Cost-Sensitive Face Recognition

Yin Zhang and Zhi-Hua Zhou, *Senior Member, IEEE*

Abstract—Most traditional face recognition systems attempt to achieve a low recognition error rate, implicitly assuming that the losses of all misclassifications are the same. In this paper, we argue that this is far from a reasonable setting because in almost all application scenarios of face recognition, different kinds of mistakes will lead to different losses. For example, it will be troublesome if a door-locker based on a face recognition system misclassified a family member as a stranger such that s/he were not allowed to enter the house; but it will be a much more serious disaster if a stranger were misclassified as a family member and allowed to enter the house. We propose a framework which formulates face recognition problem as a multi-class cost-sensitive learning task, and develop two theoretically sound methods for this task. Experimental results demonstrate the effectiveness and efficiency of the proposed methods.

Index Terms—Face recognition, cost-sensitive face recognition, cost-sensitive learning, multi-class cost-sensitive learning.

1 Introduction

Face recognition has attracted much research effort for many years. Many successful face recognition systems have been developed, such as [2], [3], [19], [23], and Zhao et al. [25] provided a good survey. The operation of face recognition systems can be divided into two modes, i.e., *verification* mode and *identification* mode. In the verification mode, the system validates whether the individual is the identity s/he claims to be; in the identification mode, the system recognizes the individual by searching the database to find who the individual is or s/he does not belong to the database [11]. In this paper, we concern about identification. To the best of our knowledge, most of those face recognition systems attempt to achieve a low recognition error rate.

Pursuing a minimum error rate always implies that the system assumes that any misclassification will cause the same amount of loss since it simply tries to minimize the number of mistakes. Although this assumption is widely taken, we argue that it is not really reasonable because for most real-world applications, different kinds of mistakes generally lead to different amount of losses. For example, consider a door-locker based on a face recognition system for a certain group (e.g., *family members* or *employees of a company*), the possible mistakes in predicting a probe face image include:

1) False acceptance, i.e., mis-recognizing an impostor as a gallery subject;
2) False rejection, i.e., mis-recognizing a gallery subject as an impostor;
3) False identification, i.e., mis-recognizing between two gallery subjects.

In traditional face recognition systems, these errors are treated equally. It is evident, however, that these errors will cause different amount of losses. When the second error occurs, a gallery subject is mistakenly rejected, which is troublesome; but if compared with the first error, the second one is less serious since it would be a disaster if an impostor is mistakenly allowed to enter the house. The third error also causes some trouble since members in the house might have different private rooms or the company wants to record the staff attendant, yet such an error is obviously much less serious than the first and the second ones.

There are many other applications where the severity of different kinds of errors varies. For example, a salesman at a shop may want to increase her/is chances of identifying old customers even at the cost of misclassifying some new customers as old ones. Thus, the false acceptance (mis-recognizing a new customer as an old one) may not be as serious as the false rejection (missing an old customer) or even the false identification (mis-recognizing two old customers).

It is clear that these three kinds of errors are quite different and simply taking error rate as the measure of the performance may not be a good choice. In the following, without loss of generality, we assume that the false rejection is more serious than false identification, and the false acceptance is the most serious error. Other situations will be considered in our experiments later.

In the machine learning and data mining communities, a kind of classification problems called *cost-sensitive learning* has been studied for years [1], [5], [7], [13], [26]. In such settings, 'cost' information is introduced to measure the severity of misclassification and different costs reflect different amount of losses. The purpose of cost-sensitive learning is to minimize *total cost* rather than *total error*. There are two kinds of cost-sensitive problems, i.e., problems with *class-dependent cost* [5], [7], [13], [26] or *example-dependent cost* [1]. In the former kind of problems, the cost is determined by error type; that is, misclassifying any example of the i-th class into

- *Manuscript received December 6, 2008.*
This research was supported by the National Science Foundation of China (60635030, 60721002), the Jiangsu Science Foundation (BK2008018), and the National Fundamental Research Program of China (2010CB327900).
The authors are with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China (E-mail: {zhangyin, zhouzh}@lamda.nju.edu.cn).

the j-th class will always have the same cost, while misclassifying an example into different classes may lead to different costs. In the latter kind of problems, the cost is determined by the example, while different examples will have different costs even when the error types are the same.

Inspired by cost-sensitive learning, in this paper, we formulate the face recognition problem as a multi-class cost-sensitive learning task. For convenience of discussion, we consider a situation that, accepting any impostor will result in the same loss, and misclassifying a gallery subject as another gallery subject or an impostor will result in different amount of losses. So, our problem is a class-dependent cost-sensitive problem. In contrast to conventional face recognition systems which try to minimize the total error rate, we try to minimize the total cost, aiming to prevent disasters caused by mistakes with high costs. Note that in our concerned setting, the costs of different kinds of misclassifications are given by the user according to their requirement. For example, if the user believes that letting an impostor in will cause a disaster ten times more severe than blocking a legal person, s/he would set the cost for the former as ten times as that of the latter.

Some previous biometric studies considered the difference between false acceptance and false rejection, e.g. [11], [14], and used the receiver operating characteristic (ROC) curve to select a proper threshold for classification. These methods can be viewed as implicitly using cost information since the threshold depends on the cost. To the best of our knowledge, our work is the first attempt on explicitly formulating the face recognition problem as a cost-sensitive learning problem and trying to minimize the total cost directly. This formulation is more natural and solid. Moreover, those ROC-based methods focused on binary classification problems, while face recognition is inherently a multi-class problem while extending ROC methods to multi-class is non-trivial [12].

In this paper, we propose two new cost-sensitive methods, *mcKLR* and *mckNN*, for face identification problems. mcKLR is an inductive learning method derived from Bayes decision theory, while mc$k$NN is a cost-sensitive version of k-nearest neighbor classifier. Both methods handle multi-class cost-sensitive classification problems. mc$k$NN is particularly suitable for situations where group members vary frequently, while mcKLR is more suitable for situations with stable group members. Experimental results on AR and FERET databases validate the effectiveness and efficiency of our methods.

The rest of this paper is organized as follows. In Section 2 we formulate the cost-sensitive face recognition problem. In Section 3 we briefly introduce some existing multi-class cost-sensitive learning methods. Then we propose the mcKLR and mc$k$NN methods in Section 4 and report on our experiments in Section 5. In Section 6 we discuss how to construct the cost matrix effectively from the interaction with users. Finally, we conclude the paper in Section 7.

2 Problem Formulation

Denote a face image by $\boldsymbol{x}$ and y for its label. Considering that there are M gallery subjects, denoted by $y = G_1, \cdots, G_M$ and many impostors, denoted by a meta-class $y = I$. Traditional face recognition systems try to generate a hypothesis $\phi(\boldsymbol{x})$ minimizing the expected **error rate**: Err $= E_{\boldsymbol{x},y}\big(\mathbb{I}(\phi(\boldsymbol{x}) \neq y)\big)$, where $\mathbb{I}$ is indicator function which takes 1 when $\phi(\boldsymbol{x}) \neq y$ and 0 otherwise. Thus, these systems implicitly assume that the costs of all kinds of mistakes are the same. As mentioned before, however, such assumption is often not reasonable and different mistakes are generally associated with different costs. As analyzed above, our problem is class-dependent cost-sensitive and we can categorize the costs into three types:

1) Cost of false acceptance, C_{IG};
2) Cost of false rejection, C_{GI};
3) Cost of false identification, C_{GG}.

According to our discussion above about the severity of differen types of errors, it is evident that C_{IG}, C_{GI} and C_{GG} are unequal. Given a cost setting according to the user's intention, according to [7], we can reassign $C_{IG} = C_{IG}/C_{GG}$, $C_{GI} = C_{GI}/C_{GG}$ and $C_{GG} = 1$ with optimal solution unchanged. Here, for the ease of understanding, we still preserve the original formulation. We can construct a cost matrix $\mathbf{C}$ as shown in Table 1, where C_{ij} indicates the cost of misclassifying a face image of the i-th person as the j-th person. The diagonal elements of $\mathbf{C}$ are all zero since there is no loss for correct recognition. Usually it is easy for users to specify which kind of mistake is with a higher cost and which is with a lower cost. Thus, we assume that the cost matrix is given by users and we will focus on how to make the face recognition system behave well given a cost matrix. At the end of the paper we will try to determine the cost matrix by interacting with users.

It is clear that our concerned problem is an $(M+1)$-class cost-sensitive learning problem and the hypothesis $\phi(\boldsymbol{x})$ should minimize the expected **cost**: Cost $= E_{\boldsymbol{x},y}(C_{y\phi(\boldsymbol{x})})$. Since $E_{\boldsymbol{x},y}(C_{y\phi(\boldsymbol{x})}) = E_{\boldsymbol{x}}\left(E_{y|\boldsymbol{x}}(C_{y\phi(\boldsymbol{x})}|\boldsymbol{x})\right)$, minimizing $E_{\boldsymbol{x},y}(C_{y\phi(\boldsymbol{x})})$ is equivalent to minimizing $E_{y|\boldsymbol{x}}(C_{y\phi(\boldsymbol{x})}|\boldsymbol{x})$ on every $\boldsymbol{x}$. Hence we can define the expected loss of predicting $\boldsymbol{x}$ by $\phi(\boldsymbol{x})$ as: $\text{loss}\big(\boldsymbol{x}, \phi(\boldsymbol{x})\big) = E_{y|\boldsymbol{x}}(C_{y\phi(\boldsymbol{x})}|\boldsymbol{x})$. For our problem, we have

$$\text{loss}\big(\boldsymbol{x}, \phi(\boldsymbol{x})\big) =$$
$$\begin{cases} \sum\limits_{\substack{m=1 \\ m\neq\tau}}^{M} \mathbf{P}(G_m|\boldsymbol{x})C_{GG} + \mathbf{P}(I|\boldsymbol{x})C_{IG} & \text{if } \phi(\boldsymbol{x}) = G_\tau \\ \sum\limits_{m=1}^{M} \mathbf{P}(G_m|\boldsymbol{x})C_{GI} & \text{if } \phi(\boldsymbol{x}) = I \end{cases} \tag{1}$$

where we denote $\mathbf{P}(y = G_m|\boldsymbol{x})$ and $\mathbf{P}(y = I|\boldsymbol{x})$ as $\mathbf{P}(G_m|\boldsymbol{x})$ and $\mathbf{P}(I|\boldsymbol{x})$ for simplicity. Therefore, in order

TABLE 1
The cost matrix

	G_1	$\cdots$	G_M	I
G_1	0	$\cdots$	C_{GG}	C_{GI}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
G_M	C_{GG}	$\cdots$	0	C_{GI}
I	C_{IG}	$\cdots$	C_{IG}	0

to minimize the total cost, the optimal prediction of $\boldsymbol{x}$ should be

$$\phi^*(\boldsymbol{x}) = \underset{\phi(\boldsymbol{x})\in\{G_1,\cdots,G_M,I\}}{\arg\min} \operatorname{loss}\big(\boldsymbol{x},\phi(\boldsymbol{x})\big) \qquad (2)$$

Multi-class cost-sensitive learning algorithms try to solve this minimization problem. In the next section we will briefly introduce some existing multi-class cost-sensitive methods.

3 Multi-Class Cost-Sensitive Learning

3.1 Rescaling

Rescaling [7], [26] is a general approach that can be used to make cost-blind learning algorithms cost-sensitive. The principle is to enable the influences of the higher-cost classes to be larger than that of the lower-cost classes. The rescaling approach can be realized in many ways, such as assigning training examples of different classes with different weights [7], [22], sampling the classes according to their costs [6], [7], [17], or moving the decision threshold [5], [7]. The rescaling approach is effective in dealing with binary-class problems.

Zhou and Liu [26] analyzed the reason why the traditional rescaling approach is often not effective on multi-class problems and revealed that it is helpful only when all the classes can be consistently rescaled simultaneously. Based on the analysis, a new approach was proposed, which should be the choice if the user wants to use rescaling for multi-class cost-sensitive learning. For an (M+1)-class problem, if each class can be assigned with an optimal weight w_m ($1 \le m \le M+1$, $w_m > 0$), it is desired that the weights satisfy $w_i/w_j = C_{ij}/C_{ji}$ for every two classes i and j. It implies the following $\binom{M+1}{2}$ number of constraints must be satisfied, which is called as the *consistency* condition [26]:

$$\begin{array}{llll}
\frac{w_1}{w_2}=\frac{C_{12}}{C_{21}}, & \frac{w_1}{w_3}=\frac{C_{13}}{C_{31}}, & \cdots, & \frac{w_1}{w_{(M+1)}}=\frac{C_{1(M+1)}}{C_{(M+1)1}} \\
 & \frac{w_2}{w_3}=\frac{C_{23}}{C_{32}}, & \cdots, & \frac{w_2}{w_{(M+1)}}=\frac{C_{2(M+1)}}{C_{(M+1)2}} \\
 & \cdots & \cdots & \cdots \\
 & & & \frac{w_M}{w_{(M+1)}}=\frac{C_{M(M+1)}}{C_{(M+1)M}}
\end{array}$$

In order to perform rescaling simultaneously, $\boldsymbol{w} = (w_1, w_2, \cdots, w_{M+1})^T$ must be the non-trivial solution of a linear equation system with the coefficient matrix:

$$\begin{bmatrix}
C_{21} & -C_{12} & 0 & \cdots & 0 \\
C_{31} & 0 & -C_{13} & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots & 0 \\
C_{(M+1)1} & 0 & 0 & \cdots & -C_{1(M+1)} \\
0 & C_{32} & -C_{23} & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots & 0 \\
0 & C_{(M+1)2} & 0 & \cdots & -C_{2(M+1)} \\
0 & 0 & 0 & \cdots & -C_{M(M+1)}
\end{bmatrix} \qquad (3)$$

It is equivalent to requiring the coefficient matrix (3) to have a rank smaller than $(M+1)$. Otherwise, rescaling may not be effective on multi-class problems. In our cost-sensitive face recognition task, the coefficient matrix's rank is M. Therefore, theoretically we can use rescaling to solve this problem.

MetaCost [5] is popularly used to make cost-blind learning algorithms cost-sensitive. Actually this is also a rescaling method which relabels training examples to minimize Bayesian risk by threshold moving; in other words, this method works by rescaling data based on the posterior probability and the cost setting. As MetaCost directly modifies the labels rather than assigning weights to examples, the consistency problem mentioned above does not exist; however, from empirical study, Ting [21] indicated that the internal cost-sensitive classifier employed by MetaCost to relabel the training examples may outperform the final model without additional computation, and therefore he did not recommend MetaCost.

Our experimental results reveal that the rescaling methods do not work well on our cost-sensitive face recognition task and the reason will be discussed in Section 5.

3.2 Multi-Class Cost-Sensitive SVM (mcSVM)

Support vector machine (SVM) has been successfully applied to face recognition [10], [15]. Since SVM was originally designed for binary classification and our cost-sensitive face recognition task is a multi-class problem, multi-class extensions are needed. The *one-vs-one* and *one-vs-all* are two popular strategies to handle the gap between multi-class problems and binary classifiers by decomposing a multi-class problem into a series of binary-class problems. These approaches, however, may fail under various circumstances [13], [15]. Lee et al. [13] derived a multi-class cost-sensitive SVM, i.e., mcSVM. In this method, for an (M+1)-class classification problem, the example $\boldsymbol{x}$'s label y is extended to an (M+1)-dimensional label vector, denoted by $\boldsymbol{y}$, where $\boldsymbol{y}$ takes 1 for the y-th element and $-1/M$ for the others. For instance, if the i-th example falls into the 1st class, then $\boldsymbol{y}_i = (1, -1/M, \cdots, -1/M)^T$. Accordingly, an ($M$+1)-tuple of separating functions $\boldsymbol{f}(\boldsymbol{x}) = \big(f_1(\boldsymbol{x}), \cdots, f_{M+1}(\boldsymbol{x})\big)^T$ is defined, where $f_m(\boldsymbol{x}) = h_m(\boldsymbol{x}) + b_m$, $h_m \in \mathcal{H}_K$ and $b_m \in \mathbb{R}$. $\mathcal{H}_K$ is a reproducing kernel Hilbert space (RKHS) with the reproducing kernel function $K(\cdot,\cdot)$, where $\boldsymbol{f}(\boldsymbol{x})$ is with the sum-to-zero constraint $\sum_{m=1}^{M+1} f_m(\boldsymbol{x}) = 0$ for any $\boldsymbol{x}$.

Define the loss function for mcSVM as $\mathbf{L}\big(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}\big) = \mathbf{C}_{y\cdot}\big(\boldsymbol{f}(\boldsymbol{x})-\boldsymbol{y}\big)_+$, where $\mathbf{C}_{y\cdot}$ is the y-th row of the cost matrix $\mathbf{C}$ and $\big(\boldsymbol{f}(\boldsymbol{x})-\boldsymbol{y}\big)_+$ is the generalized hinge loss function, $\big((f_1(\boldsymbol{x}) - \boldsymbol{y}_1)_+, \cdots, (f_{M+1}(\boldsymbol{x}) - \boldsymbol{y}_{(M+1)})_+\big)^T$. Lee et al. [13] proved that the minimizer of expected risk $E_{\boldsymbol{x},\boldsymbol{y}}\big(\mathbf{L}(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y})\big)$ under the sum-to-zero constraint is $\boldsymbol{f}^*(\boldsymbol{x}) = \big(f_1^*(\boldsymbol{x}), \cdots, f_{M+1}^*(\boldsymbol{x})\big)^T$ with

$$f_\tau^*(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \tau = \underset{m=1,\cdots,M+1}{\arg\min} \ \operatorname{loss}(\boldsymbol{x}, m) \\ -1/M & \text{otherwise.} \end{cases} \tag{4}$$

Here $\operatorname{loss}(\boldsymbol{x}, m) = \sum_{m'=1}^{M+1} \mathbf{P}(m'|\boldsymbol{x})C_{m'm}$ as defined in Section 2. It means that the best predicted label of the new example $\boldsymbol{x}$ under Bayes decision rule is the subscript of the maximum of separating functions, i.e.,

$$\phi(\boldsymbol{x}) = \arg\max_m f_m(\boldsymbol{x}) = \underset{m=1,\cdots,M+1}{\arg\min} \ \operatorname{loss}(\boldsymbol{x}, m) \tag{5}$$

For finite case $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, the expected risk is replaced by the empirical risk. Considering structural risk, the optimization objective can be written as:

$$\frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|} \mathbf{C}_{y_i\cdot}\big(\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{y}_i\big)_+ + \frac{1}{2}\lambda \sum_{m=1}^{M+1} \|f_m\|_{H_K}^2 \tag{6}$$

Actually, when $M = 1$, the generalized hinge loss function reduces to the binary hinge loss and if all the misclassification cost is 1, mcSVM reduces to the traditional binary cost-blind SVM.

4 Our Methods

4.1 Multi-Class Cost-Sensitive KLR (mcKLR)

4.1.1 Derivation

We can define a meta-class G as $y = G \iff \exists m \ y = G_m$ and $\mathbf{P}(G|\boldsymbol{x}) = \mathbf{P}(\bigcup_{m=1}^M G_m|\boldsymbol{x}) = \sum_{m=1}^M \mathbf{P}(G_m|\boldsymbol{x})$. Then from Eq. 1 we have

$$\begin{aligned} \operatorname{loss}(\boldsymbol{x}, G_\tau) &= \sum_{\substack{m=1\\m\neq\tau}}^{M} \mathbf{P}(G_m|\boldsymbol{x})C_{GG} + \mathbf{P}(I|\boldsymbol{x})C_{IG} \\ &= \left(\mathbf{P}(G|\boldsymbol{x}) - \mathbf{P}(G_\tau|\boldsymbol{x})\right)C_{GG} + \mathbf{P}(I|\boldsymbol{x})C_{IG} \\ &= \mathbf{P}(G|\boldsymbol{x})C_{GG} + \mathbf{P}(I|\boldsymbol{x})C_{IG} - \mathbf{P}(G_\tau|\boldsymbol{x})C_{GG} \end{aligned} \tag{7}$$

As $\boldsymbol{x}$ can be labeled as either G or I, we have $\mathbf{P}(G|\boldsymbol{x}) + \mathbf{P}(I|\boldsymbol{x}) = 1$. So Eq. 7 becomes

$$\begin{aligned} &\operatorname{loss}(\boldsymbol{x}, G_\tau) \\ &\quad= \big(1 - \mathbf{P}(I|\boldsymbol{x})\big)C_{GG} + \mathbf{P}(I|\boldsymbol{x})C_{IG} - \mathbf{P}(G_\tau|\boldsymbol{x})C_{GG} \\ &\quad= C_{GG} + \mathbf{P}(I|\boldsymbol{x})(C_{IG} - C_{GG}) - \mathbf{P}(G_\tau|\boldsymbol{x})C_{GG} \end{aligned} \tag{8}$$

and

$$\operatorname{loss}(\boldsymbol{x}, I) = \sum_{m=1}^{M} \mathbf{P}(G_m|\boldsymbol{x})C_{GI} = \mathbf{P}(G|\boldsymbol{x})C_{GI} \tag{9}$$

To minimize the loss, we should choose the minimum from the $M + 1$ items below:

$$\begin{cases} C_{GG} + \mathbf{P}(I|\boldsymbol{x})(C_{IG} - C_{GG}) - \mathbf{P}(G_1|\boldsymbol{x})C_{GG} \\ \vdots \\ C_{GG} + \mathbf{P}(I|\boldsymbol{x})(C_{IG} - C_{GG}) - \mathbf{P}(G_M|\boldsymbol{x})C_{GG} \\ \mathbf{P}(G|\boldsymbol{x})C_{GI} \end{cases} \tag{10}$$

Subtract $C_{GG} + \mathbf{P}(I|\boldsymbol{x})(C_{IG} - C_{GG})$ from every item, then the last item becomes

$$\begin{aligned} &\mathbf{P}(G|\boldsymbol{x})C_{GI} - C_{GG} - \mathbf{P}(I|\boldsymbol{x})(C_{IG} - C_{GG}) \\ &\quad= \big(1 - \mathbf{P}(I|\boldsymbol{x})\big)C_{GI} - C_{GG} - \mathbf{P}(I|\boldsymbol{x})(C_{IG} - C_{GG}) \\ &\quad= -\mathbf{P}(I|\boldsymbol{x})(C_{GI} + C_{IG} - C_{GG}) + (C_{GI} - C_{GG}) \end{aligned} \tag{11}$$

So we have an equivalent problem of choosing the minimum from:

$$\begin{cases} -\mathbf{P}(G_1|\boldsymbol{x})C_{GG} \\ \vdots \\ -\mathbf{P}(G_M|\boldsymbol{x})C_{GG} \\ -\mathbf{P}(I|\boldsymbol{x})(C_{GI} + C_{IG} - C_{GG}) + (C_{GI} - C_{GG}) \end{cases} \tag{12}$$

Divide $-C_{GG}$ from every item and denote

$$\beta = \frac{C_{GI} + C_{IG} - C_{GG}}{C_{GG}} \tag{13}$$

and

$$\Delta = \frac{C_{GI} - C_{GG}}{C_{GG}} \tag{14}$$

Then the problem becomes choosing the maximum from

$$\begin{cases} \mathbf{P}(G_1|\boldsymbol{x}) \\ \vdots \\ \mathbf{P}(G_M|\boldsymbol{x}) \\ \beta\mathbf{P}(I|\boldsymbol{x}) - \Delta \end{cases} \tag{15}$$

4.1.2 Optimization

By using logistic regression [27], we define an M-tuple of separating functions $\boldsymbol{f}(\boldsymbol{x}) = \big(f_1(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})\big)^T$ and the loss function $\mathbf{L}\big(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}), y\big)$ as

$$\begin{aligned} \mathbf{L}\big(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}), y\big) = &\sum_{\tau=1}^{M}\bigg(- \ln\frac{e^{f_\tau(\boldsymbol{x})}}{1 + \sum_{m=1}^M e^{f_m(\boldsymbol{x})}}\bigg)I(y = G_\tau) \\ &+ \bigg(- \ln\frac{1}{1 + \sum_{m=1}^M e^{f_m(\boldsymbol{x})}}\bigg)I(y = I) \end{aligned} \tag{16}$$

On the $(\boldsymbol{x}, y)$ space with *pdf* $p(\boldsymbol{x}, y)$, the optimal separating function $\boldsymbol{f}^*(\boldsymbol{x})$ is the minimizer of the expectation of $\mathbf{L}$. Since $E_{\boldsymbol{x},y}(\mathbf{L}) = E_{\boldsymbol{x}}\big(E_{y|\boldsymbol{x}}(\mathbf{L}|\boldsymbol{x})\big)$, to minimize $E_{\boldsymbol{x},y}(\mathbf{L})$ we can minimize $E_{y|\boldsymbol{x}}(\mathbf{L}|\boldsymbol{x})$ on every $\boldsymbol{x}$, where

$$\begin{aligned} E_{y|\boldsymbol{x}}(\mathbf{L}|\boldsymbol{x}) = &\sum_{\tau=1}^{M}\bigg(- \ln\frac{e^{f_\tau(\boldsymbol{x})}}{1 + \sum_{m=1}^M e^{f_m(\boldsymbol{x})}}\bigg)\mathbf{P}(G_\tau|\boldsymbol{x}) \\ &+ \bigg(- \ln\frac{1}{1 + \sum_{m=1}^M e^{f_m(\boldsymbol{x})}}\bigg)\mathbf{P}(I|\boldsymbol{x}) \end{aligned} \tag{17}$$

Set the partial derivative with respect to every f_τ to zero and we get the minimizer

$$f_\tau^*(\boldsymbol{x}) = \ln \frac{\mathbf{P}(G_\tau|\boldsymbol{x})}{\mathbf{P}(I|\boldsymbol{x})}, \tag{18}$$

for $\tau = 1, \cdots, M$. Through $f_1^*, \cdots, f_M^*$ we construct a new function f_I^*:

$$\begin{aligned} f_I^*(\boldsymbol{x}) &= \ln \frac{\beta \mathbf{P}(I|\boldsymbol{x}) - \Delta}{\mathbf{P}(I|\boldsymbol{x})} \\ &= \ln \left(\beta - \Delta \left(1 + \sum_{\tau=1}^{M} e^{f_\tau^*(\boldsymbol{x})} \right) \right) \end{aligned} \tag{19}$$

Thus, choosing the maximum from $\{f_1^*(\boldsymbol{x}), \cdots, f_M^*(\boldsymbol{x}), f_I^*(\boldsymbol{x})\}$ is equivalent to choosing the maximum from Eq. 15. That is, the optimal predicted label of $\boldsymbol{x}$ under the Bayes decision rule is

$$\phi(\boldsymbol{x}) = \begin{cases} G_\tau & \text{if } f_\tau^* \text{ is the maximum} \\ I & \text{if } f_I^* \text{ is the maximum} \end{cases} \tag{20}$$

For finite case $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, the expectation of loss is replaced by empirical risk

$$\begin{aligned} \mathbf{L}(\mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} \Bigg(& \sum_{\tau=1}^{M} \left(-\ln \frac{e^{f_\tau(\boldsymbol{x}_i)}}{1 + \sum_{m=1}^{M} e^{f_m(\boldsymbol{x}_i)}} \right) I(y_i = G_\tau) \\ & + \left(-\ln \frac{1}{1 + \sum_{m=1}^{M} e^{f_m(\boldsymbol{x}_i)}} \right) I(y_i = I) \Bigg). \end{aligned} \tag{21}$$

As did in mcSVM, assume $f_m(\boldsymbol{x}) = h_m(\boldsymbol{x}) + b_m$, $h_m \in \mathcal{H}_K$ and $b_m \in \mathbb{R}$. The optimization objective can be expressed as

$$\mathbf{L}(\mathcal{D}) + \frac{1}{2}\lambda \sum_{m=1}^{M} \|f_m\|_{H_K}^2 \tag{22}$$

Note that the optimization problem is similar to the optimization form of multi-class kernel logistic regression (KLR) [8], [27]. Therefore, we can use the similar optimization technique of KLR to handle our problem and we call our method mcKLR.

We can see that all the information about the misclassification costs, i.e., C_{GG}, C_{GI} and C_{IG}, is embedded into β and Δ. In the training process, we only need to find the optimal $f_\tau^*(\boldsymbol{x})$ $(1 \le \tau \le M)$, since $f_I^*(\boldsymbol{x})$ is dependent on them. Therefore, no cost information is needed, which means the training process is independent from the cost setting. Cost-sensitivity is achieved by $f_I^*(\boldsymbol{x})$ alone, which is constructed from $f_\tau^*(\boldsymbol{x})$, β and Δ in the test process, as shown in Eq. 19. Therefore, if the cost setting (i.e., $C_{IG} : C_{GI} : C_{GG}$) changes, we only need to adjust the prediction process rather than retraining the whole system. This is very efficient since the predictions of mcKLR can be updated according to the changing cost matrix on-line. Moreover, mcKLR can output all possible predictions corresponding to different β and Δ settings almost all at once with negligible time consumption. We will discuss this further in Section 6.

4.2 Multi-Class Cost-Sensitive k-Nearest Neighbor (mc$k$NN)

k-nearest neighbor (kNN) is possibly one of the simplest machine learning algorithms. For any new instance, its label is decided by majority voting among the labels of its nearest k neighbors in the training set. There is no explicit training process and all computation is deferred to the prediction step. Therefore, kNN is a lazy learning method. Different distance types can be used to determine the k nearest neighbors of the unseen instance, such as Euclidian distance and Mahalanobis distance.

Similar to mcKLR, here we propose a cost-sensitive kNN, denoted by mc$k$NN. According to the statistical information gained from those neighboring instances, Bayes decision theory is utilized to determine the label of the test instance.

First, for a test instance $\boldsymbol{x}$, its k nearest neighbors in the training set, $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_k$, are identified. Then, labels of the k nearest neighbors are regarded as new features for $\boldsymbol{x}$. Denote the new features for $\boldsymbol{x}$ as $\boldsymbol{z} = \{y_1, y_2, \cdots, y_k\}$. Thus, the posterior probability of $\boldsymbol{x}$ having the label y can be expressed as $\mathbf{P}(y|\boldsymbol{z}) = \mathbf{P}(y|y_1, y_2, \cdots, y_k)$. Assume that the k nearest neighbors of $\boldsymbol{x}$ are conditionally independent; in other words, the i-th nearest neighbor of $\boldsymbol{x}$ is independent with the previous $(i-1)$ nearest neighbors. Then, we have

$$\mathbf{P}(y_1, y_2, \cdots, y_k|y) = \mathbf{P}(y_1|y)\mathbf{P}(y_2|y) \cdots \mathbf{P}(y_k|y) \tag{23}$$

where the likelihood, $\mathbf{P}(y_i|y)$ $(i = 1, \cdots, k)$, implies the probability of that the instance with label y_i is one of the k nearest neighbors of the instance with label y. This assumption is mainly used for the computational simplicity and otherwise we will suffer from combinatorial explosion. In our experiments it can be found that the assumption on the conditional independence is reasonable under most circumstances. Based on the assumption of Eq. 23, the posterior probability $\mathbf{P}(y|\boldsymbol{z})$ can be expressed by Bayes rule as

$$\mathbf{P}(y|\boldsymbol{z}) = \frac{\mathbf{P}(y)\mathbf{P}(\boldsymbol{z}|y)}{\mathbf{P}(\boldsymbol{z})} = \frac{\mathbf{P}(y)\mathbf{P}(y_1|y) \cdots \mathbf{P}(y_k|y)}{\mathbf{P}(\boldsymbol{z})} \tag{24}$$

Here $\mathbf{P}(y_i|y)$ $(i = 1, \cdots, k)$ and $\mathbf{P}(y)$ can be estimated from the training set. Assume that there are s instances in the training set with label y, denoted as $\boldsymbol{x}_1^y, \cdots, \boldsymbol{x}_s^y$, and in the k nearest neighbors of each $\boldsymbol{x}_t^y$ $(1 \le t \le s)$ there are k_t instances with label y_i. We have

$$\mathbf{P}(y_i|y) = \frac{\sum_{t=1}^{s} k_t}{k \times s} \tag{25}$$

and the prior probability $\mathbf{P}(y)$ can be estimated from the proportion of the training samples with label y among all the training samples as

$$\mathbf{P}(y) = \frac{|\mathcal{D}_y|}{|\mathcal{D}|} \tag{26}$$

where $\mathcal{D}_y$ is the subset of samples with label y.

In contrast to traditional kNN which uses only the information of training set in prediction, we can compute

$\mathbf{P}(y_i|y_j)$ for each label pair (y_i, y_j) and $\mathbf{P}(y_i)$ for each label y_i before prediction. With this information, almost no additional computation is needed if compared with traditional kNN in the test process.

Having $\mathbf{P}(y_i|y)$ beforehand we can get the posterior probability $\mathbf{P}(y|\boldsymbol{z})$ for a new instance as in Eq. 24. In the cost-sensitive formulation, similar to Eq. 1, we have the cost of predicting $\boldsymbol{x}$ as $\phi(\boldsymbol{x})$:

$$\text{loss}\big(\boldsymbol{x}, \phi(\boldsymbol{x})\big) = \begin{cases} \sum\limits_{\substack{m=1 \\ m\neq\tau}}^{M} \mathbf{P}(G_m|\boldsymbol{z})C_{GG} + \mathbf{P}(I|\boldsymbol{z})C_{IG} & \text{if } \phi(\boldsymbol{x}) = G_\tau \\ \sum\limits_{m=1}^{M} \mathbf{P}(G_m|\boldsymbol{z})C_{GI} & \text{if } \phi(\boldsymbol{x}) = I \end{cases} \tag{27}$$

Replacing $\mathbf{P}(G_m|\boldsymbol{z})$ and $\mathbf{P}(I|\boldsymbol{z})$ with the expression in Eq. 24, we have

$$\text{loss}\big(\boldsymbol{x}, \phi(\boldsymbol{x})\big) = \frac{1}{\mathbf{P}(\boldsymbol{z})} \times \begin{cases} \sum\limits_{\substack{m=1 \\ m\neq\tau}}^{M} \mathbf{P}(G_m) \prod\limits_{i=1}^{k} \mathbf{P}(y_i|G_m)C_{GG} + \mathbf{P}(I) \prod\limits_{i=1}^{k} \mathbf{P}(y_i|I)C_{IG} & \\ & \text{if } \phi(\boldsymbol{x}) = G_\tau \\ \sum\limits_{m=1}^{M} \mathbf{P}(G_m) \prod\limits_{i=1}^{k} \mathbf{P}(y_i|G_m)C_{GI} & \text{if } \phi(\boldsymbol{x}) = O \end{cases} \tag{28}$$

For each $\boldsymbol{x}$, $1/\mathbf{P}(\boldsymbol{z})$ is constant and can be omitted. $\mathbf{P}(G_m)$, $\mathbf{P}(I)$, $\mathbf{P}(y_i|G_m)$ and $\mathbf{P}(y_i|I)$ are all estimated from the training set by Eq. 25 and 26 beforehand. Then, we can get the optimal prediction of $\boldsymbol{x}$ as

$$\phi^*(\boldsymbol{x}) = \underset{\phi(\boldsymbol{x})\in\{G_1,\cdots,G_M,I\}}{\arg\min} \text{loss}\big(\boldsymbol{x}, \phi(\boldsymbol{x})\big) \tag{29}$$

Notice that since we compute Eqs. 25 and 26 in the training process, the cost of computing Eq. 28 is negligible compared with that of finding the k nearest neighbors of $\boldsymbol{x}$. Therefore, the time complexity of mc$k$NN is almost as the same as kNN.

mcKLR and mc$k$NN are adapted for different applications. For situations where the group members vary frequently, such as enrolling new members or cancelling old members, mc$k$NN may be a good choice. To handle the change of the group members, mc$k$NN only need to add the corresponding information of the new member or delete the corresponding information of the old member. While for mcKLR, we have to retrain the model. For situations with stable group members, mcKLR should be preferred since it is often more effective and efficient in prediction than mc$k$NN. Our experiments will demonstrate those properties.

It is worth noting that our focus in this paper is on face recognition and in most real face recognition applications, C_{IG}, C_{GI} and C_{GG} can be treated approximately as the same for all gallery subjects and impostors. So, in this paper we only consider such case. However, with slight modifications in our derivation from the Bayes decision rules, both mcKLR and mc$k$NN can be extended to the general case where the costs of misclassifying any pair of subjects may be different. Note the property that the training process is independent from the cost setting still remains for the general case.

5 Experiments

5.1 Configuration

5.1.1 Face Databases

In our experiments, AR [18] and FERET [20] face databases are used.

- AR: 126 subjects, each with 26 face images from 2 sessions. The images include frontal view faces with different facial expressions, illumination conditions and occlusions. Since our main purpose is to study cost-sensitive face recognition and no specific steps are taken to handle occlusions, the images without occlusions are used. Every image is cropped by a 165×120 rectangular mask and scaled so that the distances between the two eyes are almost the same for all images. Then the images are grayed and histogram equalized.
- FERET: 1,199 subjects, with a total of 14,051 images. The images contain variations in lighting, facial expressions, pose angles, etc. We choose images of frontal view with different expressions and illumination for our experiment. The preprocessing on FERET images is similar to that on AR except that the mask is 75×65.

The training sets and test sets each includes N_G images per randomly selected M subjects that are treated as gallery subjects, and N_I images from the remaining subjects as impostor images. Such random selection is repeated for 10 times and the average results are recorded.

5.1.2 Features

Four different kinds of features are extracted from the face images, by using Principle Component Analysis (PCA) [23], Locality Preserving Projections (LPP) [9], Linear Discriminate Analysis (LDA) [3] and Local Binary Pattern (LBP) [2], respectively.

- PCA: Aims at identifying a lower-dimensional space maximizing the variance among the data.
- LPP: Aims at finding a linear transformation preserving local structure information of original data.
- LDA: Aims at identifying a lower-dimensional space minimizing the inter-class similarity while maximizing the intra-class similarity simultaneously. It is the only supervised feature extraction method among those four methods.
- LBP: Uses the concatenated LBP feature distributions extracted from face image regions as the face descriptor. It works effectively on recognizing faces with different occlusions, poses and expressions [2]. For AR, we divide every image into 5×5 blocks with 33×24 pixels. For FERET, we divide every image

into 5×5 blocks with 15×13 pixels. For AR and FERET, we use $LBP_{8,2}^{u2}$. The subscript $(8, 2)$ means sampling 8 points on neighbor region which is a circle with radius of 2 and the superscript $u2$ means using uniform patterns.

5.1.3 Classifiers

We study three cost-blind methods, including kNN, multi-class cost-blind support vector machine (mbSVM) [13] and multi-class cost-bind kernel logistic regression (mbKLR) [27], and three cost-sensitive methods, including mcSVM [13] and our methods mc$k$NN and mcKLR.

- kNN: The training set is used as the gallery set. Every time we identify 3 nearest neighbors for the probe image, i.e., $k = 3$, and then majority voting is used for the prediction.
- mbSVM: mbSVM is the cost-blind version of mcSVM which regards all kinds of costs as the same. RBF kernel is used as the kernel function. The same kernel setting is employed in mbKLR/mcSVM/mcKLR. Five-fold cross validation is used for selecting the RBF kernel's width from e^{-3} to e^{3} and regulation coefficient from e^{-10} to e^{0}.
- mbKLR: mbKLR is the cost-blind version of mcKLR which regards all kinds of costs as the same.
- mc$k$NN: In the training process, we learn the conditional probability $\mathbf{P}(y_i|y_j)$ from the training set for each label pair of (y_i, y_j) and the prior probability $\mathbf{P}(y_i)$ for each label. In the test process, we identify $k = 3$ nearest neighbors for the probe image and use Eq. 29 to predict the probe's label.
- mcSVM: The algorithm proposed in [13] is adopted.
- mcKLR: The training process of mcKLR is as the same as mbKLR, while the cost is considered in making prediction for probe images.

We also compare our methods with Rescaling and MetaCost [5] by using mbKLR as the elementary cost-blind method. The results are not presented here since Rescaling and MetaCost simply predict every image as impostor and are thus useless. We believe that the reason of their failure is caused by the imbalance between impostors and gallery subjects. Liu and Zhou [16] have studied this problem and indicated that if class imbalance and unequal costs occur simultaneously, to rescale the classes in proportion to the cost ratio is no more optimal. Determining the optimal rescaling ratio in this case, however, is still an open problem.

We also compare our methods with two methods derived from the traditional ROC curve based methods:

- VF (verification mode): The test subject needs to claim an identity first. Then the system decides whether the test subject is the one s/he claims to be. The system can adjust the threshold according to the false accept rate (FAR). This method is intrusive. To make it non-intrusive, we try two strategies to avoid the requirement for the test subject to claim an identity. The first one, denoted by VF$_{ram}$, is to randomly assign a gallery subject to the test subject. The other one, denoted by VF$_{ave}$, is to assign a certain gallery subject to the test subject. In our experiments, we try through all the gallery subjects and report the averaged results. Note that M binary classifiers are needed.
- WL ('watch list' mode): First, the system decides whether the test subject is among the gallery subjects. If yes, the system then identifies the test subject among the gallery subjects, which is a traditional multi-class classification problem. Similarly, the system can adjust the threshold according to the FAR in the first step. Note that one binary classifier and one M-class classifier are needed.

5.2 Results

5.2.1 Fixed Influential Factors

First, we fix all the influential factors as shown in Table 2.

TABLE 2
Experimental settings

Database	M	N_G	N_I	$C_{IG} : C_{GI} : C_{GG}$
AR	10	7	300	20:2:1
FERET	20	3	300	20:2:1

We compare the total cost, total error rate (err), error rate of false acceptance (err$_{IG}$) and error rate of false rejection (err$_{GI}$) of cost-blind methods, kNN, mbSVM, mbKLR and cost-sensitive methods, mc$k$NN, mcSVM, mcKLR. The results are shown in Table 3.

From Table 3 we can find that the cost-sensitive methods have much smaller total cost than their cost-blind counterparts. It is evident that the cost-sensitive methods achieve this by preventing high-cost errors (err$_{IG}$) while slightly increasing low-cost errors (err$_{GI}$). mcKLR has the lowest total cost on all databases with all features except on FERET with LDA, where mc$k$NN is the best.

The above results have demonstrated that the cost-sensitive methods will yield less costs compared to cost-blind methods. So, in the following experiments we will not include the results of cost-blind methods for convenience.

Then we compare cost-sensitive method with ROC-based methods, i.e., WL and VF. The binary and M-class classifier used are binary and M-class cost-blind KLR, respectively. The compared cost-sensitive method is mcKLR. Since $C_{IG} > C_{GI}$, we vary the acceptance threshold from 0.05 to 0.5 with 0.05 as interval. The results are shown in Fig. 1.

From the figures we can see mcKLR is better than WL and VF (both VF$_{ram}$ and VF$_{ave}$) under all thresholds. It is reasonable since our method directly optimizes the total cost, while WL and VF embed the cost information implicitly via the acceptance threshold, yet the relationship between the acceptance threshold and the performance is not clear. Note that in this paper, we argue that the face

TABLE 3
Comparison on total cost (cost), total error rate (err), high-cost error rate (err_{IG}) and low-cost error rate (err_{GI}) on the AR and FERET databases. (The better performance between any cost-sensitive method and its cost-blind counterpart is underlined, while the best performance among all methods is **bolded**.)

Database		AR						FERET					
Feature		*kNN*	*mckNN*	*mbSVM*	*mcSVM*	*mbKLR*	*mcKLR*	*kNN*	*mckNN*	*mbSVM*	*mcSVM*	*mbKLR*	*mcKLR*
PCA	cost	198.1	144.5	105.7	100.6	91.8	**60.6**	207.9	136.6	191.2	107.5	96.2	**59.8**
	err_{IG}(%)	1.57	0.40	0.23	**0.03**	0.70	0.07	2.03	0.43	1.93	0.20	1.03	**0.07**
	err_{GI}(%)	72.71	84.86	65.43	70.29	**35.14**	40.14	70.33	91.17	62.67	79.50	**28.33**	46.50
	err(%)	15.65	16.84	12.59	13.38	**7.38**	7.76	13.83	15.89	12.06	13.44	**5.64**	7.81
LPP	cost	271.2	163.0	100.1	97.8	91.0	**79.4**	164.4	115.0	112.4	110.3	106.4	**104.4**
	err_{IG}(%)	2.67	0.67	0.13	0.10	0.20	**0.07**	1.13	**0.00**	0.20	0.20	0.13	**0.00**
	err_{GI}(%)	78.43	87.14	65.43	65.29	56.00	**53.00**	**79.33**	95.50	83.67	81.83	82.00	87.00
	err(%)	17.38	17.30	12.62	12.54	10.92	**10.41**	14.50	16.03	14.11	13.83	**13.78**	14.50
LDA	cost	75.3	75.3	129.5	111.7	71.7	**64.0**	80.6	**78.8**	200.3	199.2	94.6	84.6
	err_{IG}(%)	0.17	0.17	0.97	0.90	**0.03**	**0.03**	0.23	**0.03**	2.13	1.70	0.10	0.07
	err_{GI}(%)	46.57	46.57	50.71	**40.57**	49.29	43.71	**55.50**	64.00	60.17	81.00	73.83	67.17
	err(%)	8.97	8.97	10.51	8.65	9.54	**8.51**	**9.44**	10.69	11.83	14.92	12.39	11.25
LBP	cost	290.4	153.4	116.4	113.0	101.2	**99.8**	237.2	129.8	117.9	116.3	88.0	**85.5**
	err_{IG}(%)	3.53	0.97	0.33	**0.07**	0.17	**0.07**	2.73	0.37	0.20	0.20	0.17	**0.00**
	err_{GI}(%)	**53.00**	65.71	68.29	76.57	64.00	67.43	**60.17**	89.50	88.17	86.33	64.83	71.17
	err(%)	14.03	14.14	13.41	15.03	**12.68**	13.19	12.58	15.33	14.89	14.75	**11.00**	11.89

recognition task is inherently a cost-sensitive task since different misclassifications will cause different amount of losses. This decides that the total cost should be used as the evaluation criterion. Although WL or VF can play well with their own criteria, those criteria are not right in our concerned problem and it is meaningless to us to get better values on WL or VF criteria.

5.2.2 Varying Influential Factors

Then, we study the performance of the compared cost-sensitive methods with different numbers of gallery subjects, i.e., with varying M. For AR, M varies from 5 to 30 with 5 as interval. For FERET, M varies from 10 to 80 with 10 as interval. The results are shown in Fig. 2. The results of mcSVM on FERET when $M \geq 40$ are not shown because its efficiency is very poor and we did not get results even after waiting for a whole week. On all databases and under all kinds of features except LDA, the performance of mcKLR is always the best. Using the LDA features, mc$k$NN is also a good choice.

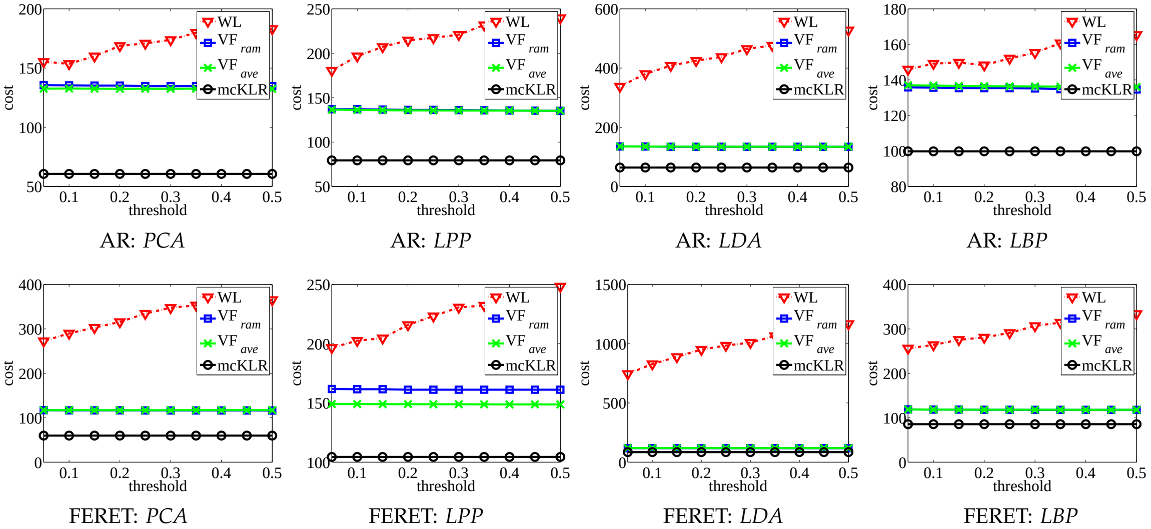
Note that the cost ratios given by the user reflect the desirable tradeoff between different kinds of errors according to her/is intention. For example, if the user thought that one false acceptance is more serious than 49 false rejections, s/he could set $C_{IG} : C_{GI} = 50 : 1$; while if s/he thought that one false acceptance is just more serious than 9 false rejections, s/he could set $C_{IG} : C_{GI} = 10 : 1$. It is interesting to compare the methods with different cost ratios to see whether they can adapt to different scenarios well. Here, we split $C_{IG} : C_{GI} : C_{GG}$ into 2 parts: C_{IG}/C_{GI} and C_{GI}/C_{GG}. C_{GG} is always set as 1. First, we fix $C_{GI}/C_{GG} = 2$ and select C_{IG}/C_{GI} from {0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20}. Note that here we consider both the conditions $C_{IG} > C_{GI}$ and $C_{IG} < C_{GI}$, which cover the two kinds of applications discussed in Section 1. Then, we fix $C_{IG}/C_{GI} = 10$ and select C_{GI}/C_{GG} from {0.1, 0.2, 0.5, 1, 2, 5, 10}. The results are shown in Fig. 3 and Fig. 4, respectively. The cost ratio axes in Figs. 3 and 4 are in log-scale for a better plot. With the cost ratio changes, mcKLR is the best choice on most cases and with the LDA feature, mc$k$NN is also a good choice. Note that in the experiments studying the varying C_{IG}/C_{GI}, although C_{IG} increases exponentially, the total costs of mc$k$NN and mcKLR do not increase exponentially. This owes to that mc$k$NN and mcKLR control the total cost via reducing the high-cost misclassification.

We also study the influence of the number of nearest neighbors, i.e., k, in kNN and mc$k$NN. The results are shown in Fig. 5. For most experiment settings, mc$k$NN is better than kNN and is also more robust to the setting of k. On experiments using LDA feature, the advantage of mc$k$NN over kNN is not obvious. A possible reason may be that LDA is a supervised feature extraction method aiming to find a subspace with the maximal discriminative ability, and such a process does not take the cost information into consideration and is optimal for the cost-blind methods. How to include the cost information into LDA is an interesting future work.

Overall, the above experiments show that mcKLR achieves the best performance on almost all databases using all kinds of features, under all numbers of gallery subjects and all cost ratios. It is clear that from the view of recognition result, mcKLR is the best choice among the compared methods.

It is interesting to notice that Lee et al. [13] has proved that when the optimal solution is obtained, the objective function of mcSVM is equivalent to Bayes decision rule with unequal costs. In our experiments, however, we find that mcSVM rarely performs better than mcKLR. The reason may be that the objective function of mcKLR was directly derived from Bayes decision rule with unequal costs. On the contrary, the inferior of mc$k$NN compared with mcKLR may be caused by the inferior of kNN, which can be observed in the experiments.

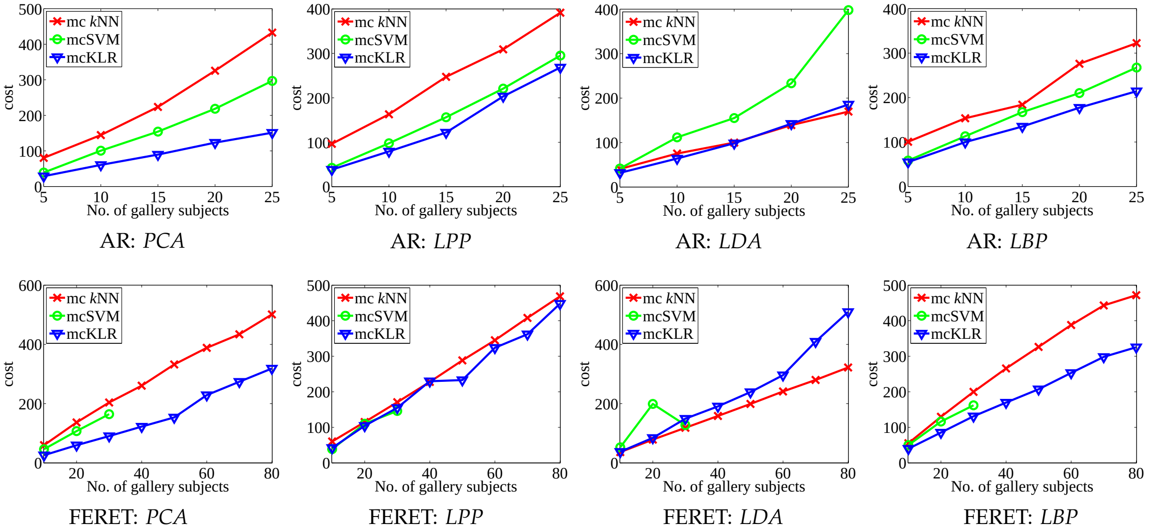
We also compare the computational costs of kNN, mc$k$NN, mcSVM and mcKLR. We record the average training and test time costs of different classifiers in



AR: *PCA* AR: *LPP* AR: *LDA* AR: *LBP*

FERET: *PCA* FERET: *LPP* FERET: *LDA* FERET: *LBP*

Fig. 1. Comparison of mcKLR and WL/VF with varying threshold. The "database: *feature*" under each figure indicates the database and feature used in the experiments.



AR: *PCA* AR: *LPP* AR: *LDA* AR: *LBP*

FERET: *PCA* FERET: *LPP* FERET: *LDA* FERET: *LBP*

Fig. 2. Comparison with different number of gallery subjects. The "database: *feature*" under each figure indicates the database and feature used in the experiments.

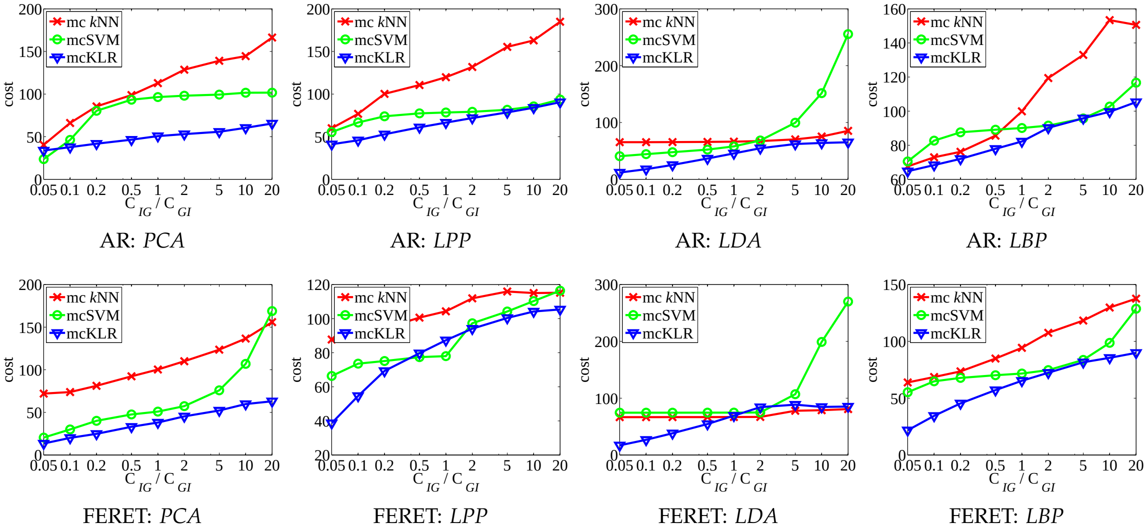
Table 4. PCA feature is used here. The experiments are conducted on a PC with CPU 2.66GHz(×64) and 4GB memory. We can see that the test time cost of mcKLR is almost as the same as that of mcSVM, but its training time cost is much smaller than that of mcSVM, especially on the larger database, FERET. There is no training process for kNN and the training process of mc$k$NN is much more efficient compared with mcSVM and mcKLR. The test time of mc$k$NN and kNN, just as the analysis in Section 4.2, is similar.

TABLE 4
Comparison of the training/test time costs (in seconds) of different classifiers with PCA features.

		kNN	mc$k$NN	mcSVM	mcKLR
AR	train	–	0.220	167.297	30.587
	test	0.882	0.895	0.036	0.048
FERET	train	–	0.049	1194.880	46.693
	test	0.251	0.263	0.025	0.028

6 Learning the Cost Matrix

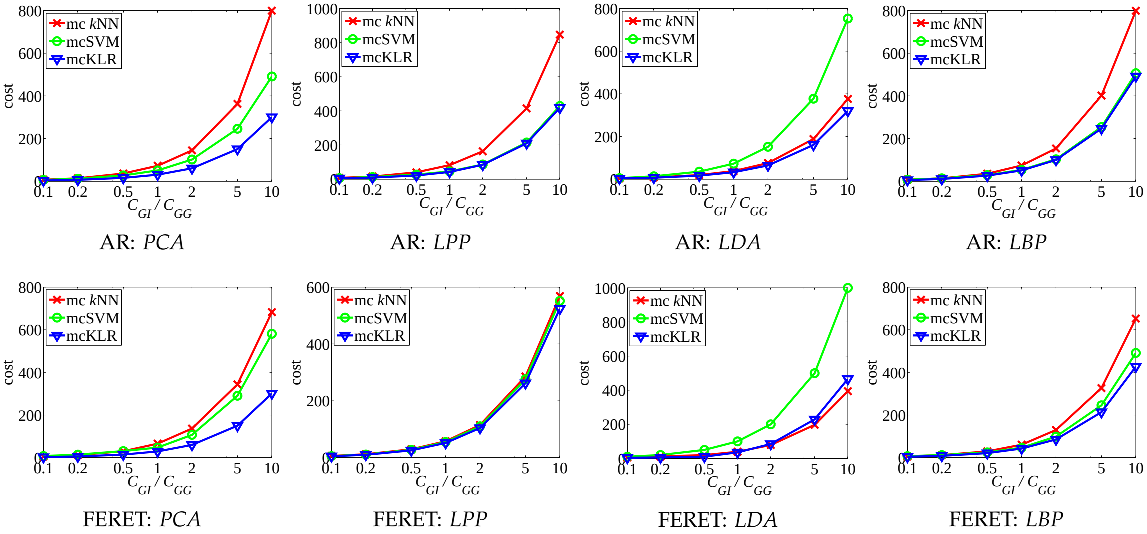
The misclassification costs in face recognition reflect the belief/demand of the user on how severe one type of mistake against another type of mistake. Even for the same data, different users may have different belief/demand. Therefore, in above discussion we assume that the cost matrix is given by the user. For some cases where the cost can be measured by money or time consuming, it is easy for the user to provide clear cost ratios for the system specifically. However, in many cases this is not so easy. The first reason is that although the user may know that false acceptance is much more serious than other types of misclassification, s/he probably



AR: *PCA* AR: *LPP* AR: *LDA* AR: *LBP*

FERET: *PCA* FERET: *LPP* FERET: *LDA* FERET: *LBP*

Fig. 3. Comparison with different settings of C_{IG}/C_{GI}. The “database: *feature*” under each figure indicates the database and feature used in the experiments.



AR: *PCA* AR: *LPP* AR: *LDA* AR: *LBP*

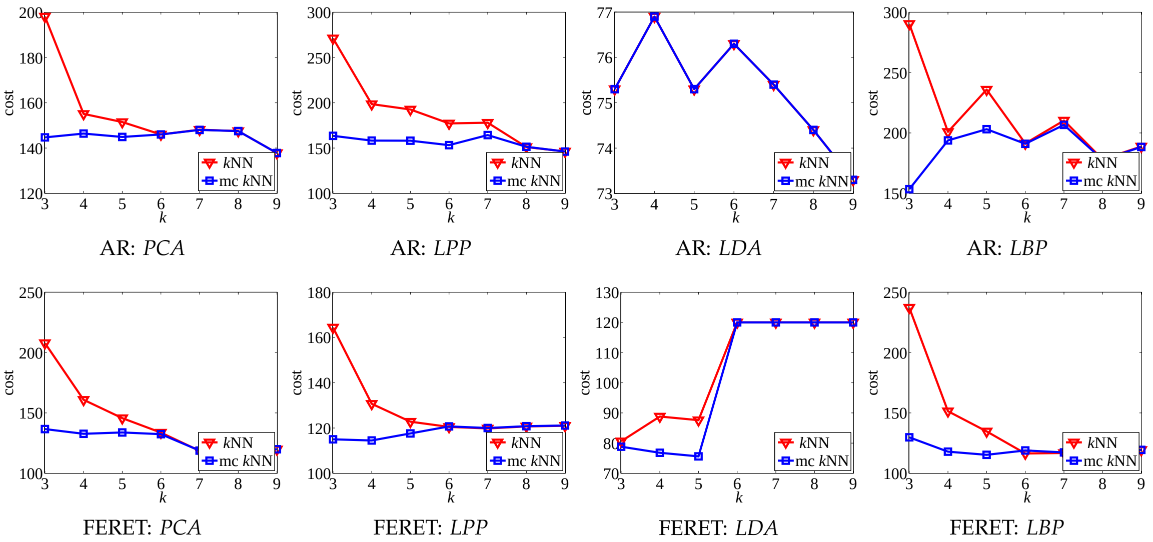
FERET: *PCA* FERET: *LPP* FERET: *LDA* FERET: *LBP*

Fig. 4. Comparison with different settings of C_{GI}/C_{GG}. The “database: *feature*” under each figure indicates the database and feature used in the experiments.

could not provide a clear measurement on how serious err$_{IG}$ is, compared with err$_{GI}$ and err$_{GG}$. For example, the difference between $C_{IG} : C_{GI} : C_{GG} = 20 : 2 : 1$ and $C_{IG} : C_{GI} : C_{GG} = 200 : 2 : 1$ may be not very different in the mind of the user although the learning results would be very different. The second reason is that learning from a cost ratio other than the given cost ratio may maximize classifier utility [4]. Therefore, refining the cost matrix given by users or learning a cost matrix via the interaction with users is desired for a cost-sensitive system. However, this remains an open problem in the study of cost-sensitive learning.

Cross-validation is widely used for parameter tuning and if we regard the cost ratios as parameters, it can be used for selecting a proper cost setting. Generally, to get the result (false acceptance/rejection/identification ratio) under one possible parameter setting, one classifier needs to be trained. For example, if we vary C_{IG}/C_{GI} among $\{0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20\}$ and C_{GI}/C_{GG} among $\{0.1, 0.2, 0.5, 1, 2, 5, 10\}$, $9 \times 7 = 63$ classifiers are needed. So, a direct application of cross validation is quite inefficient.

In this paper, we provide an attempt to learning the cost matrix in an efficient way, where only one classifier is needed to be trained for obtaining results for all parameter settings. Notice in mc$k$NN, the training process of learning the conditional and prior probabilities is independent with the cost matrix. In the test process, firstly the k nearest neighbors of a probe image are identified and then the label of the probe image is decided using the neighbor information and the cost ratios. The computation cost of the second step is negligible if compared



AR: *PCA* AR: *LPP* AR: *LDA* AR: *LBP*

FERET: *PCA* FERET: *LPP* FERET: *LDA* FERET: *LBP*

Fig. 5. Comparison of kNN and mc$k$NN with different settings of k. The "database: *feature*" under each figure indicates the database and feature used in the experiments.

with that of the first step. Therefore, we can get all the results (err$_{IG}$, err$_{GI}$ and err$_{GG}$) under different cost settings simultaneously by training only one classifier. mcKLR has similar properties. The training process is independent with cost ratio and in the test process, after getting the posterior probability, the cost ratio is used to adjust the prediction. Therefore, we can get all the results under different cost settings simultaneously.

After obtaining the results under different cost settings on the validation sets, the user can pick a result which is believed to be the best suited to her/is intention, and then the corresponding cost setting will be used for the training process.

In the following 5-fold cross-validation experiments, we vary C_{IG}/C_{GI} among {0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20} and C_{GI}/C_{GG} among {0.1, 0.2, 0.5, 1, 2, 5, 10}. The results of mc$k$NN and mcKLR on AR database using PCA feature are shown in Fig. 6. The results are obtained as the average on the 5 validation sets. The x-axis and y-axis (both in log-scale) correspond to the cost ratio C_{IG}/C_{GI} and C_{GI}/C_{GG}, respectively. The z-axis is the error rate. We only show err$_{GI}$ and err$_{IG}$ for simplicity. After observing the results, the user may be able to decide which kind of tradeoff between the misclassifications is suited to her/is intention.

7 Conclusion

This paper extends our preliminary research [24] which argues that simply pursuing a low error rate in face recognition is not as reasonable as it might have been expected before, because different kinds of mistakes generally lead to different amount of losses. We formulate face recognition as a multi-class cost-sensitive learning task and under such formulation, we try to minimize the total cost rather than the total error rate. To the best of our knowledge, this is the first study on cost-sensitive face recognition. We propose two simple methods, mc$k$NN and mcKLR. The former is suitable for situations where the group members vary frequently while the latter is more suitable for situations with stable group members. Compared with ROC-based methods which embed the cost information implicitly via adapting the acceptance threshold, our methods optimize the total cost directly and thus obtain better performance. Experiments show that cost-sensitive methods is always better than their cost-blind counterparts on all databases with any features, while the mcKLR method usually achieves the best performance. In addition, rather than simply waiting the user to give the cost matrix, in this paper we also present a study on learning the cost matrix via interaction with the user.

Acknowledgments

The authors would like to thank Ji Zhu for guidance on implementing mbKLR, Yoonkyung Lee for sharing the code of mcSVM, Xu-Ying Liu and Yu-Yin Sun for reading a preliminary draft, and the helpful comments and suggestions from the associate editor and anonymous reviewers.

References

[1] N. Abe, B. Zadrozny, and J. Langford, "An iterative method for multi-class cost-sensitive learning," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, 2004, pp. 3–11.

[2] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 4, pp. 711–720, 1997.

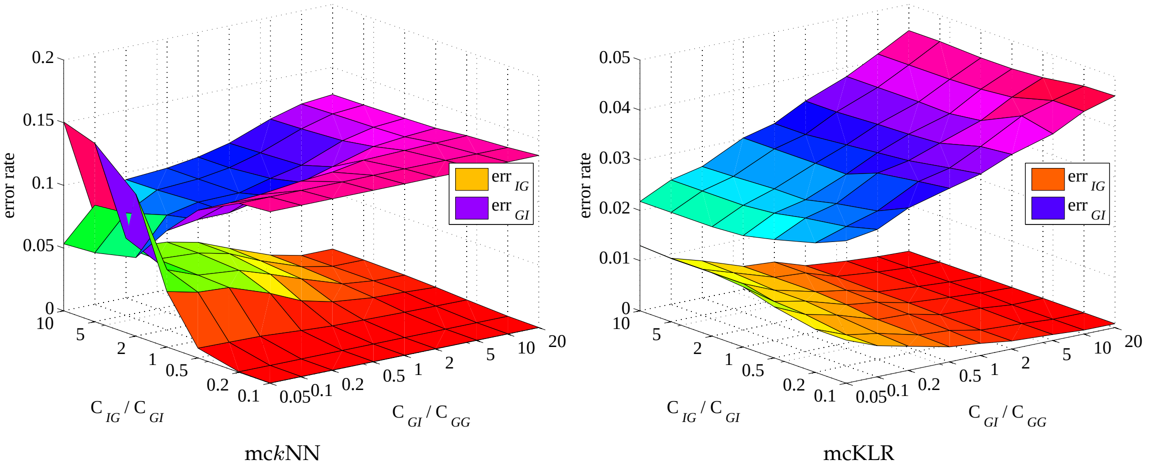


Fig. 6. Comparison of err$_{IG}$ and err$_{GI}$ of mc$k$NN and mcKLR under different cost settings.

[4] M. Ciraco, M. Rogalewski, and G. M. Weiss, "Improving classifier utility by altering the misclassification cost ratio," in *Proceedings of the KDD'05 Workshop on Utility-Based Data Mining*, Chicago, IL, 2005, pp. 46–52.

[5] P. Domingos, "MetaCost: A general method for making classifiers cost-sensitive," in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, 1999, pp. 155–164.

[6] C. Drummond and R. C. Holte, "C4.5 and class imbalance and cost sensitivity: Why under-sampling beats over-sampling," in *Working Notes of the ICML'03 Workshop on Learning from Imbalanced Datasets*, Washington, DC, 2003.

[7] C. Elkan, "The foundations of cost-sensitive learning," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, Seattle, WA, 2001, pp. 973–978.

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer, 2001.

[9] X. He, S. Yan, Y. Hu, P. Niyogi, and H. J. Zhang, "Face recognition using laplacian faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

[10] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: Global versus component-based approach," in *Proceedings of the 8th IEEE International Conference on Computer Vision*, vol. 2, Vancouver, Canada, 2001, pp. 688–694.

[11] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 4–20, 2004.

[12] T. C. Landgrebe and R. P. Duin, "Efficient multiclass ROC approximation by decomposition via confusion matrix perturbation analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 801–822, 2008.

[13] Y. Lee, Y. Lin, and G. Wahba, "Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 67–81, 2004.

[14] F. Li and H. Wechsler, "Open set face recognition using transduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1686–1697, 2005.

[15] Z. Li and X. Tang, "Bayesian face recognition using support vector machine and face clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, Washington, DC, 2004, pp. 374–380.

[16] X.-Y. Liu and Z.-H. Zhou, "The influence of class imbalance on cost-sensitive learning: An empirical study," in *Proceedings of the 6th IEEE International Conference on Data Mining*, Hong Kong, China, 2006, pp. 970–974.

[17] M. A. Maloof, "Learning when data sets are imbalanced and when costs are unequal and unknown," in *Working Notes of the ICML'03 Workshop on Learning from Imbalanced Datasets*, Washington, DC, 2003.

[18] A. M. Martinez and R. Benavente, "The AR face database," CVC, Tech. Rep. 24, 1998.

[19] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, 1997.

[20] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[21] K. M. Ting, "An empirical study of MetaCost using boosting algorithms," in *Proceedings of the 11th European Conference on Machine Learning*, Barcelona, Spain, 2000, pp. 413–425.

[22] ——, "An instance-weighting method to induce cost-sensitive trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 3, pp. 659–665, 2002.

[23] M. A. Turk and A. Pentland, "Eigenface for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[24] Y. Zhang and Z.-H. Zhou, "Cost-sensitive face recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008.

[25] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, no. 1, pp. 399–458, 2003.

[26] Z.-H. Zhou and X.-Y. Liu, "On multi-class cost-sensitive learning," in *Proceedings of the 21st National Conference on Artificial Intelligence*, Boston, MA, 2006, pp. 567–572.

[27] J. Zhu and T. Hastie, "Classification of gene microarrays by penalized logistic regression," *Biostatistics*, vol. 5, no. 3, pp. 427–443, 2004.

Yin Zhang received the BSc degree in computer science from Fudan University, China, in 2007. He is now a graduate student in the Department of Computer Science & Technology at Nanjing University, and is a member of the LAMDA Group. His research interest is in computer vision and machine learning.

Zhi-Hua Zhou (S'00-M'01-SM'06) received the BSc, MSc and PhD degrees in computer science from Nanjing University, China, in 1996, 1998 and 2000, respectively, all with the highest honors.

He joined the Department of Computer Science & Technology at Nanjing University as an assistant professor in 2001, and is currently Cheung Kong Professor and Director of the LAMDA group. His research interests are in artificial intelligence, machine learning, data mining, pattern recognition, information retrieval, evolutionary computation and neural computation. In these areas he has published over 70 papers in leading international journals or conference proceedings.

Dr. Zhou has won various awards/honors including the National Science & Technology Award for Young Scholars of China (2006), the Award of National Science Fund for Distinguished Young Scholars of China (2003), the National Excellent Doctoral Dissertation Award of China (2003), the Microsoft Young Professorship Award (2006), etc. He is an Associate Editor of *IEEE Transactions on Knowledge and Data Engineering*, Associate Editor-in-Chief of *Chinese Science Bulletin*, and on the editorial boards of *Artificial Intelligence in Medicine*, *Intelligent Data Analysis*, *Science in China*, etc. He is the founder of ACML, Steering Committee member of PAKDD and PRICAI, Program Committee Chair/Co-Chair of PAKDD'07, PRICAI'08 and ACML'09, vice Chair or area Chair of conferences including IEEE ICDM'06, IEEE ICDM'08, SIAM DM'09, ACM CIKM'09, etc., and General Chair/Co-Chair or Program Committee Chair/Co-Chair of a dozen of native conferences. He is the chair of the Machine Learning Society of the Chinese Association of Artificial Intelligence (CAAI), vice chair of the Artificial Intelligence & Pattern Recognition Society of the China Computer Federation (CCF), and chair of the IEEE Computer Society Nanjing Chapter. He is a member of AAAI and ACM, and a senior member of IEEE, IEEE Computer Society and IEEE Computational Intelligence Society.