Pattern Recognition 45 (2012) 1136–1145

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Extract minimum positive and maximum negative features for imbalanced binary classification

Jinghua Wang [a], Jane You [a,*], Qin Li [b], Yong Xu [c]

[a] Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong
[b] Shenzhen University, Guangdong 518055, People's Republic of China
[c] Harbin Institute of Technology, Shenzhen Graduate School, Shenzhen, Guangdong 518055, People's Republic of China

ARTICLE INFO

Article history:
Received 13 March 2011
Received in revised form
11 August 2011
Accepted 7 September 2011
Available online 12 September 2011

Keywords:
Pattern classification
Feature subspace extraction
Imbalanced binary classification
Minimum positive feature
Maximum negative feature

ABSTRACT

In an imbalanced dataset, the positive and negative classes can be quite different in both size and distribution. This degrades the performance of many feature extraction methods and classifiers. This paper proposes a method for extracting minimum positive and maximum negative features (in terms of absolute value) for imbalanced binary classification. This paper develops two models to yield the feature extractors. Model 1 first generates a set of candidate extractors that can minimize the positive features to be zero, and then chooses the ones among these candidates that can maximize the negative features. Model 2 first generates a set of candidate extractors that can maximize the negative features, and then chooses the ones that can minimize the positive features. Compared with the traditional feature extraction methods and classifiers, the proposed models are less likely affected by the imbalance of the dataset. Experimental results show that these models can perform well when the positive class and negative class are imbalanced in both size and distribution.

1. Introduction

As one of the fundamental problems in machine learning, learning from imbalanced datasets has attracted much attention in recent years [1,2]. In this paper, we limit our study on the imbalanced binary classification problem if not specified. The imbalance has at least two forms. One form of imbalance is the number of samples, where one class has much more samples than the other class. Another form of imbalance is that the distributions of different classes are quite different. A typical imbalanced binary classification problem is the task of verification. In this task, the positive class consists of the representations of one object and negative class consists of anything else. It is an imbalanced problem because (1) the positive class has fewer samples than the negative class; (2) the positive samples (representations of one object) form a cluster while the negative samples (which can be anything that different from the positive samples) do not.

Imbalanced data degrade the performances of many dimension reduction or feature extraction methods. When presented with imbalanced datasets, some methods tend to yield feature extractors that favor the majority class, such as principal component analysis (PCA) [3]. The unsupervised PCA seeks the feature extractors that maximize the total scatter. Its feature extractor will be largely determined by the majority class if one class has much more samples than the other class. Some feature extraction methods cannot perform well on imbalanced datasets because they are essentially developed only for the balanced datasets, such as Fisher discriminant analysis (FDA) [4,5]. The supervised FDA aims to maximize the between class scatter and minimize the within class scatter. It is developed based on the assumption that samples from two classes are subjected to Gaussian distributions.

Many standard classifiers tend to favor the majority class on imbalanced data. Support vector machine (SVM) refers to the samples that near boundaries as support vectors and seeks the separating hyperplane that maximizes the separation margin between the hypothesized concept boundary and the support vectors [1]. The SVMs are inherently biased toward the majority class because they aim to minimize the total error. Multilayer perceptron (MLP) is proved to have difficulty in learning from imbalanced datasets [6]. Because of their ability of avoiding the so-called overfitting, the simple and robust linear classifiers are attractive, such as linear discriminant analysis (LDA), minimum square error (MSE), and support vector machine (SVM) [7]. These classifiers make an implicit assumption that the positive and negative classes can be roughly separated by a hyperplane [8]. However, this assumption is violated in many imbalanced datasets where only the positive samples form a cluster, as detailed in

* Corresponding author. Tel.: +852 2766 7293; fax: +852 2774 0842.
E-mail addresses: csjihuwang@comp.polyu.edu.hk (J. Wang), csyjia@comp.polyu.edu.hk (J. You).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.09.004

Section 2. This explains why the performances of these linear classifiers are significantly degraded by the imbalanced datasets.

Different from the discriminative methods (LDA, MSE, and SVM), Gaussian mixture model (GMM) [9] is a generative method. In GMM, the distribution of the samples is modeled by a linear combination of two or more Gaussian distributions [10–12]. GMM has been used in many fields [10–12], and can deal with the imbalanced problem if the parameters of the Gaussian distributions are well fixed. The main difficulty in GMM is to estimate the number of Gaussians to use [13].

This paper proposes a method for imbalanced binary classification. The proposed method seeks feature extractors that can generate minimum positive and maximum negative features in terms of absolute value. In other words, the positive features extracted by a feature extractor are expected to be in an interval $[-\xi,\xi]$, and the negative features fall into $(-\infty,-\xi)\cup(\xi,+\infty)$, where ξ is a positive scalar. This agrees with the situation in a verification task where positive samples cluster together and the negative samples may not. To obtain the feature extractors, this paper proposes two models and designs algorithms to solve these models. While model 1 first minimizes the positive features then maximizes the negative features, model 2 first maximizes the negative features then minimizes the positive features. After projecting the samples onto feature extractors, the proposed method classifies the features based on their weighted distances to the origin.

The advantages of the proposed method are mainly summarized as follows:

Firstly, the proposed method is less likely affected by the imbalanced distributions of the positive and negative classes in two aspects. Different from the traditional feature extraction methods that assume the positive and negative samples cluster together, the proposed method only requires the positive samples cluster together (the negative samples can either cluster together or not). Different from the traditional linear classifiers that require the samples can be roughly separated by a single hyperplane, the proposed method can perform well if two parallel hyperplanes can separate the positive samples from the negative ones.

Secondly, the proposed method is less likely affected by the imbalanced sizes of the positive and negative classes. The positive and negative samples are independently input to two steps in the proposed algorithms. Thus, the two classes have equal power in determining the final feature extractors even though one class may consist of much more samples than the other class.

Thirdly, the proposed method significantly reduces the misclassification of the outliers into positive class. Different from traditional methods that assign two symmetric half-spaces to positive class and negative class, our method assigns two asymmetrical areas to these two classes. As the area for the positive class is much smaller than that of the negative class, the outliers are not likely to be misclassified into the positive class.

The rest of this paper is organized as follows. Section 2 describes the background and motivation. Section 3 presents the proposed method. Section 4 presents the experiments and Section 5 draws a conclusion.

2. Background and motivation

We consider a binary classification problem, where the d dimensional column vectors $x_1,x_2,\ldots,x_{l_1}$ are samples from the positive class with class label $y_i=1(1\le i\le l_1)$ and $x_{l_1+1},x_{l_1+2},\ldots,x_{l_1+l_2}$ from the negative class with class label $y_i=-1(l_1+1\le i\le l)$. The total number of samples is l, where $l=l_1+l_2$. We denote the matrix consists of all the training samples as $X=\begin{bmatrix}x_1 & x_2 & \cdots & x_l\end{bmatrix}$, and the vector consists of all the class labels as $Y=\begin{bmatrix}y_1 & y_2 & \cdots & y_l\end{bmatrix}^T$.

Imbalanced datasets degrade the performance of many feature extraction methods. Due to "curse of dimensionality" [14,15], a feature extraction procedure is necessary in some tasks [16–18]. The subspace-based feature extraction methods [19–24] perform well on balanced data. However, they tend to generate feature extractors that favor the majority class if one class dominates the other.

Imbalanced datasets degrade the performances of many classifiers. After feature extraction, a classifier maps the input feature vector space to the output class label space. In our binary classification problem, the class labels are $+1$ and -1. Most classifiers try to estimate the separate surface of these two classes in some way [25]. Three popular classifiers are K-nearest neighbor (KNN), multilayer perceptron (MLP), and support vector machine (SVM). Though these classifiers perform well on balanced datasets, they are proved to have difficulty in classifying imbalanced datasets [1,6,7].

Because of their ability of avoiding the so-called overfitting, the simple and robust linear classifiers (LDA, MSE, and SVM) are attractive [7]. However, their performances are significantly degraded by the imbalance of the dataset if: (1) the sizes of the positive and negative classes are imbalanced; (2) the samples of one class form a cluster while those of the other class do not. Another problem of these linear classifiers is that they tend to misclassify outliers into positive class. The rest of this section shows two problems of the linear classifiers (on imbalanced datasets), which classify a sample x based on the sign of the value

$$f(x)=x^Tw+w_0 \tag{1}$$

where w is the coefficient vector and w_0 is the threshold.

Firstly, the linear classifiers fail to work if their common implicit assumption does not hold. The goal of a linear classifier is to seek a hyperplane for classification. This hyperplane divides the sample space into two half-spaces, and in them respectively fall the samples of two classes. This goal is achievable only under an implicit assumption that the positive and negative classes can be roughly classified by a single hyperplane. Fig. 1 shows the distribution of face images belonging to three different persons. All of these images are from the Yale face database [26]. In the verification of face 3 in Fig. 1, the negative class consists of two distant subclusters (faces 1 and 2). This violates the above assumption and thus incapacitates the linear classifiers. As the positive class represents a particular object while the negative class represents the whole "rest of the world" in a verification problem [19], it is common that the negative class and positive class are not linearly separable. So, the linear classifiers are usually not applicable in this imbalanced binary classification problem.

Secondly, linear classifiers tend to misclassify outliers. Considering x^Tw+w_0 as the feature of sample x, linear classifiers classify x only based on the sign of this feature. They classify a sample into positive class if and only if it associates with a positive feature. This

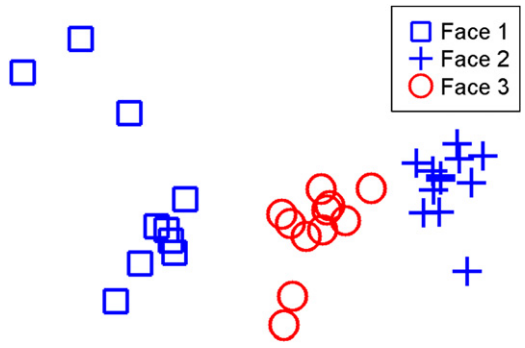

Fig. 1. The distribution of the face images of three different individuals.

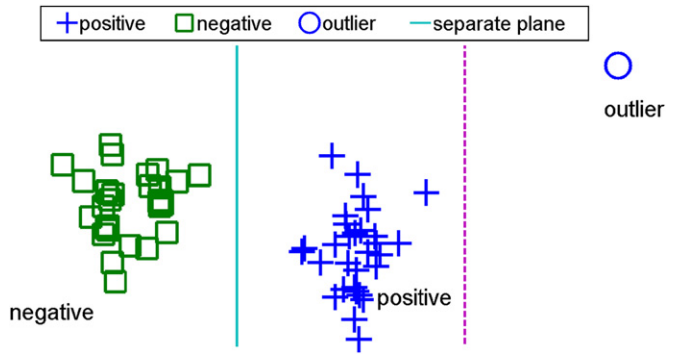

Fig. 2. The classification of an outlier. The outlier is misclassified into the positive class by the solid line.

feature can infinitely approach zero or be infinitely large. However, it is problematic in some situations to classify the testing sample x into the positive class if its feature is too large. Fig. 2 shows an example. In this figure, the crosses ("+") denote the positive training samples and the squares ("□") denote the negative training samples. The solid line (separating hyperplane) can separate the positive samples from the negative samples. Traditional methods consider a novel sample is positive if it lies right to the solid line. Based on this, the circle ("○") representing a testing sample will be classified into the positive class. However, this circle is far away from all positive samples and should be considered as an outlier in the negative class. Such outliers are unavoidable in verification problems, because it is hardly possible to collect a representative training set for the negative class.

Traditional classifiers misclassify the outlier into the positive class mainly because they classify a sample only based on the sign of the feature x^Tw+w_0 and do not take the absolute value of this feature into consideration. One reasonable way to avoid such misclassification is bounding the positive feature from both below and above using two values, instead of only bounding it from below using zero.

3. Proposed method

In this section, we propose a new method for imbalanced binary classification. For simplicity, we consider the samples include an extra dimension with fix value 1 and the threshold w_0 (in Eq. (1)) turns to be an additional dimension of the coefficient vector. Also, as only the direction of the coefficient vector w is important for the classification, we restrict it to have a unit norm. This coefficient vector is also referred to as the feature extractor.

Section 3.1 introduces the basic idea of the proposed method. Sections 3.2 and 3.3 propose two models and algorithms to solve these models. Section 3.4 presents the classification procedure of the proposed method and discussion.

3.1. Basic idea

The principal of the proposed method is to seek a pair of parallel hyperplanes $h^{\pm}(x): w^Tx=\pm\xi$ for classification, as shown in Fig. 3. The positive samples are expected to be clustered in the belt area A defined as follows:

$$A: -\xi \le w^Tx \le +\xi \tag{2}$$

The negative samples are expected to be in the area $\overline{A}$ defined as follows:

$$\overline{A}: w^Tx > +\xi \cup w^Tx < -\xi \tag{3}$$

Compared with the negative samples, the positive samples are nearer to the hyperplane $h^0(x): w^Tx=0$. Different from traditional linear classifiers that assign two symmetric half-spaces to positive class and negative class, our method assigns two asymmetrical areas to these two classes. To reflect the imbalance of the positive class and negative class, our method assigns a "larger" area for the negative class.

Alternatively, we can regard the scalar w^Tx as the feature of sample x after projecting onto the feature extractor w. From Eqs. (2) and (3), we know that the positive features fall into the interval $[-\xi,\xi]$, and the negative features fall into $(-\infty,-\xi)\cup(\xi,+\infty)$. To enlarge the separable, this method seeks the minimum positive and maximum negative features in terms of absolute value for classification.

Ideally, we can obtain the feature extractor w by solving the following l inequalities:

$$\begin{cases} |w^Tx_i| \le \xi & i=1,2,\ldots,l_1 \\ |w^Tx_i| > \xi & i=l_1+1,l_1+2,\ldots,l \end{cases} \tag{4}$$

However, there are three problems in solving these inequalities. Firstly, there is no solution for the inequalities (4) in some cases. The existence of a solution for inequalities (4) means we can classify the training samples using the hyperplanes $h^{\pm}(x): w^Tx=\pm\xi$ with one hundred percent. This is not the case for many real classification problems. Secondly, when inequalities (4) are solvable and have infinite solutions, we have no straightforward criterion to assess the solutions and choose the best ones. Thirdly, solving a set of inequalities as many as the training samples is time consuming.

In the following, we modify the model (4) and generate two new models. By solving the new models, we work out the feature extractors w efficiently.

3.2. Model 1

Model 1 is a special case of the model in Eq. (4) where the parameter ξ is set to be zero. This model minimizes the positive features to be zero and maximizes the negative features, as follows:

$$\max_w \|X_2^Tw\|_2 \ \ s.t. \ \ \|X_1^Tw\|_2=0 \tag{5}$$

where the matrices X_1 and X_2, respectively, consist of all the positive and negative samples.

Note that, model 1 maximizes the norm of the negative feature vector, instead of maximizing the smallest negative feature. If it is necessary to focus on the classification of the boundary samples, we can revise this model as follows:

$$\max_w \underline{f} \ \ s.t. \ \ \|X_1^Tw\|_2=0 \tag{6}$$

where $\underline{f}=\inf\{x_i^Tw | l_1+1 \le i \le l\}$. Because solving Eq. (5) is much faster than solving Eq. (6), we adopt the model in Eq. (5) in this paper. Model in Eq. (5) has open solutions which are detailed in the following paragraphs.

To solve model in Eq. (5) efficiently, we design a two-step procedure. The first step generates a set of candidate feature extractors onto which the positive samples have zero projections. From this set, the second step takes the vectors onto which the negative samples have the maximum projections as the feature extractors.

To generate a set of vectors onto which the positive samples have zero projections, the first step solves the following linear equation system:

$$X_1^TX_2\mu=M\mu=0 \tag{7}$$

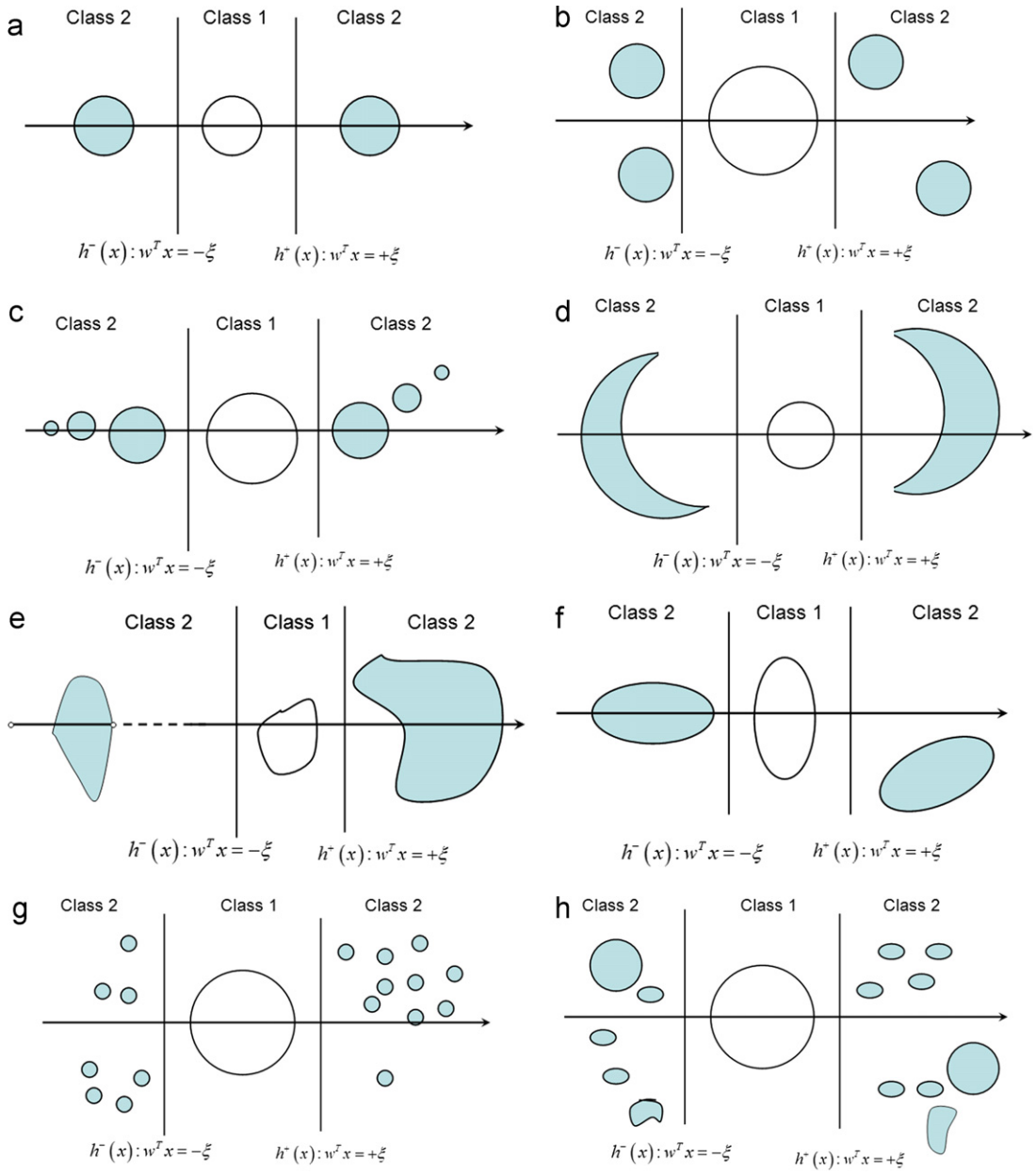

Fig. 3. Separate class 1 (positive) from class 2 (negative) using two parallel hyperplanes.

In verification problems, positive samples are normally fewer than negative samples, i.e. $l_1 < l_2$. The matrix $M = X_1^T X_2 \in R^{l_1 \times l_2}$ has fewer rows than columns. Thus, the linear equation system (7) has a set of nonzero solutions. It can be easily proved that, corresponding to each nonzero solution μ of Eq. (7), the nonzero vector $X_2\mu$ is orthogonal to all the positive samples. If $U = \begin{bmatrix} \mu_1 & \mu_2 & \cdots & \mu_k \end{bmatrix}$ are a set of linearly independent solutions of Eq. (7), we can easily prove that the positive samples have zero projections on the vectors in the following set:

$$S_1 = \{v | v = X_2 U\alpha, \alpha \in R^{k\times 1}\} \tag{8}$$

where α is a coefficient vector.

From the set S_1, the second step chooses vectors that can generate maximum negative projections. The projections of the negative samples onto $v = X_2 U\alpha$ form a vector $X_2^T X_2 U\alpha \in R^{l_2\times 1}$. We can maximize this projection vector as follows:

$$\max \|X_2^T X_2 U\alpha\|_2 = \max \|X_2^T X_2 U\alpha\|_2^2 = \max \alpha^T U^T X_2^T X_2 X_2^T X_2 U\alpha = \max \alpha^T N\alpha \tag{9}$$

The vector α should be the eigenvector of the matrix $N = U^T X_2^T X_2 X_2^T X_2 U \in R^{k\times k}$ corresponding to the leading eigenvalues. As can be seen from Eq. (8), there is a one-to-one correspondence between the α and v. Thus, we can work out the feature extractor v once obtaining the vector α.

In summary, we perform the following algorithm to solve model 1:

Algorithm 1.

Step 1: solve the linear equation system (7) and generate a set of linear independent solutions $\mu_1 \quad \mu_2 \quad \cdots \quad \mu_k$;
Step 2: solve the maximization problem (9) by performing an eigendecomposition procedure; work out the feature extractor $v_i = X_2 U\alpha_i$ where α_i is an eigenvector of the matrix N.

If the dimensionality of the training data is high, step 1 can generate many linearly independent vectors that orthogonal to

the positive samples. Then, step 2 takes the most discriminative vectors as the feature extractors.

3.3. Model 2

We propose the second model as follows

$$\min_{\max\|X_2^T v\|_2} \|X_1^T v\|_2 \tag{10}$$

Among all the vectors v onto which the negative samples have maximum projections, this model picks out the ones onto which the positive samples have minimum projections and takes them as the feature extractors.

We design a two-step procedure to solve this model. The first step generates a set of vectors onto which the negative samples have projections as large as possible. From this set, the second step picks out the vectors onto which the positive samples have minimum projections.

The first step generates a set of vectors onto which the negative samples have maximum projections by solving the following maximization problem:

$$\max_w \|X_2^T v\|_2 = \max_{w,w_0} \|X_2^T v\|_2^2 = \max_{w,w_0} v^T X_2 {X_2}^T v = \max_{w,w_0} v^T P v \tag{11}$$

where $P = X_2 {X_2}^T \in R^{d\times d}$ and $v \in R^{d\times 1}$ is a coefficient vector. The eigenvectors $e_1, e_2, \ldots, e_j$ of the matrix P corresponding to the nonzero eigenvalues are the solution of the maximization problem in Eq. (11). Thus, v should be a in the subspace spanned by these eigenvectors, and in the following set:

$$S_2 = \{v | v = E\beta, \beta \in R^{j\times 1}\} \tag{12}$$

where $E = \begin{bmatrix} e_1 & e_2 & \cdots & e_j \end{bmatrix} \in R^{d\times j}$ and β is the coefficient vector.

The second step picks out vectors from S_2 onto which the positive samples have projections as small as possible by solving the following minimization problem

$$\min \|{X_1}^T E\beta\| = \min \beta^T E^T X_1 {X_1}^T E\beta = \min \beta^T Q \beta \tag{13}$$

where $Q = E^T X_1 {X_1}^T E \in R^{j\times j}$. The solutions of the minimization problem in Eq. (13) are the eigenvectors of the matrix Q corresponding to the minimum eigenvalues. As Q is a semi-positive definite matrix, its eigenvalues are larger than or equal to zero. Denoting the eigenvectors corresponding to the minimum eigenvalues as $\beta_1, \beta_2, \ldots, \beta_h$, we work out the coefficient vectors using $v_i = E\beta_i$.

In summary, we perform the following algorithm to solve model 2

Algorithm 2.

Step 1: solve the maximization problem (11) by eigendecomposing the matrix $P = X_2 X_2^T$ and generate a set of eigenvectors $E = \begin{bmatrix} e_1 & e_2 & \cdots & e_j \end{bmatrix}$ corresponding to the maximum eigenvalues;

Step 2: solve the minimization problem (13) by eigendecomposing the matrix $Q = E^T X_1 {X_1}^T E$ and generate a set of eigenvectors $\beta_1, \beta_2, \ldots, \beta_h$ corresponding to the minimum eigenvalues; calculate the feature extractors using $v_i = E\beta_i$.

3.4. Classification and discussion

Projecting the samples onto the feature extractors $v_i (i = 1,2,\ldots,n)$ output by Algorithms 1 and 2, we can obtain minimum positive features and maximum negative features. Ideally, the positive features are within the interval $[-\xi_i, -\xi_i]$ and the negative features within $(-\infty, -\xi_i) \cup (\xi_i, +\infty)$. If the proposed method generates only one feature extractor, the positive features are near to the origin and negative features are far away from the origin, as shown in Fig. 4(a).

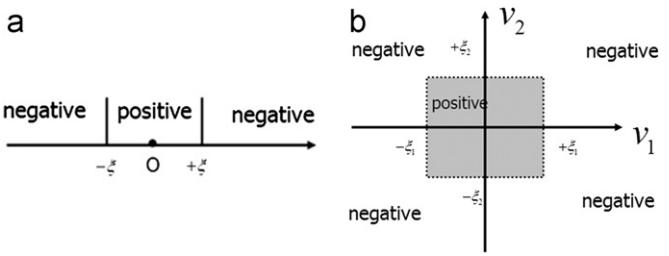

Fig. 4. The projections of samples onto feature extractors: (a) one feature extractor; (b) two feature extractors.

The features of these two classes can be separated by two points. If the proposed method generates two feature extractors, the positive features are in a rectangle and the negative features are out of the rectangle as shown in Fig. 4(b). The situation is similar when we have more feature extractors. In Fig. 4, the positive and negative features are expected to lie in two asymmetrical areas of the feature space. While positive features cluster together, the negative features can be scatter anywhere else. This agrees with the situation in verification problems and avoids the two problems mentioned in Section 2. Firstly, as the proposed method does not require the negative samples cluster together, it can perform well when the negative class consists of a number of distant subclusters. Secondly, as the proposed method confines the positive samples in a small area, it can correctly classify an outlier into the negative class.

The feature extraction results of a test sample x form a vector $z = \begin{bmatrix} z_1 & z_2 & \cdots & z_n \end{bmatrix}^T$, where z_i is the projection of x onto the feature extractor v_i. We can classify the feature z based on its distance to the origin. In this paper, we adopt the weighted block distance as follows:

$$d(x) = \sum_{i=1}^{n} w_i |z_i| = \sum_{i=1}^{n} w_i |v_i^T x| \tag{14}$$

where w_i is the weight for the ith feature extractor. If this distance is larger than a threshold, x is classified into the negative class; or else, it is classified into the positive class.

The same to the other methods, the proposed method takes into account both the positive and negative samples in the training stage. However, different from the traditional methods that input the positive and negative samples concurrently, the proposed method inputs one class after the other. This keeps our method away from the influence of the imbalanced sizes of the positive and negative classes. In the proposed two-step algorithms for the two models, either the positive or negative class is independently input to one step. Even if the training set is imbalanced, the majority class cannot dominate the minority class.

Taking the one-against-others strategy, we can extend the proposed method to deal with the c-class problem (c is the number of classes), as follows:

Training procedure: (generate feature extractors for each class)
For each $1 \le l \le c$, take the samples in the lth class as the positive samples and the rest as the negative samples; perform Algorithm 1 or 2 to generate the feature extractors $v_1^l, v_2^l, \cdots, v_{k_k}^l$ for the lth class.
Classification procedure (classify the test sample x)
For each $1 \le l \le c$, calculate $d_l(x) = \sum_{i=1}^{k_l} w_i |(v_i^l)^T x|$ and classify x into the jth class if $d_j(x) = \min_{1\le l\le c} d_l(x)$.

4. Experiments

In this section, we first compare our method with different classifiers (back propagation, GMM, and five different forms of SVM)

on two synthetic datasets in Section 4.1 and object verification in Section 4.2. Then, we compare our method with different feature extraction methods (FDA, PCA, and LPP) on face verification in Section 4.3. The experimental results validate the feasibility of the proposed method.

4.1. Synthetic data classification

The first dataset is drawn from three 2-dimensional random vectors Ω_0, Ω_1 and Ω_2, each of which has Gaussian distribution with covariance matrix of *diag*{1,3}. The mean of the first random vector Ω_0 is (0,0), and those of Ω_1 and Ω_2 are respectively (5,1) and $(-5,1)$. We draw 50 positive training samples from Ω_0, and draw 500 negative training samples respectively from Ω_1 and Ω_2. Thus, the training set consists of 50 positive samples and 1000 negative samples. Fig. 5 shows the distribution of the training samples. The testing set also consists of 50 positive samples drawn from Ω_0, and 1000 negative samples drawn from Ω_1 and Ω_2.

The second dataset is drawn from five 2-dimensional random vectors Ξ_0, Ξ_1, Ξ_2, Ξ_3, and Ξ_4, each of which has Gaussian distribution with covariance matrix of *diag*{0.1,0.1}. The means of these five random vectors are respectively (0,0), $(-1.5,-1.5)$, $(+1.5,-1.5)$, $(-1.5,+1.5)$, $(+1.5,+1.5)$. We draw 20 positive training samples from Ξ_0, and draw 40, 400, 40, 400 negative training samples respectively from Ξ_1, Ξ_2, Ξ_3, and Ξ_4. Thus, the training set consists of 20 positive samples and 880 negative samples. Fig. 6 shows the distribution of the training samples. The testing set also consists of 20 positive samples from Ξ_0, and 40, 400, 40, 400 negative samples respectively drawn from Ξ_1, Ξ_2, Ξ_3, and Ξ_4.

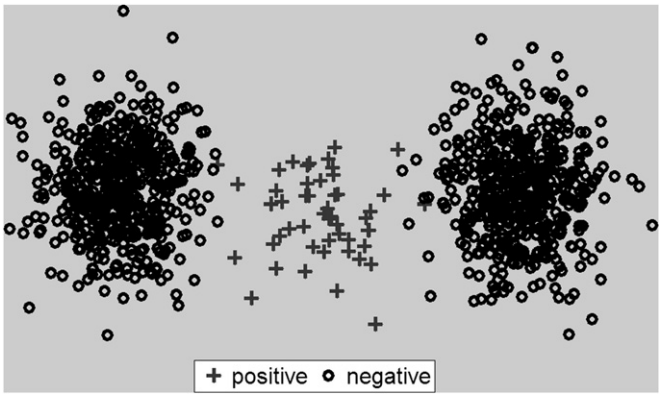

Fig. 5. The distribution of the first synthetic dataset.

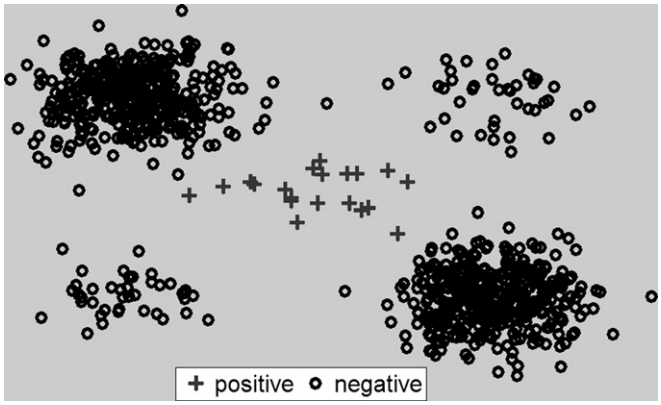

Fig. 6. the distribution of the second synthetic dataset.

Both of these two synthetic datasets consist of imbalanced positive and negative samples because of two reasons. Firstly, the positive samples are much fewer than the negative samples. The minority class only account for 4.76% in the first dataset and 2.22% in the second dataset of all the training samples. Secondly, the distributions of the positive and negative samples are quite different. While the positive samples are drawn from a single random vector, the negative samples are drawn from no less than two random vectors. While the positive samples form a single cluster, the negative samples form no less than two clusters.

In a binary classification problem, a true positive (TP) means a positive sample is correctly classified into the positive class and a true negative (TN) means a negative sample is correctly classified into the negative class. We use true positive rate (TPR) and true negative rate (TNR) to evaluate the performance of different methods. TPR is the ratio between the number of TP and that of all the positive samples and TNR is the ratio between the number of TN and that of the negative samples.

We compare our method with Multilayer perceptron (MLP) [6], which is a popular artificial neural network, and four different forms of SVM: linear support vector machine (LSVM) [8], Gaussian support vector machine (GSVM) [8], polynomial support vector machine (PSVM) [8], and one-class support vector machine (OSVM) [27]. We also compare our method with GMM [9]. In GMM, we suppose the numbers of Gaussian distributions are known (3 for the first dataset and 5 for the second dataset). Table 1 lists the performances of these methods on the synthetic datasets.

Table 1 shows that each of the listed methods has a TNR higher than 95%. It indicates that these methods can correctly classify the majority negative samples. However, many minority samples are misclassified by LSVM, GSVM, PSVM, OSVM, and MLP. The TPRs of them are lower than 90% on both of these datasets. It indicates that these methods are biased toward the majority negative class on these imbalanced binary classification problems. Because the samples are drawn from random vectors that follow the Gaussian distributions and the number of these distributions are known, GMM achieves both high TPRs and TNRs.

Our two models are robust to the imbalances in size and distribution and achieve high TPRs as well as high TNRs (larger than 92%). As can be seen from Algorithm 1 and Algorithm 2, the positive and negative samples are input independently in two steps. Thus, the feature extractors are not affected by the imbalanced class sizes. Also, as the proposed models only require the positive samples cluster together, they can achieve good performances on these datasets where the negative samples are drawn from several random vectors.

4.2. Object verification

We perform the experiment of object verification on the Columbia University Image Library (COIL20) [28]. It contains 20 objects. The images of each object are taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. The size of each image is 32×32 pixels, with 256 gray levels per

Table 1
The performance (TPR and TNR) of different methods on synthetic datasets (%)

Methods	LSVM	GSVM	PSVM	OSVM	MLP	GMM	Model 1	Model 2
The first dataset								
TPR	86.0	86.0	89.0	62.0	32.7	98.0	98.0	99.7
TNR	99.8	99.8	99.3	99.9	97.4	99.6	95.2	99.5
The second dataset								
TPR	13.6	85.0	77.5	68.0	60.8	95.0	98.7	92.6
TNR	99.8	99.8	99.7	99.9	98.9	99.6	99.5	99.8

pixel. Thus, each image is represented by a 1024-dimensional vector. Fig. 7 shows twenty example images, one for each object.

In our experiment, the training set consists of 14 images for each of the first 10 objects, totally 140 images. In the verification of a particular object, its images are positive and the images of the other objects are negative. Thus, we have 14 positive and 126 negative training images. The testing set consists of 58 images of the first 10 objects and 72 images of the last 10 objects. The configuration of the training and testing set are shown in Table 2. Because there are 10 classes in this experiment, we set the number of Gaussian distributions in GMM to be 10. We have also tested GMM by setting the number of Gaussian distributions to be 20 and 100, the experimental results are similar to those with 10 distributions.

From the experimental results listed in Table 3, we know that the TPRs and TNRs of the proposed two models always rank in the top three in each column, except TNR of model 1 in class 3 and TPR of model 2 in class 8. In class 3, LSVM and MLP achieve higher TNR than model 1, however, their TPRs (51.7% and 67.6%) are much lower than that of model 1 (83.3%). Though model 2 has smaller TPR (97.1%) than LSVM and OSVM (98.4% and 99.4%) in class 8, its TNR (100%) is higher than theirs (90.5% and 98.5%). When verifying class 8, we can use model 2, which can achieve highest TPR and TNR. Generally, LSVM, PSVM, GSVM, GMM, and MLP achieve much higher TNR than TPR. This indicates that these methods are highly affected by the imbalance of the training data. Though OSVM is not highly affected by the imbalance and achieves comparable TPR and TNR, its TNR is normally smaller than the other methods.

To compare the accuracies of the two proposed models and the six traditional methods, we conduct a series of Wilcoxon signed-rank tests at 1% significance level. We devote each Wilcoxon signed-rank test to determine whether the accuracies of a proposed model are statistically different from those of one traditional method. In all of the 12 tests, the results indicate that the differences between the proposed models and the traditional methods are statistically meaningful.

In these experiments, the images of the last 10 objects are only included in the testing dataset and all of them are considered as the negative samples. As they are not included in the training dataset, some of them are misclassified into the positive class by traditional methods. It is a similar situation to the outlier in Fig. 2. However, by confining the positive samples in a relatively small area, the proposed models greatly reduce such misclassifications.

4.3. Face verification

One standard face database is the Carnegie Mellon University Pose, Illumination and Expression database (CMU PIE database) [29]. The CMU PIE database totally consists of more than 40,000 facial images of 68 people. In the construction of this database, the images of each individual are captured under 43 different illumination conditions, across 13 different poses, and with 4 different expressions.

Fig. 7. Examples in the COIL20 database.

Table 2
The configuration of the training and testing set in the verification of one object.

	Images of the first 10 objects	Images of the last 10 objects
Training set	140 images (14 positive and 14×9 negative samples)	0
Testing set	580 images (58 positive and 58×9 negative samples)	72×10 negative samples

Table 3
Experimental results (TPR and TNR) of object verification on COIL 20 (%).

	Class 1		Class 2		Class 3		Class 4		Class 5	
	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR
LSVM	87.9	**99.7**	68.9	**99.6**	51.7	**99.4**	**98.3**	**99.9**	50.0	**99.2**
PSVM	67.9	91.8	63.7	97.8	46.5	92.4	74.4	**100**	68.2	89.5
GSVM	51.0	91.2	72.4	84.6	75.8	89.2	55.2	97.3	**79.3**	86.5
OSVM	**90.6**	98.4	**84.5**	96.6	**84.2**	93.8	94.3	96.7	75.7	88.3
MLP	87.9	98.7	51.7	99.5	67.6	**99.7**	86.2	99.6	48.3	99.0
GMM	19.8	96.3	20.0	89.2	33.6	88.9	59.2	85.6	14.5	94.7
Model 1	**91.4**	**99.8**	**93.1**	**100**	**83.3**	99.3	**100**	**100**	**82.8**	**99.7**
Model 2	**96.6**	**100**	**96.6**	**100**	**84.5**	**99.5**	**98.2**	**99.9**	**86.2**	**99.9**
	Class 6		Class 7		Class 8		Class 9		Class 10	
	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR	TPR	TNR
LSVM	34.8	98.8	87.9	98.8	**98.4**	90.5	44.8	**99.9**	91.4	99.7
PSVM	65.5	94.5	74.1	95.4	87.9	99.3	69.0	97.9	87.5	99.4
GSVM	**93.1**	73.5	63.8	**100**	50.0	**100**	77.6	85.9	78.0	**100**
OSVM	86.6	92.5	90.4	97.2	**99.4**	98.5	**95.7**	96.8	**92.5**	99.7
MLP	68.0	**99.4**	**91.4**	99.5	53.5	99.1	56.9	99.8	48.3	99.7
GMM	25.0	86.1	27.9	82.3	92.4	84.3	25.0	83.8	32.6	85.0
Model 1	**87.5**	**99.8**	**100**	**100**	**100**	**100**	**100**	**100**	**95.7**	**100**
Model 2	**98.3**	**100**	**96.6**	**100**	97.1	**100**	**94.8**	**100**	**98.0**	**99.8**

Fig. 8. Examples in the CMU PIE database.

We use a subset contains the face images of 10 individuals under five poses (C05, C07, C09, C27, C29) and all different illuminations and expressions. Fig. 8 shows 40 face images of a person. There are 170 images for each of the 50 individual. The training set consists of 30 images of each individual, and the test set consists of the rest images.

We verify each of the ten individuals 10 times with independent training sample set, and list the TPR, TNR, and D (number of feature extractors) in Table 4. In the verification of one individual, positive samples are his/her face images and negative samples are images of the others. In this experiment, we compare our feature extraction methods with other methods, including PCA [3,20], FDA [4,5], and LPP [30]. The features extracted by our method are classified based their weighted distances to the origin, as shown in Eq. (14). The features extracted by other methods are classified using the GSVM.

The feature extractors in PCA are eigenvectors of the total scatter matrix corresponding to all the nonzero eigenvalues. With 300 training samples, we obtain 299 feature extractors at the most. The number of feature extractors in FDA is $c-1$, where c is the number of classes. In a verification task, there are two classes and we have only one FDA-based feature extractor. The LPP-based feature extractors are obtained by solving a generalized eigenvalue problem. We keep all the PCA and LPP feature extractors for dimension reduction. When solving the maximization problem (9) in model 1 and Eq. (11) in model 2, we keep the feature extractors that account for 95% of the spectrum.

Generally, the proposed models achieve higher TPR and TNR when verifying these 10 faces, as shown in Table 4. They achieve higher both TPR and TNR than the other methods on face 2 and face 3. In some cases, the other three feature extraction methods can achieve higher TNR than the proposed models. However, none of them can achieve higher TPR. In other words, compared with the proposed method, the other methods are more likely to misclassify the samples in the minority class. This indicates these methods are affected by the imbalance of the training dataset. The Wilcoxon signed-rank tests (with the same settings in Section 4.2) demonstrate that the differences of classification accuracies are statistically meaningful. Also, the proposed models have fewer feature extractors than PCA and LPP. The numbers of feature extractors in model 1 are no larger than 8 and those in model 2 are no larger than 19, while PCA and LPP respectively have 299 and 30 feature extractors. In summary, this table shows that the proposed method achieve higher classification accuracy using fewer feature extractors.

5. Conclusion

This paper proposes a method for extracting minimum positive and maximum negative features in terms of absolute value for imbalanced binary classification. Corresponding to each feature extractor is a pair of parallel hyperplanes to separate the positive samples from the negative ones, as shown in Fig. 3. This differentiates our method from the traditional linear classifiers that try to separate the samples using a single hyperplane. To obtain the feature extractors, this paper presents two models. Model 1 first generates a set of candidate feature extractors that can minimize the positive features to be zeros, and then chooses the ones among these candidates that can maximize the negative features. Model 2 first generates a set of candidate feature extractors that can maximize the negative features, and then chooses the ones among these candidates that can minimize the positive features.

In our experiments, while the positive samples are representations of one object, the negative samples are representations of

Table 4
Experimental results of face verification on CMU PIE subset TPR and TNR (%), and D (number of feature extractors).

	Face 1			Face 2			Face 3			Face 4		
	TPR	TNR	D	TPR	TNR	D	TPR	TNR	D	TPR	TNR	D
PCA	80.7	**98.1**	299	72.9	96.0	299	92.1	95.9	299	79.2	**99.4**	299
FDA	37.8	95.2	1	27.9	87.4	1	39.3	91.6	1	35.3	93.8	1
LPP	71.4	96.2	30	85.0	93.9	30	71.4	96.3	30	69.3	**97.9**	30
Model 1	**92.1**	97.3	7	**93.6**	**98.1**	8	**97.1**	**98.0**	3	**92.1**	95.9	5
Model 2	**92.9**	**97.5**	19	**92.9**	**96.3**	17	**96.4**	**97.2**	19	**97.1**	90.0	12
	Face 5			Face 6			Face 7			Face 8		
	TPR	TNR	D	TPR	TNR	D	TPR	TNR	D	TPR	TNR	D
PCA	81.4	**99.0**	299	85.0	**98.2**	299	76.4	**94.8**	299	84.3	**96.2**	299
FDA	22.9	89.7	1	42.4	92.2	1	22.9	88.3	1	29.3	93.9	1
LPP	79.3	**99.1**	30	82.8	97.2	30	88.6	93.9	30	84.3	95.7	30
Model 1	**94.3**	90.1	6	**99.3**	96.1	3	**92.1**	**95.2**	8	**91.4**	**95.8**	3
Model 2	**92.9**	91.3	18	**96.4**	**98.9**	14	**92.9**	92.7	9	**93.5**	95.7	19
		Face 9						Face 10				
		TPR		TNR		D		TPR		TNR		D
PCA		70.8		**96.3**		299		**97.8**		**98.9**		299
FDA		23.7		**93.8**		1		58.6		96.5		1
LPP		82.1		92.9		30		91.4		97.5		30
Model 1		**91.4**		93.3		5		96.4		98.0		3
Model 2		**93.9**		92.7		12		**97.9**		**98.4**		4

more than two objects. Different from the positive samples that cluster together, the negative samples form no less than two subclusters. In other words, the distributions of the positive and negative classes are imbalanced. This degrades the performance of many traditional feature extraction methods and classifiers. However, as the proposed method only requires the positive samples cluster together, it can perform well and achieve high accuracy. Thus, the proposed method is less likely affected by the imbalanced distributions of the positive and negative samples.

In the training stage of many traditional feature extraction methods and classifiers, the positive and negative samples are input concurrently. In the proposed two models, however, the positive and negative samples are input independently in two steps. This alleviates the effect of the imbalanced sizes of the positive and negative classes.

In the classification stage, the proposed method assigns two asymmetrical areas to the imbalanced positive and negative classes. This restricts the positive features in a relatively small area. Thus, the outliers are less likely misclassified into the positive class.

Acknowledgment

The authors are most grateful for the constructive advice on the revision of the manuscript from the anonymous reviewers. The funding support from Hong Kong Government under its GRF scheme (5341/08E and 5366/09E) and the research grant from Hong Kong Polytechnic University (1-ZV5U) are greatly appreciated.

References

[1] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.
[2] T.M. Khoshgoftaar, J. Van Hulse, A. Napolitano, Supervised neural network modeling: an empirical investigation into learning from imbalanced data with labeling errors, IEEE Transactions on Neural Networks 21 (5) (2010) 813–830.
[3] M. Kirby, L. Sirovich, Application of the Karhunen–Loeve procedure for the characterization of human faces, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1) (1990) 103–108.
[4] Y. Xu, J.-Y. Yang, Z. Jin, A novel method for Fisher discriminant analysis, Pattern Recognition 37 (2) (2004) 2004.
[5] J. Yang, J.-Y. Yang, D. Zhang, What's wrong with Fisher criterion? Pattern Recognition 35 (11) (2002) 2665–2668.
[6] Y.L. Murphey, H. Guo, L. Feldkamp, Neural learning from imbalanced data, Applied Intelligence, special issue on Neural Networks and Applications 21 (2004) 117–128.
[7] W. Chen, C.E. Metz, M.L. Giger, K. Drukker, A novel hybrid linear/nonlinear classifier for two-class classification: theory algorithm and applications, IEEE Transaction on Medical Imaging 29 (2) (2010) 428–441.
[8] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, New York, 2001.
[9] B. Scherrer, Gaussian Mixture Model Classifiers, 2007. Available online at 〈http://www.music.mcgill.ca~scherrer/MUMT611/a03/Scherrer07GMM.pdf〉.
[10] D.A. Reynolds, Speaker identification and verification using Gaussian mixture speaker models, Speech Communication 17 (1–2) (1995) 91–108.
[11] P. Bansal, K. Kant, S. Kumar, A. Sharda, S. Gupta, Improved hybrid model of HMM/GMM for speech recognition, Intelligent Information and Engineering Systems (2008) 69–74.
[12] D.A. Reynolds, R.C. Rose, Robust text-independent speaker identification using Gaussian mixture speaker models, IEEE Transactions on Speech and Audio Processing 3 (1) (1995) 72–83.
[13] J.V.B. Soares, R.M. Cesar-Jr., Segmentation of retinal vasculature using wavelets and supervised classification: theory and implementation, in: H.F. Jelinek, M.J. Cree (Eds.), Automated Image Detection of Retinal Pathology, CRC Press, 2007.
[14] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton Univ. Press, NJ, 1961.
[15] T. Zhang, K. Huang, X. Li, J. Yang, D. Tao, Discriminative orthogonal neighborhood-preserving projections for classification, IEEE Transaction on Systems, Man, and Cybernetics-part B: Cybernetics 40 (1) (2010) 253–263.
[16] D. Tao, X. Li, X. Wu, S.J. Maybank, Geometric mean for subspace selection, IEEE Transaction on Pattern Analysis and Machine Intelligence 31 (2) (2009) 260–274.
[17] J. Sun, D. Tao, S. Papadimitriou, P. Yu, C. Faloutsos, incremental tensor analysis: theory and applications, ACM Transactions on Knowledge Discovery from Data 2 (3) (2008) 1–37.
[18] T. Zhang, X. Li, D. Tao, J. Yang, A unifying framework for spectral analysis based dimensionality reduction, International Joint Conference on Neural Network (2008) 1671.
[19] X. Jiang, Asymmetric principal component and discriminant analyses for pattern classification, IEEE Transaction on Pattern Analysis and Machine Intelligence 31 (5) (2009) 931–937.
[20] M. Turk, A. Pentland, Eigenfaces for recognition, Cognitive Neuroscience 3 (1) (1991) 72–86.

[21] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, Face recognition by independent component analysis, IEEE Transaction on Neural Networks 13 (6) (2002) 1450–1464.
[22] J. Yang, A.F. Frangi, J. Yang, D. Zhang, Z. Jin, KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition, IEEE Transaction on Pattern Analysis and Machine Intelligence 27 (2) (2005) 230–244.
[23] D.D. Lee, H.S. Seung, Learning the parts of objects by nonnegative matrix factorization, Nature 401 (1999) 788–791.
[24] P. Penev, J. Atick, Local feature analysis: a general statistical theory for object representation, Network: Computation in Neural Systems 7 (3) (1996) 477–500.
[25] G. Polzlbauer, T. Lidy, A. Rauber, Decision manifolds—a supervised learning algorithm on self-organization, IEEE Transaction on Neural Networks 19 (9) (2008) 1518–1530.
[26] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 643–660.
[27] L.M. Manevitz, M. Yousef, One-class SVMs for document classification, Journal of Machine Learning Research 2 (2001) 139–154.
[28] S.A. Nene, S.K. Nayar, H. Murase, Columbia Object Image Library (COIL-20), Technical Report CUCS-005-96, 1996.
[29] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression database, IEEE Transaction on Pattern Analysis and Machine Intelligence 25 (12) (2003) 1615–1618.
[30] X.F. He, S. Yan, Y.X. Hu, N.P.H.-J. Zhang, Face recognition using Laplacianfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.

Jinghua Wang received his B.S. degree in Computer Science from the Shandong University and his M.S. degree from the Harbin Institute of Technology. He is currently a Ph.D. candidate in the Department of Computing, The Hong Kong Polytechnic University. His current research interests are in the areas of pattern recognition and image processing.

Jane You obtained her B.Eng. in Electronic Engineering from the Xi'an Jiaotong University in 1986 and Ph.D. in Computer Science from the La Trobe University, Australia, in 1992. She was a lecturer at the University of South Australia and senior lecturer at the Griffith University from 1993 till 2002. Currently she is a professor at the Hong Kong Polytechnic University. Her research interests include image processing, pattern recognition, medical imaging, biometrics computing, multimedia systems, and data mining.

Qin Li received his B.Eng. degree in computer science from the China University of Geoscience, the M.Sc. degree (with distinction) in computing from the University of North-Umbria at Newcastle, and the Ph.D. degree from the Hong Kong Polytechnic University. His current research interests include medical image analysis, biometrics, image processing, and pattern recognition.

Yong Xu was born in Sichuan, China, in 1972. He received his B.S. degree, M.S. degree in 1994 and 1997 respectively. He received the Ph.D. degree in Pattern recognition and Intelligence System at NUST(China) in 2005. Now he works at the Shenzhen graduate school, Harbin Institute of Technology. His current interests include feature extraction, biometric, face recognition, machine learning, and image processing.