. Then we choose the value of t as that with the highest mean accuracy across fivefold CV. Finally, we use the above chosen value of t to train the model on all training data, and use the model to predict the unseen/test data.

4.1. Multiple Features data set

The first data set used is Multiple Features data set, which consists of 2000 examples of 10 handwritten digits ('0–9') with six sets of features, 200 examples for each digit. All of feature sets describe the data set from different views. The six feature set and number of features are listed as follows:

(1) Fourier coefficients of the character shapes (**Fou**, 76)
(2) Profile correlations (**Fac**, 216)
(3) Karhunen–Love coefficients (**Kar**, 64)
(4) Pixel averages in 2 by 3 windows (**Pix**, 240)
(5) Zernike moments (**Zer**, 47)
(6) Morphological features (**Mor**, 6).

Any two of them can be used as two working views. So there will be 15 view settings in total. For each class, we randomly split data set in two halves, and the first half is used for training base classifiers and the second half is for testing. Thus there are 100 training examples and 100 testing examples for each class. Averaged results over 10 independent runs are recorded.

Table 2 shows the accuracies of different algorithms on Multiple Features data set, where bold denotes the highest value at that data

[1] http://archive.ics.uci.edu/ml/
[2] http://www.cs.uiuc.edu/~dengcai2/Data/FaceData.html

Table 2
Recognition accuracies on Multiple Features data set (%).

Base classifier	DATA		CCA	DCCA	bgCCA	bgDCCA	bsCCA	bsDCCA	RCE
J48	**fac**	**fou**	79.67	85.90	84.78	87.85	88.14	88.90	**94.64***
	fac	**kar**	82.47	93.90	89.29	95.43	93.14	96.46	**96.75**
	fac	**mor**	72.83	82.60	75.67	83.46	76.09	83.38	**85.98***
	fac	**pix**	79.85	92.77	87.08	94.72	91.16	96.11	**96.81***
	fac	**zer**	71.95	84.56	78.68	86.46	82.93	87.50	**93.80***
	fou	**kar**	79.46	85.73	86.60	88.68	89.51	90.23	**92.84***
	fou	**mor**	71.79	77.44	76.41	78.90	76.34	78.19	**80.60***
	fou	**pix**	74.53	83.38	81.50	86.27	83.99	88.19	**91.78***
	fou	**zer**	74.47	80.12	79.64	82.08	81.73	83.05	**83.85**
	kar	**mor**	73.49	79.80	77.48	81.97	78.28	80.88	**83.88**
	kar	**pix**	83.13	89.11	89.31	91.61	93.04	93.68	**94.03**
	kar	**zer**	71.67	83.77	81.64	87.02	85.94	88.12	**92.14***
	mor	**pix**	71.54	78.91	73.04	79.95	73.73	79.68	**82.38***
	mor	**zer**	69.12	77.01	72.70	77.48	71.67	75.67	**79.60***
	pix	**zer**	65.88	82.06	76.62	84.28	79.70	86.24	**90.66***
Nearest neighbor	**fac**	**fou**	85.71	89.02	85.91	89.00	84.05	88.00	**95.08***
	fac	**kar**	94.74	97.62	94.54	97.65	92.92	96.75	**97.77**
	fac	**mor**	74.81	82.40	75.23	82.49	73.56	81.41	**87.84***
	fac	**pix**	86.15	97.48	85.69	**97.49**	86.48	96.77	94.64*
	fac	**zer**	85.18	87.84	85.56	87.83	83.62	86.69	**96.79***
	fou	**kar**	89.83	90.16	90.18	90.11	88.21	88.68	**95.75***
	fou	**mor**	73.04	77.33	75.41	77.44	73.85	76.20	**80.21***
	fou	**pix**	75.67	88.88	76.66	88.85	74.46	87.12	**91.99***
	fou	**zer**	81.95	82.59	81.85	82.61	79.92	81.43	**84.78***
	kar	**mor**	75.34	79.87	77.33	79.99	75.58	78.77	**86.34***
	kar	**pix**	93.82	95.17	93.74	95.24	92.80	93.82	**95.67**
	kar	**zer**	90.17	87.65	90.24	87.67	88.69	86.34	**96.42***
	mor	**pix**	71.94	79.10	72.88	79.18	70.56	77.43	**83.99***
	mor	**zer**	68.31	75.19	71.79	75.22	69.60	73.51	**77.44***
	pix	**zer**	83.18	87.32	82.73	87.30	79.65	85.89	**94.26***
Naive Bayes	**fac**	**fou**	88.76	89.26	88.51	89.35	86.69	88.30	**95.41***
	fac	**kar**	94.11	97.67	93.98	97.69	92.94	96.38	**97.74**
	fac	**mor**	77.00	84.18	76.04	84.24	75.83	84.18	**84.34**
	fac	**pix**	93.27	97.62	92.86	**97.61**	90.83	95.93	97.40
	fac	**zer**	84.73	88.80	85.08	88.64	83.91	88.38	**94.83***
	fou	**kar**	91.03	90.46	91.13	90.40	89.28	90.38	**94.18***
	fou	**mor**	75.32	79.59	77.24	**79.62**	76.76	79.59	79.38
	fou	**pix**	86.05	89.20	86.70	89.15	83.35	88.70	**93.24***
	fou	**zer**	82.33	84.26	82.45	84.34	81.71	84.26	**84.64**
	kar	**mor**	74.83	83.24	77.84	83.20	77.93	83.24	**83.29**
	kar	**pix**	93.88	95.75	94.21	95.71	91.90	92.87	**95.83**
	kar	**zer**	88.29	89.94	89.53	89.94	86.14	89.94	**94.67***
	mor	**pix**	75.34	81.30	75.19	81.14	75.25	81.30	**81.31**
	mor	**zer**	70.99	79.57	75.49	**79.69**	75.61	79.57	79.59
	pix	**zer**	83.30	89.07	83.64	88.94	80.21	89.07	**93.30***

set and underline (if applicable) denotes the second highest one. It can be seen from Table 2 that in nearly all cases RCE achieves the best performances among all methods. Table 2 also indicates that the ensemble versions of CCA and DCCA, i.e. bgCCA, bsCCA, bgDCCA and bsDCCA, are superior to CCA and DCCA, respectively, but they are inferior to RCE in most cases. We have also performed a statistical test (paired t test at 95% significance level) between RCE and the best performing method excluding RCE. The star in the last column of Tables 2 indicates that the results between RCE and the best performing method excluding RCE are significant. Tables 2 shows that in most cases RCE performs significantly better than other methods.

In order to study the effect of subspace dimensionality d and ensemble size h on performances of algorithms, accuracies of different algorithms over a series of values of reduced dimensions and under different ensemble sizes are, respectively, shown in Figs. 2 and 3. Three typical view settings are picked, which covers all of six views in Multiple Feature data set. Fig. 2 indicates that in most cases RCE achieves better accuracies than other methods at various dimensions. On the other hand, Fig. 3 shows that RCE is superior to other ensemble methods in most cases and the curves of all methods tend to be steady after the ensemble size is larger than 15.

4.2. Internet Advisement data set

The second data set used in our experiments is the Internet Advisements data set from UCI repository, which represents 3279 web images (459 Ads. and 2820 Non-ads.) with 1558 attributes. All attributes, except four missing value, can be split into five sets covering urls and text descriptions. They are

(1) 472 attributes from ancurl terms, i.e. urls provided by images (**Anc**);
(2) 111 attributes from alt terms, i.e. alternative text descriptions when some errors occur (**Alt**);
(3) 19 attributes from caption terms, i.e. caption texts of images (**Cap**);
(4) 495 attributes from origurl terms, i.e. original or source urls of images (**Org**);
(5) 457 attributes from url terms, i.e. urls of web pages where the images are placed (**Url**).

The task is to predict whether a web image is an advisement. In the experiment, each of the five attribute sets was picked out in

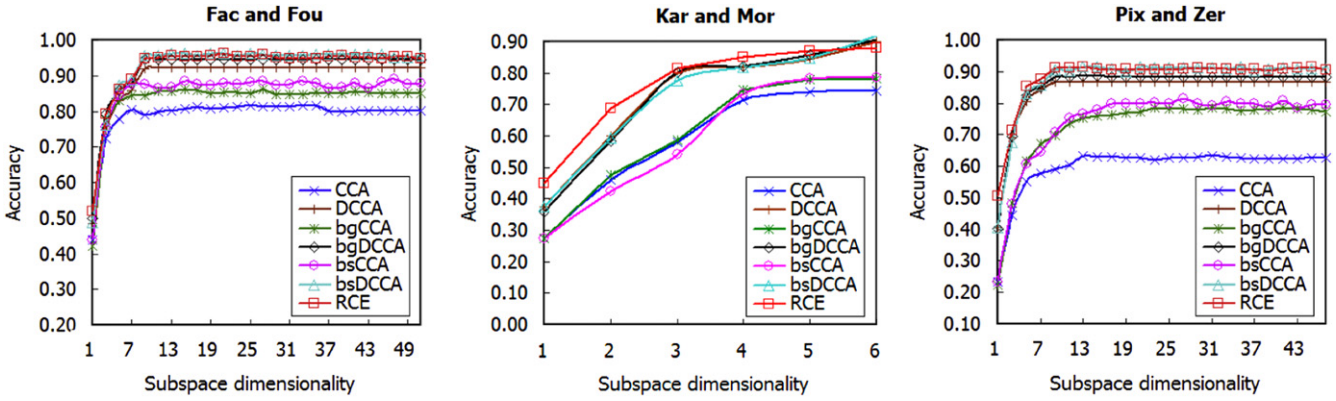

Fig. 2. Recognition accuracies of all methods with different subspace dimensions on Multiple Feature data set. J48 is selected as base classifier and the ensemble sizes of all ensemble methods are set to 15.

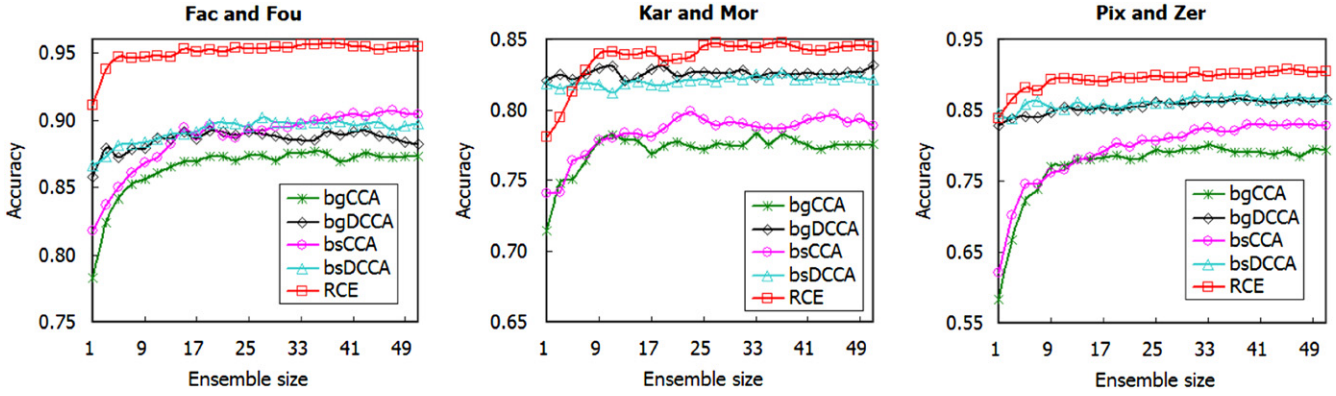

Fig. 3. Recognition accuracies of all ensemble methods with different ensemble sizes on Multiple Feature data set. J48 is selected as base classifier.

Table 3
Recognition accuracies on Internet Advisement data set (%).

Base classifier	DATA	CCA	DCCA	bgCCA	bgDCCA	bsCCA	bsDCCA	RCE
J48	**Anc**	84.47	90.29	88.99	91.04	85.45	90.67	**92.34**
	Alt	89.39	94.39	91.72	95.38	89.21	90.82	**95.41**
	Cap	83.01	94.40	86.20	94.53	83.15	93.45	**95.26**
	Org	86.47	91.86	90.21	91.45	87.77	89.32	**93.27**
	Url	85.67	93.92	90.61	93.40	87.83	89.81	**94.80**
Nearest neighbor	**Anc**	73.47	73.48	73.28	73.46	73.59	73.10	**74.65**
	Alt	76.77	75.54	76.71	75.47	75.02	75.75	**78.00**
	Cap	72.94	74.40	72.95	74.26	71.93	74.03	**79.26***
	Org	75.08	74.10	74.95	74.09	74.66	74.03	**76.45**
	Url	76.08	76.37	76.05	76.35	75.15	77.77	**78.10**
Naive Bayes	**Anc**	87.29	93.18	87.12	93.16	87.11	**93.18**	87.92*
	Alt	92.02	96.31	91.81	**96.34**	88.73	96.31	96.20*
	Cap	90.13	95.02	90.08	**95.08**	87.55	94.53	93.68*
	Org	90.28	93.69	90.00	**93.79**	85.12	93.69	92.14
	Url	92.57	94.89	92.35	95.03	87.58	94.89	**95.37***

turn as the first view, and the remaining sets as the second view. Thus there are five settings in total in this data set. For Ads. or Non-ads, about 230 positive samples and 230 negative samples are drawn randomly from the data set as training set, and the rest as testing set. There are about 460 training samples and 2189 testing samples. The process is repeated with 10 independent runs and the averaged results are recorded.

Table 3 shows accuracies of different algorithms on Internet Advisement data set. As before, in Table 3 we use bold to denote the highest value and underline (if applicable) to denote the second highest one. Also, the star in the last column of Tables 3 indicates the significance test results between RCE and the best performing method excluding RCE. It can be see from Table 3 when using J48 and Nearest Neighbor classifiers RCE achieves better accuracies than other algorithms in all cases but the differences between RCE and the second best method are not significantly in most cases. On the other hand, Table 3 also indicates that when using Naive Bayes as base classifier, RCE is inferior to bgDCCA and bsDCCA in most cases.

4.3. Face recognition data sets

In this subsection, we use three face databases in face recognition experiments, YALE, ORL, and CMU PIE. YALE database contains 165 gray scale images of 15 individuals. There are 11 images for each subject with different illumination conditions and expressions: center/left/right-light, wearing glasses or not, happy, normal, right-light, sad, sleepy, surprised, and wink. ORL, also called AT&T database, consists of 400 images of 40 subjects, 10 images for each subject. These images are photographed in different times, with changing lightning, facial expressions. The size of original image is 92 by 112 pixels, with 256 gray levels per pixel. The CMU PIE database contains a huge collection of face images, under varying pose, illumination and expressions. There are 68 subjects in PIE database, each with 13 different poses, 43 different illumination conditions, and 4 different expressions. A subset with frontal pose (C27) was used in this paper.

Fig. 4. Local binary pattern histogram.

Some preprocessing steps had been done for images in these databases [23,24]. Face areas are cropped, and the size of each cropped image is 64 by 64 pixels, which is used as the first view. Actually images with different resolutions can provide information at different levels and can be regarded as different views of images, thus we resize each image to 32×32 pixels to produce the second view. Then, double Daubechies wavelet transform is performed on all images and the low-frequency images are used as the third view.

We obtain the fourth view of local binary pattern (LBP) histograms, which have been proved to be efficient patterns for representing face images [25]. In the experiments, each image is divided into 4 by 4 local regions firstly, 64×64 pixels for each region, and then LBP histograms were calculated over all 16 regions (see Fig. 4). In Fig. 4, the first image is original 64 by 64 image and the second image is its LBP representation, which is divided into 16 local region (the third image). LBP histograms are calculated over all of the regions. So the dimensionality for the view of LBP histogram is $59 \times 16 = 944$ (more details can be found in [25]).

Table 4
Recognition accuracies on YALE, ORL, and PIE data set (%).

Base classifier	DATA	CCA	DCCA	bgCCA	bgDCCA	bsCCA	bsDCCA	RCE
J48	**YALE1**	36.78	42.78	59.78	64.78	61.11	42.78	**81.78***
	YALE2	39.33	40.56	60.11	58.56	58.22	40.56	**75.22***
	YALE3	41.22	45.44	64.44	67.67	62.33	45.44	**83.33***
	ORL1	34.40	42.00	62.95	71.55	61.30	44.25	**83.20***
	ORL2	41.90	45.80	66.95	74.25	70.50	50.65	**86.05***
	ORL3	33.70	45.05	63.00	74.55	65.30	45.05	**89.10***
	PIE1(10)	53.77	65.61	77.81	85.81	81.42	88.71	**90.63***
	PIE1(15)	60.31	75.24	82.51	89.00	86.95	92.12	**93.74***
	PIE1(20)	64.05	78.38	85.38	91.03	89.19	93.94	**95.05***
	PIE2(10)	50.31	67.25	77.82	85.64	80.85	89.22	**91.39***
	PIE2(15)	58.16	75.80	81.32	88.98	86.40	92.33	**94.18***
	PIE2(20)	62.05	78.89	84.69	91.35	89.21	94.11	**95.21***
	PIE3(10)	44.21	61.84	75.59	85.70	80.23	89.69	**92.70***
	PIE3(15)	53.45	71.74	81.58	90.13	85.78	94.13	**95.66***
	PIE3(20)	58.12	75.52	85.09	91.64	89.63	95.78	**96.72***
Nearest neighbor	**YALE1**	55.11	91.67	55.89	91.89	56.22	91.11	**92.44**
	YALE2	65.11	85.22	65.56	**85.56**	63.22	85.11	83.56
	YALE3	67.78	93.44	68.22	**93.44**	67.33	93.22	92.22
	ORL1	79.80	93.35	79.60	93.45	79.55	92.75	**93.90**
	ORL2	84.00	93.50	84.20	93.40	83.20	92.90	**93.65**
	ORL3	86.75	97.35	86.95	97.40	86.25	97.05	**98.15**
	PIE1(10)	88.21	92.28	88.16	**92.29**	86.20	90.76	91.68*
	PIE1(15)	92.66	**95.34**	92.56	95.31	91.15	94.12	95.03
	PIE1(20)	94.82	**96.47**	94.79	96.40	93.85	95.55	96.46
	PIE2(10)	88.38	92.07	88.30	**92.08**	86.48	90.31	92.05
	PIE2(15)	93.11	95.25	93.07	**95.24**	91.54	93.82	95.23
	PIE2(20)	95.25	**96.60**	95.20	96.56	94.17	95.36	96.60
	PIE3(10)	89.12	96.26	89.25	96.24	86.95	95.57	**96.59**
	PIE3(15)	93.88	98.06	93.98	98.06	92.46	97.75	**98.20**
	PIE3(20)	96.37	98.90	96.42	98.91	95.15	98.59	**98.92**
Naive Bayes	**YALE1**	78.33	73.33	71.22	75.11	77.00	73.33	**89.33***
	YALE2	64.11	67.00	61.22	70.33	63.89	67.00	**81.11***
	YALE3	77.00	77.11	70.22	78.11	73.56	77.11	**89.33***
	ORL1	74.95	76.60	67.90	78.05	73.55	76.60	**91.05***
	ORL2	73.85	82.20	68.00	83.65	72.55	82.20	**91.60***
	ORL3	85.00	91.35	79.60	91.70	85.80	91.35	**95.75***
	PIE1(10)	88.24	91.69	88.09	91.79	87.73	91.62	**93.08***
	PIE1(15)	91.44	93.85	91.52	94.01	90.46	93.79	**94.80**
	PIE1(20)	92.75	95.19	92.78	95.28	91.79	94.32	**95.69**
	PIE2(10)	87.59	91.67	87.24	91.70	86.75	91.51	**92.71***
	PIE2(15)	91.07	94.16	91.12	94.24	90.15	93.73	**94.85**
	PIE2(20)	92.12	95.36	92.35	95.40	91.11	94.55	**95.76**
	PIE3(10)	91.84	94.90	91.23	94.82	91.39	94.90	**96.33***
	PIE3(15)	94.76	97.22	94.74	97.22	94.37	97.22	**97.71**
	PIE3(20)	96.25	98.31	96.39	98.37	95.68	98.31	**98.51**

Because original images are easy to obtain and the other views can be calculated from the original images, the original images (64×64) are always taken as the first view in the experiments, and the other views are taken in turn as the second view. Thus there are three settings for each data set as below:

(1) for original images (64×64) and scaled images (32×32);
(2) for original images and wavelet transformations of images;
(3) for original images and LBP histograms of images.

For YALE and ORL, the data sets are partitioned into equal size training and testing sets randomly. For CMU PIE, which is much larger than the first two, three possible partitions are provided, i.e. 10, 15 and 20 images are picked out randomly to train classifiers respectively, and the remaining images are for testing. Averaged recognition accuracies over 10 independent runs are recorded.

Table 4 shows accuracies of different algorithms on Yale, ORL and PIE data sets, where the definitions of bold, underline and star are the same as before. It can be seen from Table 4 that RCE obtains better accuracies than other methods in most cases, especially when using J48 and Naive Bayes as the base classifiers. As in Section 4.1 on Multiple Features data set, we also plot the effects of the number of reduced dimensions and the ensemble size on performances of algorithms, and the corresponding results are shown in Figs. 5 and 6, respectively, which again validate the effectiveness of the proposed RCE method.

4.4. Further discussions

From the experimental results presented in Tables 2–4, we can see that the best performing methods excluding RCE are mostly bgDCCA, bsDCCA or DCCA. Specifically, our experimental results show that using the bootstrap techniques (including bgCCA, bgDCCA and RCE) can achieve apparent improvements on CCA and DCCA and thus be most beneficial with small and medium sized data (e.g., on face recognition data sets using J48 classifier). On the other hand, when training data are sufficient and representative, the improvements are not so apparent and in some cases may even deteriorate the performances (e.g., on Internet Advisement data set using Naive Bayes classifier). To make a further comparison between RCE and those methods, we evaluate the algorithms' robustness abilities against different levels of noisy class labels. Specifically, we randomly select a fraction of samples from training data and artificially change their class labels with wrong ones. Here we evaluate performances of classifiers using fivefold cross-validation so that training set contains more data. Fig. 7 shows the accuracies of different algorithms under different levels of noisy class labels on Multiple Features data set (Fac and Fou), ORL3 and Yale3. It can be seen from Fig. 7 that RCE is consistently superior to other methods under various levels of noisy labels. Fig. 7 also indicates that bgDCCA has better robustness to noisy labels than bsDCCA because bagging (used in bgDCCA) is more robust to noises than boosting (used in bsDCCA).

Finally, we plot the kappa-error diagrams [13] of RCE and other ensemble methods on Internet Advisements (**Cap**) and PIE database in Fig. 8. The graph was drawn based on 50 component classifiers trained in each method and J48 was used as the base classifiers. For **Cap**, the number of training example is too small to build boosting model with 50 iterations, so only bagging is performed. From Fig. 8 we can see that RCE is an effective ensemble method by making a good trade-off between accuracy and diversity.

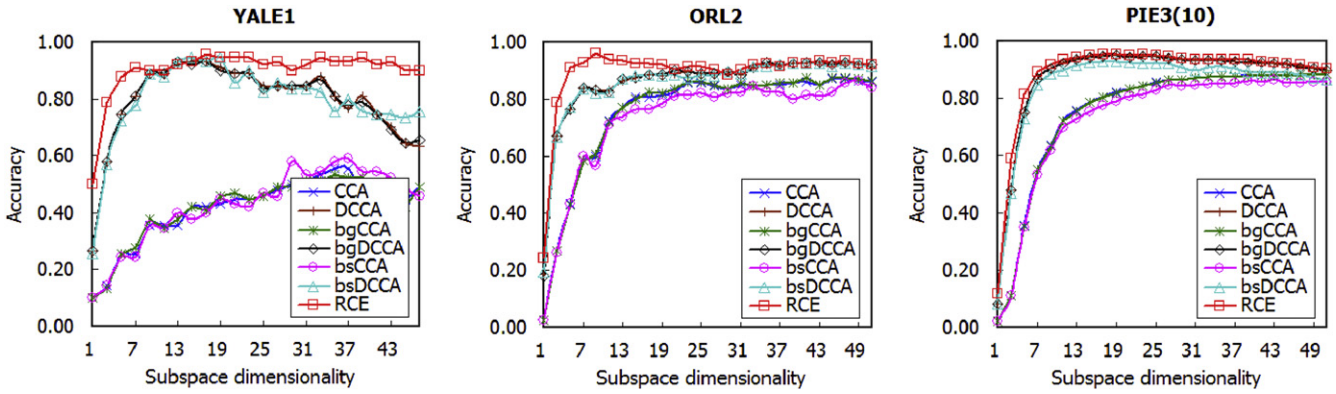

Fig. 5. Recognition accuracies of all methods with different subspace dimensions on the three face databases. Nearest neighbor classifier is used as the base classifier and the ensemble sizes of all ensemble methods are set to 15.

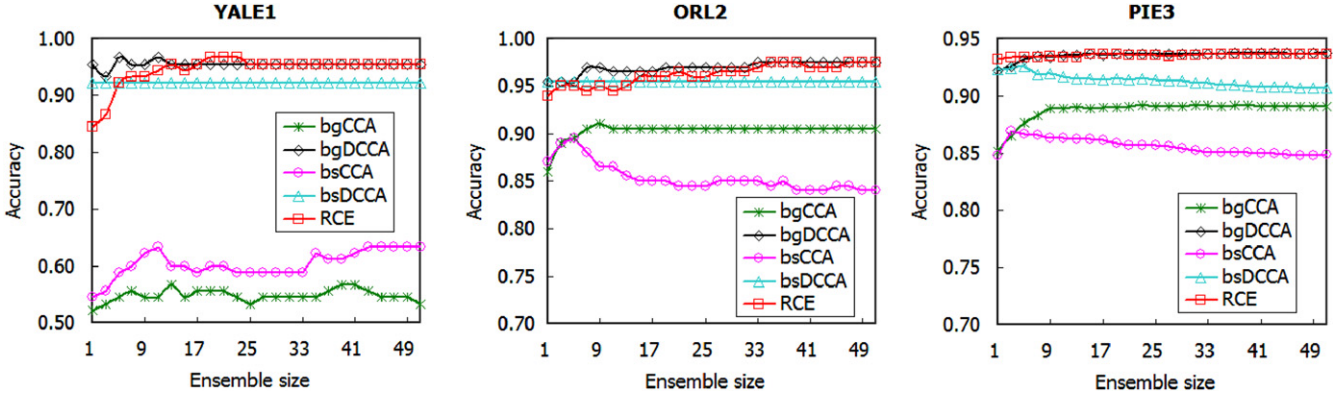

Fig. 6. Recognition accuracies of all methods with different subspace dimensions on the three face databases. Nearest neighbor classifier is used as the base classifier.

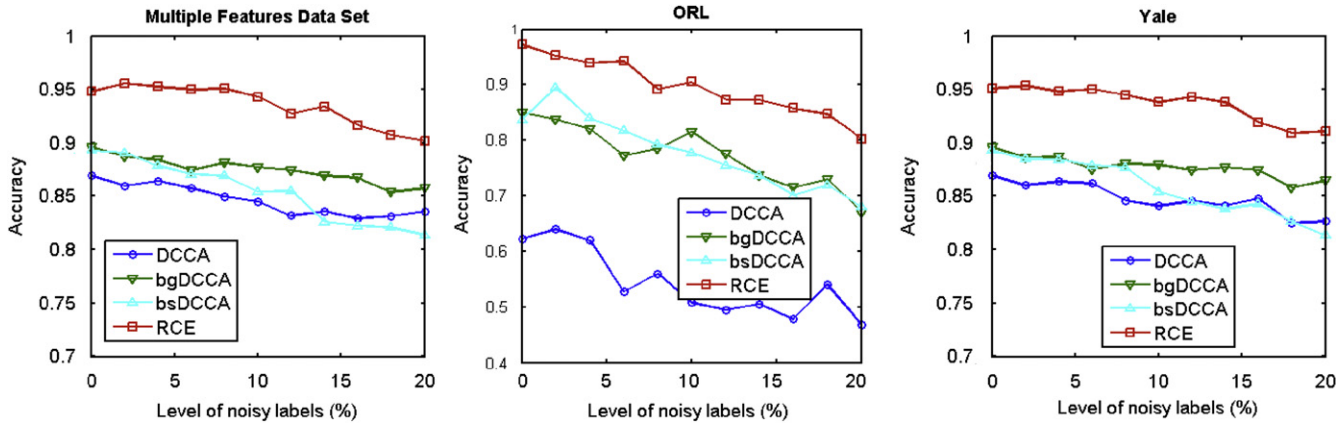

Fig. 7. Recognition accuracies under different levels of noisy class labels.

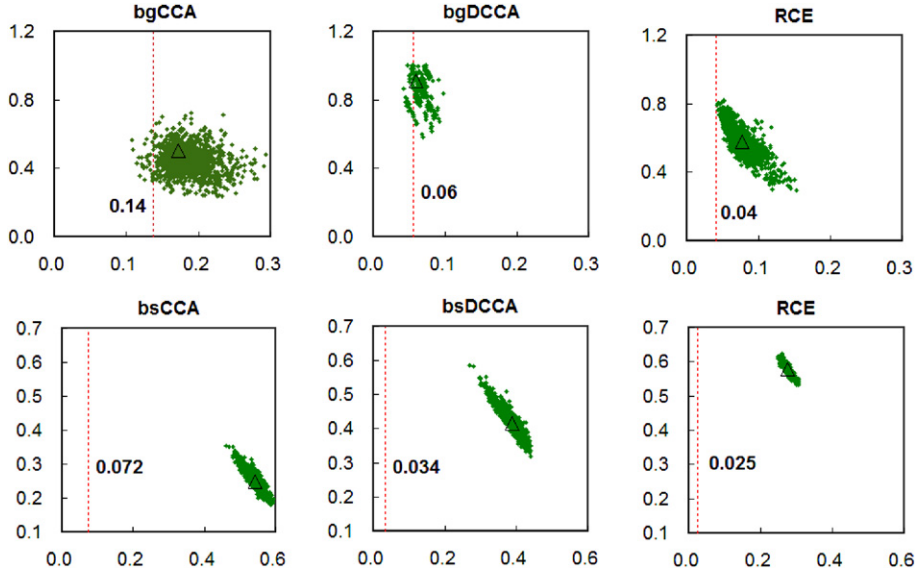

Fig. 8. Kappa-error diagrams for RCE and other ensemble methods on Internet Advisement data set (**Cap**, the top row) and PIE database (the third view combinations, the bottom row). In each plot, *x*-axis and *y*-axis represent pairwise average error and pairwise diversity (kappa) of component classifiers, respectively. The dashed lines indicate the error rates of ensemble classifiers and the up triangles indicate the centroids of the clouds.

5. Conclusions

Ensemble paradigm constructs a set of base learner to solve a specific problem, which can improve generalization ability of single learner significantly. However, most existing ensemble methods focus on single view problem settings. It has not been intensively explored on the topic of multi-view ensemble learning. And existing multi-view ensemble methods create diverse base learners either on various views or subsets of features. In this paper, we propose a new algorithm to construct ensemble classifier in multi-view problems, named random correlation ensemble algorithm (RCE). The base classifiers are created through investigating correlated relationships between different views using canonical correlation analysis (CCA). In RCE, random within-class correlation terms are used to extract diverse correlated features between different views and component classifiers are trained based on the diverse correlated features. Since discriminative information can be preserved by including within-class correlation terms, base classifiers trained on the features can be accurate and diverse. Ensemble of them may help increasing the generalization ability of single classifier. Extensive experimental results on several multi-view data sets validate the effectiveness of RCE.

Acknowledgments

This work is supported by National Science Foundation of China under Grant No. 60875030 and the Open Projects Program of National Laboratory of Pattern Recognition (20090044).

References

[1] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Workshop on Computational Learning Theory, 1998.
[2] K. Nigam, R. Ghani, Analyzing the effectiveness and applicability of co-training, in: Proceedings of Information and Knowledge Management, 2000, pp. 86–93.
[3] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: Proceedings of the Association for Computational Linguistics, 1995, pp. 189–196.
[4] H. Hotelling, Relation between two sets of variables, Biometrica 28 (1936) 322–377.
[5] Z. Zhou, D. Zhan, Q. Yang, Semi-supervised learning with very few labeled training examples, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2007, pp. 675–680.
[6] S. Bickel, T. Scheffer, Multi-view clustering, in: Proceedings of the International Conference on Data Mining, 2004, pp. 19–26.
[7] K. Chaudhuri, S.M. Kakade, K. Livescu, K. Sridharan, Multi-view clustering via canonical correlation analysis, in: Proceedings of the International Conference on Machine Learning, 2009.

[8] S. Kakade, D. Foster, Multi-view regression via canonical correlation analysis, Learning Theory (2007) 82–96.
[9] T.G. Dietterich, Ensemble methods in machine learning, in: Proceedings of the First International Workshop on Multiple Classifier Systems, 2000, pp. 1–15.
[10] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.
[11] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Proceedings of the Thirteenth International Conference Machine Learning, 1996, pp. 148–156.
[12] T.K. Ho, The random subspace method for constructing decision forests, Pattern Analysis and Machine Intelligence 20 (8) (1998) 832–844.
[13] J.J. Rodríguez, L.I. Kuncheva, C.J. Alonso, Rotation forest: a new classifier ensemble method, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1619–1630.
[14] O. Okun, H. Priisalu, Multiple views in ensembles of nearest neighbor classifiers, in: Proceedings of the Workshop on Learning with Multiple Views, International Conference Machine Learning, 2005.
[15] H. Guo, H.L. Viktor, Mining relational databases with multi-view learning, in: Proceedings of the International Workshop on Multi-relational Mining, 2005, pp. 15–24.
[16] Z. Zhou, Y. Yu, Adapt bagging to nearest neighbor classifiers, Journal of Computer Science and Technology 20 (2004) 48–54.
[17] T. Sun, S. Chen, J. Yang, P. Shi, A novel method of combined feature extraction for recognition, in: Proceedings of the International Conference on Data Mining, 2008, pp. 1043–1048.
[18] A. Shotaro, A kernel method for canonical correlation analysis, in: Proceedings of the International Meeting on Psychometric Society, 2001.
[19] T. Diethe, D.R. Hardoon, J. Shawe-Taylor, Multiview fisher discriminant analysis, Technical Report, The NIPS Workshop Learning from Multiple Sources, 2008.
[20] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley-Interscience, 2000.
[21] Q. Sun, S. Zeng, Y. Liu, P.A. Heng, D. Xia, A new method of feature fusion and its application in image recognition, Pattern Recognition 38 (12) (2005) 2437–2448.
[22] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufmann, 2005.
[23] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Advances in Neural Information Processing Systems, 2005, pp. 507–514.
[24] D. Cai, X. He, Y.H. Hu, J. Han, T. Huang, Learning a spatially smooth subspace for face recognition, in: Computer Vision and Pattern Recognition, 2007, pp. 1–7.
[25] T. Ahonen, A. Hadid, M. Pietikainen, Face recognition with local binary patterns, in: European Conference on Computer Vision, vol. 1, 2004, pp. 469–481.

Jianchun Zhang is currently a graduate student of the Department of Computer Science and Engineering at Nanjing University of Aeronautics and Astronautics. His current research interests include pattern recognition and image processing.

Daoqiang Zhang received the B.Sc. and Ph.D. degrees in Computer Science from Nanjing University of Aeronautics and Astronautics, China, in 1999 and 2004, respectively. From 2004 to 2006, he was a postdoctoral fellow in the Department of Computer Science & Technology at Nanjing University. He joined the Department of Computer Science and Engineering of Nanjing University of Aeronautics and Astronautics as a Lecturer in 2004, and is a professor at present. His research interests include machine learning, pattern recognition, data mining, and image processing. In these areas he has published over 40 technical papers in refereed international journals or conference proceedings. He was nominated for the National Excellent Doctoral Dissertation Award of China in 2006, and won the best paper award at the 9th Pacific Rim International Conference on Artificial Intelligence (PRICAI'06). He has served as a program committee member for several international and native conferences. He is a member of Chinese Association of Artificial Intelligence (CAAI) Machine Learning Society.