Cost-Sensitive Face Recognition

Yin Zhang and Zhi-Hua Zhou*
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
{zhangyin,zhouzh}@lamda.nju.edu.cn

Abstract

Traditional face recognition systems attempt to achieve a high recognition accuracy, which implicitly assumes that the losses of all misclassifications are the same. However, in many real-world tasks this assumption is not always reasonable. For example, it will be troublesome if a face-recognition-based door-locker misclassifies 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. In this paper, we propose a framework which formulates the problem as a multi-class cost-sensitive learning task, and propose a theoretically sound method based on Bayes decision theory to solve this problem. Experimental results demonstrate the effectiveness and efficiency of the proposed method.

1. Introduction

Face recognition has attracted much research effort for many years and many successful face recognition systems emerge [15, 3, 13]. To the best of our knowledge, most of those face recognition systems try to pursue high accuracy, which implicitly assumes that any misclassification will cause the same amount of loss since they simply try to minimize the number of mistakes. However, for many real-world applications, such assumption is not always reasonable. For example, considering a face recognition system-based door-locker for a certain group (*e.g.*, a group of family members or roommates, *etc.*), there are four different types of recognition errors: 1) mis-recognizing a stranger as a group member, 2) mis-recognizing a group member as a stranger, 3) mis-recognizing between two group members and 4) mis-recognizing between two strangers. In traditional face recognition systems, these errors are treated equally. However, it is evident that these errors will cause different losses. When the second error occurs, a group member is mistakenly rejected, which is troublesome. But compared with the first error, the second one is not so serious, since it may be a disaster if a stranger is mistakenly allowed to enter the house. The third error also causes some trouble since in the house members may have different private rooms, but such an error is obviously much less serious than the first and the second ones. The last error is negligible since strangers are not allowed to enter the house, no matter who the stranger is. In particular, it is not possible to provide the system the face images of all possible strangers, so it is almost inevitable that an unseen stranger might be recognized as another stranger. But since both strangers should be rejected, the system does not lose anything if such error occurs. Therefore, those four types of errors are quite different and simply taking accuracy as the measure of the performance is not a good choice.

In the machine learning and data mining community, a kind of classification algorithms called *cost-sensitive learning* has been studied for years [2, 4, 6, 9, 16]. Under that framework, 'cost' information is introduced to measure the loss of misclassification and different costs reflect different types of losses. The purpose of cost-sensitive learning is to minimize the total cost. There are two kinds of cost-sensitive problems, *i.e.*, problems with *class-dependent cost* [4, 6, 9, 16] and problems with *example-dependent cost* [2]. When cost is class-dependent, cost is determined by error type. That is, misclassifying any example of ith class as jth class will always have the same cost while misclassifying an example as different classes may have different costs. When cost is example-dependent, examples' misclassification costs are different from each other, even when the error type is the same.

Inspired by cost-sensitive learning, we can formulate the face recognition problem mentioned above as a multi-class cost-sensitive learning problem. In this paper, we consider a situation that, letting any stranger in will lead to the same loss and misclassifying a group member as another group member or a stranger will have different losses. So our problem is a class-dependent class-sensitive problem. Then

*This research was supported by the National Science Foundation of China (60635030, 60721002) and the National High Technology Research and Development Program of China (2007AA01Z169).

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

we try to minimize the total cost instead of error rate as in conventional face recognition systems, aiming to prevent disasters caused by mistakes with large costs. Since existing methods could not handle the problem well, we propose a new method called *mcKLR* which is derived from Bayes decision theory. Experimental results validate the effectiveness and efficiency of our method.

The rest of this paper is organized as follows. In Section 2 we formulate cost-sensitive face recognition problem. In Section 3 we briefly introduce some existing multi-class cost-sensitive learning methods. Then we propose the mcKLR method in Section 4 and report on the experiments in Section 5. Finally, we conclude the paper in Section 6.

2. Problem Formulation

Denote a face image by $\boldsymbol{x}$ and y for its label. Considering that there are N 'in-group' people and many (say, M[1]) 'out-group' people, denoted by $y = G_1, \ldots, G_N$ and $O_1, \ldots, O_M$, respectively. Conventional face recognition systems try to generate a hypothesis $\phi(\boldsymbol{x})$ minimizing the expectation **error rate**: Err $= E_{\boldsymbol{x},y}\big(I(\phi(\boldsymbol{x}) \neq y)\big)$, where I is indicator function: 1 when $\phi(\boldsymbol{x}) \neq y$ and 0 otherwise. It means that they implicitly assume the costs of all kinds of mistakes are the same. However, as mentioned before, such assumption is not always reasonable and different mistakes are associated with different costs. Given a cost matrix $\mathbf{C}$[2] as shown in Table 1, C_{ij} indicates the cost of misclassifying the ith person as the jth. It is an $(N{+}M)$-class cost-sensitive learning problem and the hypothesis $\phi(\boldsymbol{x})$ should minimize the expectation **cost**: Cost $= E_{\boldsymbol{x},y}(C_{y\phi(\boldsymbol{x})})$. Because $E_{\boldsymbol{x},y}(C_{y\phi(\boldsymbol{x})}) = E_{\boldsymbol{x}}\left(E_{y|\boldsymbol{x}}(C_{y\phi(\boldsymbol{x})}|\boldsymbol{x})\right)$, 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 $\boldsymbol{x}$. Therefore we can define the expectation 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

$$\text{loss}\big(\boldsymbol{x}, \phi(\boldsymbol{x})\big) = \begin{cases} \sum\limits_{\substack{n=1 \\ n\neq k}}^{N} \mathbf{P}(G_n|\boldsymbol{x}) C_{G_n G_k} + \sum\limits_{m=1}^{M} \mathbf{P}(O_m|\boldsymbol{x}) C_{O_m G_k} & \\ & \text{if } \phi(\boldsymbol{x}) = G_k \\ \sum\limits_{n=1}^{N} \mathbf{P}(G_n|\boldsymbol{x}) C_{G_n O_k} + \sum\limits_{\substack{m=1 \\ m\neq k}}^{M} \mathbf{P}(O_m|\boldsymbol{x}) C_{O_m O_k} & \\ & \text{if } \phi(\boldsymbol{x}) = O_k \end{cases} \quad (1)$$

[1] M need not be specified since the method should be able to deal with strangers who do not appear in training set.

[2] Here we assume that there is a cost matrix given by user. Usually it is easy for a user to specify which kind of mistake is with a higher cost and which is with a lower cost. In this paper we only study how to make the face recognition system behaves well given a cost matrix. Refining the cost matrix given by the user or automatically learning a cost matrix from the data will be studied in the future.

Table 1: The cost matrix for cost-sensitive face recognition

	G_1	...	G_N	O_1	...	O_M
G_1	0	...	$C_{G_1G_N}$	$C_{G_1O_1}$	...	$C_{G_1O_M}$
...	...	...	...	...	...	...
G_N	$C_{G_NG_1}$	...	0	$C_{G_NO_1}$	...	$C_{G_NO_M}$
O_1	$C_{O_1G_1}$	...	$C_{O_1G_N}$	0	...	$C_{O_1O_M}$
...	...	...	...	...	...	...
O_M	$C_{O_MG_1}$	...	$C_{O_MG_N}$	$C_{O_MO_1}$	...	0

Table 2: The reduced cost matrix

	G_1	...	G_n	O
G_1	0	...	C_{GG}	C_{GO}
...	...	...	...	...
G_n	C_{GG}	...	0	C_{GO}
O	C_{OG}	...	C_{OG}	0

where we denote $\mathbf{P}(y = G_n|\boldsymbol{x})$ and $\mathbf{P}(y = O_m|\boldsymbol{x})$ as $\mathbf{P}(G_n|\boldsymbol{x})$ and $\mathbf{P}(O_m|\boldsymbol{x})$, respectively, for simplicity. Therefore, in order to minimize the total cost, the optimal prediction of $\boldsymbol{x}$ should be

$$\phi^*(\boldsymbol{x}) = \arg \min_{\phi(\boldsymbol{x}) \in \{G_1,\ldots,G_N,O_1\ldots,O_M\}} \text{loss}\big(\boldsymbol{x}, \phi(\boldsymbol{x})\big) \quad (2)$$

Here we can categorize the costs into four types: 1) cost of misclassifying an 'out-group' person as 'in-group', C_{OG}; 2) cost of misclassifying an 'in-group' person as 'out-group', C_{GO}; 3) cost of misclassification between two 'in-group' persons, C_{GG}; and 4) cost of misclassification between two 'out-group' persons, C_{OO}. According to our discussion above, it is evident that $C_{OG} \gg C_{GO} > C_{GG} > C_{OO} = 0$. We can consider all the 'out-group' people as belonging to a meta-class O, where $y = O \iff \exists m\ y = O_m$. So $\mathbf{P}(O|\boldsymbol{x}) = \sum_{m=1}^{M} P(O_m|\boldsymbol{x})$. Then the two parts of (1) can be rewritten as:

$$\begin{aligned} \text{loss}(\boldsymbol{x}, G_k) &= \sum_{\substack{n=1 \\ n\neq k}}^{N} \mathbf{P}(G_n|\boldsymbol{x}) C_{GG} + \sum_{m=1}^{M} \mathbf{P}(O_m|\boldsymbol{x}) C_{OG} \\ &= \sum_{\substack{n=1 \\ n\neq k}}^{N} \mathbf{P}(G_n|\boldsymbol{x}) C_{GG} + \mathbf{P}(O|\boldsymbol{x}) C_{OG} \quad (3) \end{aligned}$$

and

$$\begin{aligned} \text{loss}(\boldsymbol{x}, O) &= \sum_{n=1}^{N} \mathbf{P}(G_n|\boldsymbol{x}) C_{GO} + \sum_{\substack{m=1 \\ m\neq k}}^{M} \mathbf{P}(O_m|\boldsymbol{x}) C_{OO} \\ &= \sum_{n=1}^{N} \mathbf{P}(G_n|\boldsymbol{x}) C_{GO} \quad (4) \end{aligned}$$

Therefore it is equivalent to an (N+1)-class cost-sensitive problem and the cost matrix $\mathbf{C}$ can be reduced to the one shown in Table 2. We can use multi-class cost-sensitive learning algorithms to solve this problem.

3. Multi-Class Cost-Sensitive Learning

3.1. Rescaling

Rescaling [6, 16] is a general approach which can be used to make any cost-blind learning algorithms cost-sensitive. The principle is to enable the influences of the higher-cost classes be bigger 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, sampling the classes according to their costs, or threshold-moving [6, 5]. This approach is effective in dealing with binary-class problems. Zhou and Liu [16] indicated that it is still helpful to multi-class problem only when all the classes can be rescaled simultaneously. They also revealed that for an (N+1)-class problem, if each class can be assigned with an optimal weight w_n ($1 \leqslant n \leqslant N+1$, $w_n > 0$) after rescaling simultaneously, $\boldsymbol{w} = [w_1, w_2, \ldots, w_{N+1}]^{\mathbf{T}}$ must be the non-trivial solution of a linear equations system with the coefficient matrix:

$$\begin{bmatrix} C_{21} & -C_{12} & 0 & \ldots & 0 \\ C_{31} & 0 & -C_{13} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & 0 \\ C_{(N+1)1} & 0 & 0 & \ldots & -C_{1(N+1)} \\ 0 & C_{32} & -C_{23} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & 0 \\ 0 & C_{(N+1)2} & 0 & \ldots & -C_{2(N+1)} \\ 0 & 0 & 0 & \ldots & -C_{N(N+1)} \end{bmatrix} \tag{5}$$

It is equivalent to require the coefficient matrix (5) have a rank smaller than N+1. In our cost-sensitive face recognition task, the coefficient matrix's rank is N. Therefore theoretically we can use rescaling to solve this problem.

Another popular method, *MetaCost* [4], can also be considered as rescaling, since it relabels training examples to minimize Bayesian risk by threshold moving.

However, our experimental results reveal that rescaling methods do not work well on our cost-sensitive face recognition task and the reason will be explained in section 5.

3.2. Multi-Class Cost-Sensitive SVM (mcSVM)

Support vector machines (SVM) have been successfully applied to face recognition [7, 10]. Since SVM is originally designed for binary classification and our cost-sensitive face recognition task is a multi-class problem, we need to extend it to multi-class case. The *one-vs-one* and the *one-vs-all* strategies are popular in decomposing a multi-class problem into a series of binary-class problems. However, these approaches may fail under various circumstances [9, 10]. Lee *et al*. [9] derived a multi-class cost-sensitive SVM, *i.e*. mcSVM. In this method, for an (N+1)-class classification problem, the instance $\boldsymbol{x}$'s label y is extended to an (N+1)-dimensional label vector, denoted by $\boldsymbol{y}$. $\boldsymbol{y}$ takes 1 on the yth coordinate and $-1/N$ on the others. Accordingly, an (N+1)-tuple of separating functions $\boldsymbol{f}(\boldsymbol{x}) = \big(f_1(\boldsymbol{x}), \ldots, f_{N+1}(\boldsymbol{x})\big)$ is defined, where $f_n(\boldsymbol{x}) = h_n(\boldsymbol{x}) + b_n$, $h_n \in H_K$ and $b_n \in \mathbb{R}$. H_K is a reproducing kernel Hilbert space (RKHS) with the reproducing kernel function $K(\cdot,\cdot)$. $\boldsymbol{f}(\boldsymbol{x})$ is with the sum-to-zero constraint $\sum_{n=1}^{N+1} f_n(\boldsymbol{x}) = 0$ for any $\boldsymbol{x}$.

Define the loss function for mcSVM as $\mathbf{L}\big(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}\big) = \mathbf{C}(\boldsymbol{y}) \cdot \big(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{y}\big)_+$, where $\mathbf{C}(\boldsymbol{y})$ is the yth row of the cost matrix $\mathbf{C}$ and $\big(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{y}\big)_+$ is $\big((f_1(\boldsymbol{x}) - \boldsymbol{y}_1)_+, \ldots, (f_{N+1}(\boldsymbol{x}) - \boldsymbol{y}_{(N+1)})_+\big)$. Lee *et al*. [9] 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 $\boldsymbol{f}^*(\boldsymbol{x}) = \big(f_1^*(\boldsymbol{x}), \ldots, f_{N+1}^*(\boldsymbol{x})\big)$ with

$$f_k^*(\boldsymbol{x}) = \begin{cases} 1 & \text{if } k = \arg\min\limits_{n=1,\ldots,N+1} \text{loss}(\boldsymbol{x}, n) \\ -1/N & \text{otherwise.} \end{cases} \tag{6}$$

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

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

$$\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \mathbf{C}(\boldsymbol{y}_i) \cdot \big(\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{y}_i\big)_+ + \frac{1}{2}\lambda \sum_{n=1}^{N+1} \|f_n\|_{H_K}^2 \tag{7}$$

We find in our experiments that the performance of mcSVM is better than cost-blind methods. However, it is far from a 'good enough' method and the method described in the next section is superior to it on the cost-sensitive face recognition task.

4. Our Method

4.1. Derivation

Similar to O, we can define another meta-class G as $y = G \iff \exists n\ y = G_n$ and $\mathbf{P}(G|\boldsymbol{x}) = \sum_{n=1}^{N} \mathbf{P}(G_n|\boldsymbol{x})$. So from (3) we have

$$\begin{aligned} \text{loss}(\boldsymbol{x}, G_k) &= \sum_{\substack{n=1 \\ n \neq k}}^{N} \mathbf{P}(G_n|\boldsymbol{x}) C_{GG} + \mathbf{P}(O|\boldsymbol{x}) C_{OG} \\ &= \big(\mathbf{P}(G|\boldsymbol{x}) - \mathbf{P}(G_k|\boldsymbol{x})\big) C_{GG} + \mathbf{P}(O|\boldsymbol{x}) C_{OG} \\ &= \mathbf{P}(G|\boldsymbol{x}) C_{GG} + \mathbf{P}(O|\boldsymbol{x}) C_{OG} - \mathbf{P}(G_k|\boldsymbol{x}) C_{GG} \end{aligned} \tag{8}$$

As $\boldsymbol{x}$ can be labeled as either G or O, we have $\mathbf{P}(G|\boldsymbol{x}) + \mathbf{P}(O|\boldsymbol{x}) = 1$. So (8) becomes

$$\begin{aligned}\text{loss}(\boldsymbol{x}, G_k) =& \big(1 - \mathbf{P}(O|\boldsymbol{x})\big)C_{GG} + \mathbf{P}(O|\boldsymbol{x})C_{OG} \\ & - \mathbf{P}(G_k|\boldsymbol{x})C_{GG} \\ =& C_{GG} + \mathbf{P}(O|\boldsymbol{x})(C_{OG} - C_{GG}) \\ & - \mathbf{P}(G_k|\boldsymbol{x})C_{GG}\end{aligned} \tag{9}$$

And (4) becomes

$$\text{loss}(\boldsymbol{x}, O) = \sum_{n=1}^{N} \mathbf{P}(G_n|\boldsymbol{x})C_{GO} = \mathbf{P}(G|\boldsymbol{x})C_{GO} \tag{10}$$

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

$$\begin{cases} C_{GG} + \mathbf{P}(O|\boldsymbol{x})(C_{OG} - C_{GG}) - \mathbf{P}(G_1|\boldsymbol{x})C_{GG} \\ \vdots \\ C_{GG} + \mathbf{P}(O|\boldsymbol{x})(C_{OG} - C_{GG}) - \mathbf{P}(G_N|\boldsymbol{x})C_{GG} \\ \mathbf{P}(G|\boldsymbol{x})C_{GO} \end{cases} \tag{11}$$

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

$$\begin{aligned} & \mathbf{P}(G|\boldsymbol{x})C_{GO} - C_{GG} - \mathbf{P}(O|\boldsymbol{x})(C_{OG} - C_{GG}) \\ = \;& \big(1 - \mathbf{P}(O|\boldsymbol{x})\big)C_{GO} - C_{GG} - \mathbf{P}(O|\boldsymbol{x})(C_{OG} - C_{GG}) \\ = \;& -\mathbf{P}(O|\boldsymbol{x})(C_{GO} + C_{OG} - C_{GG}) + (C_{GO} - C_{GG}) \end{aligned}$$

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

$$\begin{cases} -\mathbf{P}(G_1|\boldsymbol{x})C_{GG} \\ \vdots \\ -\mathbf{P}(G_N|\boldsymbol{x})C_{GG} \\ -\mathbf{P}(O|\boldsymbol{x})(C_{GO} + C_{OG} - C_{GG}) + (C_{GO} - C_{GG}) \end{cases} \tag{12}$$

Divide $-C_{GG}$ from every item and denote $\beta = (C_{GO} + C_{OG} - C_{GG})/C_{GG}$ and $\Delta = (C_{GO} - C_{GG})/C_{GG}$. Then the problem becomes choosing the maximum from

$$\begin{cases} \mathbf{P}(G_1|\boldsymbol{x}) \\ \vdots \\ \mathbf{P}(G_N|\boldsymbol{x}) \\ \beta\mathbf{P}(O|\boldsymbol{x}) - \Delta \end{cases} \tag{13}$$

4.2. Optimization

Define an N-tuple of separating functions $\boldsymbol{f}(\boldsymbol{x}) = \big(f_1(\boldsymbol{x}), \ldots, f_N(\boldsymbol{x})\big)$ and loss function $\mathbf{L}\big(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}), y\big)$ as

$$\begin{aligned}\mathbf{L}\big(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}), y\big) =& \sum_{k=1}^{N} \left(-\ln \frac{e^{f_k(\boldsymbol{x})}}{1 + \sum_{n=1}^{N} e^{f_n(\boldsymbol{x})}} \right) I(y = G_k) \\ & + \left(-\ln \frac{1}{1 + \sum_{n=1}^{N} e^{f_n(\boldsymbol{x})}} \right) I(y = O)\end{aligned} \tag{14}$$

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 $\mathbf{L}$. Because $E_{\boldsymbol{x},y}(\mathbf{L}) = E_{\boldsymbol{x}}\big(E_{y|\boldsymbol{x}}(\mathbf{L}|\boldsymbol{x})\big)$, in order to minimize $E_{\boldsymbol{x},y}(\mathbf{L})$ we can minimize $E_{y|\boldsymbol{x}}(\mathbf{L}|\boldsymbol{x})$ on every $\boldsymbol{x}$, where

$$\begin{aligned}E_{y|\boldsymbol{x}}(\mathbf{L}|\boldsymbol{x}) =& \sum_{k=1}^{N} \left(-\ln \frac{e^{f_k(\boldsymbol{x})}}{1 + \sum_{n=1}^{N} e^{f_n(\boldsymbol{x})}} \right) \mathbf{P}(G_k|\boldsymbol{x}) \\ & + \left(-\ln \frac{1}{1 + \sum_{n=1}^{N} e^{f_n(\boldsymbol{x})}} \right) \mathbf{P}(O|\boldsymbol{x})\end{aligned} \tag{15}$$

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

$$f_k^*(\boldsymbol{x}) = \ln \frac{\mathbf{P}(G_k|\boldsymbol{x})}{\mathbf{P}(O|\boldsymbol{x})} \tag{16}$$

Through $f_1^*, \ldots, f_N^*$ we construct a new function f_O^*:

$$\begin{aligned} f_O^*(\boldsymbol{x}) &= \ln \frac{\beta\mathbf{P}(O|\boldsymbol{x}) - \Delta}{\mathbf{P}(O|\boldsymbol{x})} \\ &= \ln \left(\beta - \Delta \left(1 + \sum_{k=1}^{N} e^{f_k^*(\boldsymbol{x})} \right) \right) \end{aligned} \tag{17}$$

Therefore, choosing the maximum from $\{f_1^*(\boldsymbol{x}), \ldots, f_N^*(\boldsymbol{x}), f_O^*(\boldsymbol{x})\}$ is equivalent to choosing the maximum from (13). That is, the optimal predicted label of $\boldsymbol{x}$ under Bayes decision rule is

$$\phi(\boldsymbol{x}) = \begin{cases} G_k & \text{if } f_k^* \text{ is the maximum} \\ O & \text{if } f_O^* \text{ is the maximum} \end{cases}$$

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

$$\begin{aligned}\mathbf{L}(\mathcal{D}) =& \sum_{i=1}^{|\mathcal{D}|} \left(\sum_{k=1}^{N} \left(-\ln \frac{e^{f_k(\boldsymbol{x}_i)}}{1 + \sum_{n=1}^{N} e^{f_n(\boldsymbol{x}_i)}} \right) I(y_i = G_k) \right. \\ & \left. + \left(-\ln \frac{1}{1 + \sum_{n=1}^{N} e^{f_n(\boldsymbol{x}_i)}} \right) I(y_i = O) \right)\end{aligned} \tag{18}$$

As did in mcSVM, assume $f_n(\boldsymbol{x}) = h_n(\boldsymbol{x}) + b_n$, $h_n \in H_K$ and $b_n \in \mathbb{R}$. The optimization object can be expressed as

$$\mathbf{L}(\mathcal{D}) + \frac{1}{2}\lambda \sum_{n=1}^{N} \|f_n\|_{H_K}^2 \tag{19}$$

Note that the optimization problem is similar to the optimization form of multi-class kernel logistic regression (KLR) [17]. Therefore, we can use the similar optimization technique of KLR to handle our problem and we call our method mcKLR.

5. Experiments

5.1. Configuration

The AR [12], FERET [14], Extended YaleB [8] and ORL [1] face databases are used in our experiments. In the AR database, 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. In the FERET database we choose images of frontal view with different expression and illumination for our experiment. The preprocessing taken on FERET images is similar to that on AR except that the mask is 75×65. As for the YaleB database, we use the frontal view images and a 32×32 mask. As for the ORL database, all images are used and cropped by a 32×32 mask[3].

PCA is applied to the images. Then we randomly select N subjects as 'in-group' and M subjects as 'out-group'. Only M_{train} among the M 'out-group' subjects appear in the training set while the remaining $M - M_{train}$ do not. We set $M_{train} \ll M$ to simulate the true scenario that the face recognition system could only get quite a little part of the outside world and it should be able to classify unobserved 'out-group' people. Every experiment is repeated for 20 times and the average results are recorded.

We study four cost-blind methods, including nearest neighbor (abbreviated as 1-NN), LDA+nearest neighbor (abbreviated as LDA) and multi-class cost-blind SVM (abbreviated as mbSVM), multi-class cost-bind KLR (abbreviated as mbKLR) and four cost-sensitive methods, including rescaling, MetaCost, mcSVM and our method mcKLR. As for 1-NN, the gallery set is also the training set. As for LDA, linear discriminant analysis is used first to find the optimal linear subspace and then the nearest neighbor classifier is used to identify the probe image. For the rescaling method, we first resample the training set and then train mbSVM. For MetaCost, we resample 20 times with every time 80% training data to estimate posterior probability and mbSVM is used as its classifier. mbSVM is the cost-blind version of mcSVM where all cost is the same. For mbSVM , mbKLR, mcSVM and mcKLR, RBF kernel is used and the width and the regulation coefficient are selected from e^5 to e^{-5} and from e^0 to e^{-10}, respectively, with a method similar to five-fold cross-validation. In detail, we partition the images of every 'in-group' subject and all 'out-group' subjects into five subsets. For 'in-group' subjects, typical five-fold cross validation is used. But for 'out-group' subjects, in each run we only put the images of one subset into training set while the remaining four subsets into validation set. In this way, we simulate the true scenario where the face recognition system should be able to deal with unseen 'out-group' people.

[3]The cropped YaleB and ORL databases are obtained from http://www.cs.uiuc.edu/homes/dengcai2/Data/data.html

5.2. Results

We study these methods under three varying influence factors: the number of 'in-group' subjects, the number of unobserved 'out-group' subjects, and the cost ratio.

First we fix those influence factors. For every database, each 'out-group' subject has one image for testing and at most one image for training. The numbers of training images of 'in-group' subjects are different on different databases. On AR, N=3, M=60, M_{train}=30, each 'in-group' subject has 7 images for training and 7 images for testing; on FERET, N=4, M=600, M_{train}=50, each 'in-group' subject has 7 images for training and 3 images for testing; on YaleB, N=4, M=30, M_{train}=10, each 'in-group' subject has 5 images for training and 5 images for testing; on ORL, N=4, M=35, M_{train}=10, each 'in-group' subject has 5 images for training and 5 images for testing. The cost ratio C_{GG}:C_{GO}:C_{OG} is set to 1:4:200 on all databases. We compare the total cost, total error rate (err), error rate of misclassifying 'out-group' subjects as 'in-group' (err_{OG}) and error rate of misclassifying 'in-group' subjects as 'out-group' (err_{GO}). The results are shown in Table 3.

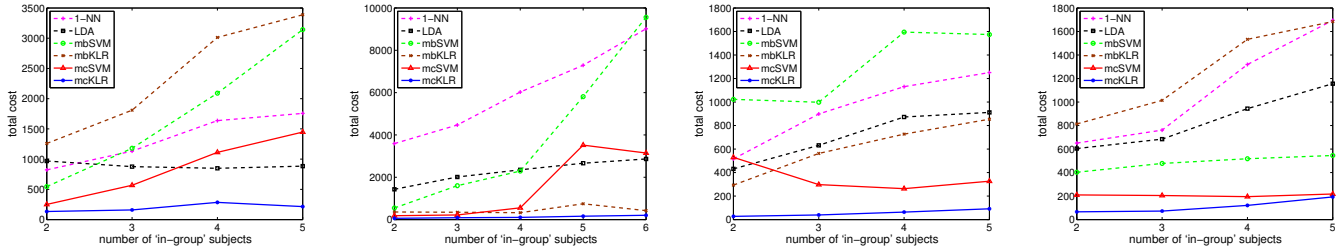
The results of rescaling and MetaCost are not presented in the table since they simply predict every image as 'out-group' and such prediction is of no use. We believe that the reason of its failure is that there are much more images of the 'out-group' class (as there may be tens of subjects) than that of every 'in-group' class (as there is only one subject). Thus there exists class imbalance. In fact, Liu and Zhou [11] has studied this problem and indicated that if class imbalance and cost-sensitive occur simultaneously, to rescale the classes in proportion to the cost ratio is no more optimal. However, determining the optimal rescaling ratio in this case is still an open problem.

From Table 3 we can find that the cost-sensitive methods have much smaller total cost than cost-blind methods although the total error rate may not be lower. It is evident that the cost-sensitive methods implement this by preventing high-cost errors while slightly increasing low-cost errors, which can be observed by comparing the performance of mbSVM and mcSVM. It is impressive that mcKLR achieves the smallest total cost on all the databases.

Then, we study the performance of the compared methods with different number of 'in-group' subjects, *i.e.*, with varying N. For FERET, N varies from 2 to 6; for AR, YaleB, and ORL, N varies from 2 to 5. The results are shown in Figure 1. Generally the performance of cost-sensitive methods are superior to cost-blind methods, although there are cases where LDA is better than mcSVM.

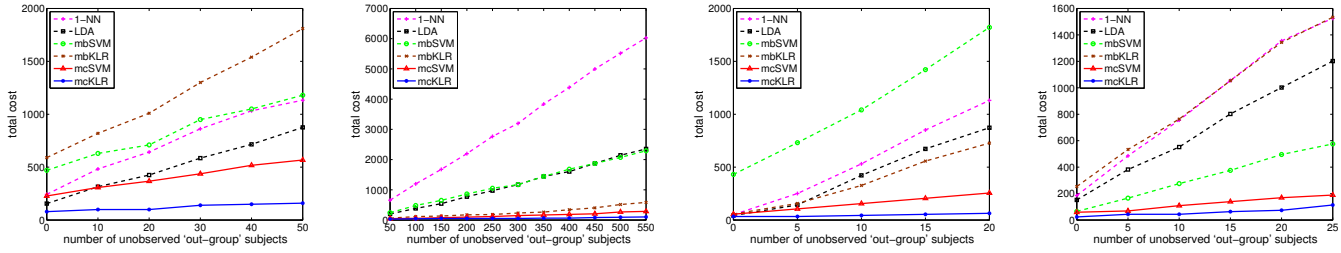
Table 3: Comparison on total cost, total error rate (err), high-cost error rate (err_{OG}), and low-cost error rate(err_{GO})

Database		Cost-blind methods				Cost-sensitive methods	
		1-NN	*LDA*	*mbSVM*	*mbKLR*	*mcSVM*	*mcKLR*
FERET	cost	6026.5	2352.8	2292.0	583.0	309.0	**102.8**
	$\text{err}_{OG}(\%)$	5.00	1.96	1.89	0.47	0.22	**0.07**
	$\text{err}_{GO}(\%)$	55.00	**5.83**	45.83	47.92	56.25	47.50
	err(%)	6.00	2.03	2.75	1.40	1.32	**1.00**
AR	cost	1132.6	875.5	1179.8	1809.7	567.5	**156.5**
	$\text{err}_{OG}(\%)$	6.88	5.38	7.00	11.06	3.13	**0.63**
	$\text{err}_{GO}(\%)$	38.33	**18.33**	70.95	47.14	80.00	70.71
	err(%)	13.81	**8.12**	20.50	18.61	19.36	15.25
YaleB	cost	1131.0	872.5	1821.2	727.2	255.3	**63.6**
	$\text{err}_{OG}(\%)$	18.67	14.50	29.80	12.00	3.33	**0.50**
	$\text{err}_{GO}(\%)$	13.00	**3.00**	37.75	8.50	68.50	42.00
	err(%)	17.50	**10.00**	35.00	11.30	30.40	17.10
ORL	cost	1525.2	1202.1	575.4	1533.5	189.0	**113.6**
	$\text{err}_{OG}(\%)$	21.57	17.14	7.86	21.86	2.14	**1.29**
	$\text{err}_{GO}(\%)$	17.75	**0.82**	31.75	4.25	48.75	29.50
	err(%)	22.00	12.27	16.55	15.64	19.09	**11.55**



(a) AR (b) FERET (c) YaleB (d) ORL

Figure 1: Comparing the methods with different number of 'in-group' subjects

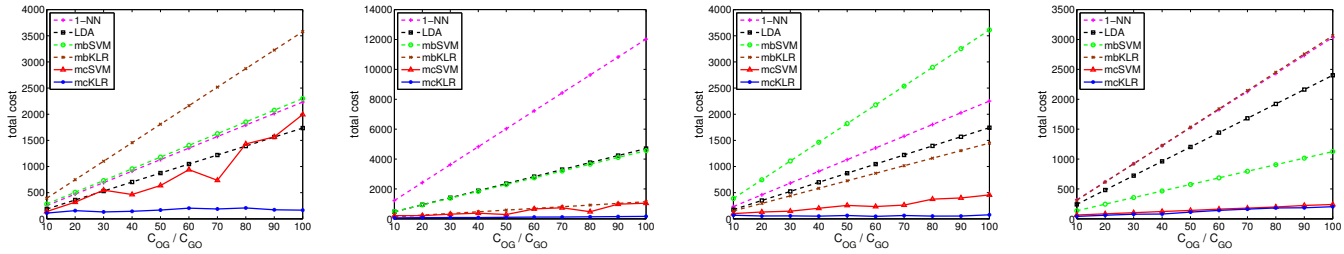


(a) AR (b) FERET (c) YaleB (d) ORL

Figure 2: Comparing the methods with different number of 'out-group' subjects unobserved in training set

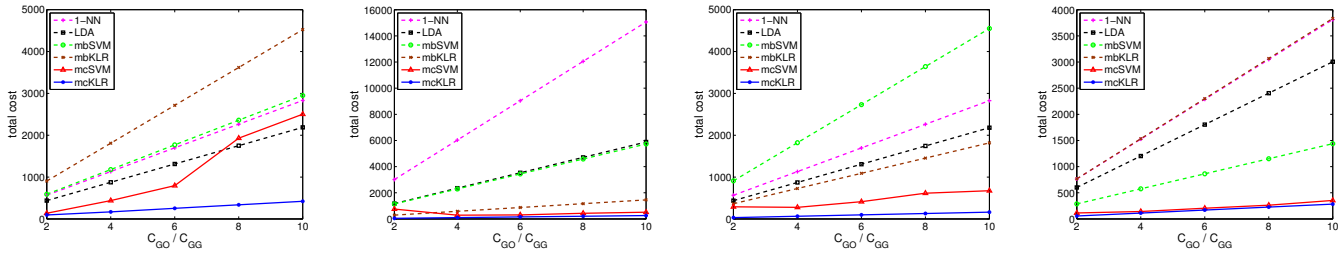
It can be found that on all databases and under all the N values the performance of mcKLR is always the best.

We also study the influence of the number of unobserved 'out-group' subjects on the performance of the compared methods. Here we fix M_{train} but vary $M - M_{train}$. For AR, $M - M_{train}$ varies from 0 to 50, with M_{train}=30; for FERET, $M - M_{train}$ varies from 50 to 550 with M_{train}=50; for YaleB, $M - M_{train}$ varies from 0 to 20 with M_{train}=10; for ORL, $M - M_{train}$ varies from 0 to 25 with M_{train}=10. The results are presented in Figure 2. Again, it can be found



(a) AR (b) FERET (c) YaleB (d) ORL

Figure 3: Comparing the methods under different C_{OG}/C_{GO}



(a) AR (b) FERET (c) YaleB (d) ORL

Figure 4: Comparing the methods under different C_{GO}/C_{GG}

that the performance of cost-sensitive methods are better than that of cost-blind methods. In particular, as the number of unobserved 'out-group' subjects increases, the gap between the performance of cost-sensitive and cost-blind methods tends to increase. The performance of mcKLR is always the best on all databases and with all number 'out-group' subjects. On FERET and ORL the performance of mcSVM is quite good, but still slightly worse than that of mcKLR. Moreover, in the task studied in this paper, the errors which often occur are the misclassifications of 'out-group' subjects to 'in-group' subjects, especially when there are a lot of unobserved 'out-group' subjects. While by incorporating costs to prevent high-cost errors, the misclassifications of 'out-group' subjects to 'in-group' subjects can be significantly reduced, and therefore the total error rate may be able to decrease. This suggests that mcKLR is also a good choice even if we use accuracy as the performance measure as in traditional face recognition systems.

Note that the cost ratios reflect the desirable tradeoff between different kinds of errors. For example, if the user thought that one false positive is more serious than 49 false negatives, s/he could set C_{GO}:$C_{OG} = 1 : 50$; while if s/he thought that one false positive is just more serious than 19 false negatives, s/he could set C_{GO}:$C_{OG} = 1 : 20$. It is necessary to compare the methods under different cost ratios to see whether the methods can adapt to different scenarios well. Here, we split C_{GG}:C_{GO}:C_{OG} into 2 parts: C_{GO}/C_{GG} and C_{OG}/C_{GO}. C_{GG} is always set as 1. First we fix $C_{GO}/C_{GG} = 4$ and vary C_{OG}/C_{GO} from 10 to 100. Then we fix $C_{OG}/C_{GO} = 50$ and vary C_{GO}/C_{GG} from 2 to 10. The results are shown in Figure 3 and 4, respectively. For cost-blind methods, the total cost is linear to the cost ratio for the change of cost ratio has no effect on their predictions. Similarly to the previous experiment, the performance of cost-sensitive methods are better than that of cost-blind methods and mcKLR is always the best.

Overall, the above experiments show that mcKLR achieves the best performance on all databases, under all number of 'in-group' subjects, all number of unobserved 'out-group' subjects, all cost ratios. It is clear that from the view of recognition result, mcKLR is the best choice among the compared methods.

It has been proved in [9] that when the optimal solution is obtained, the objective function of mcSVM is equivalent to Bayes decision rule with unequal costs. Since the objective function of mcKLR was directly derived from Bayes decision rule with unequal costs, it is not strange that mcKLR can outperform mcSVM. Actually, we have also compared mcKLR and mcSVM on problems other than face recognition, such as on UCI data sets, and also found that mcKLR outperforms mcSVM. Those results will be reported in a longer version of the paper.

We also compare the computational costs of the cost-sensitive methods mcSVM and mcKLR. We record the average training and test time costs in Table 4. The experiments are conducted on a PC with CPU 2.66GHz($\times$64) and

Table 4: Comparing the training/test time costs (in seconds)

		mcSVM	*mcKLR*
FERET	train	10.58	0.252
	test	1.07	1.13
AR	train	1.35	0.107
	test	0.0889	0.0897
YaleB	train	0.993	0.0841
	test	0.0295	0.0308
ORL	train	0.666	0.0408
	test	0.0316	0.0337

2G memory. We can see that the test time cost of mcKLR is almost as same as that of mcSVM, but its training time cost is much smaller than that of mcSVM, especially on the larger databases FERET and AR.

6. Conclusion

To the best of our knowledge, this paper presents the first study on cost-sensitive face recognition. We formulate this task as a multi-class cost-sensitive learning task, and propose the mcKLR method to solve this problem. Our experiments show that on all experimental settings the performance of mcKLR is always better than cost-blind methods and another cost-sensitive method, mcSVM. Moreover, the efficiency of mcKLR is better than mcSVM.

In this paper we assume that the cost matrix was given by user. Refining the cost matrix given by user or learning a cost matrix from the data automatically are interesting future issues. Studying cost-sensitive face recognition with serious class-imbalance problem is another interesting future work. In this paper we simply apply PCA to the face images before classification. To get a better performance, more careful preprocessing may be helpful. Applying our method to face images with occlusion or other complicated situations is also an interesting future issue.

Acknowledgements

The authors would like to thank the anonymous reviewers for helpful suggestions, Ji Zhu for guidance on implementing KLR, and Xu-Ying Liu, Min-Ling Zhang and Jun-Ming Xu for comments on a preliminary draft.

References

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html. 5

[2] 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*, pages 3–11, Seattle, WA, 2004. 1

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(4):711–720, 1997. 1

[4] 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*, pages 155–164, San Diego, CA, 1999. 1, 3

[5] 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. 3

[6] C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 973–978, Seattle, WA, 2001. 1, 3

[7] 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*, pages 688–694, Vancouver, Canada, 2001. 3

[8] K. C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005. 5

[9] 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 American Statistical Association*, 99(465):67–81, 2004. 1, 3, 7

[10] 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*, pages 374–380, Washington, DC, 2004. 3

[11] 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*, pages 970–974, Hong Kong, China, 2006. 5

[12] A. M. Martinez and R. Benavente. The AR face database. Technical Report 24, CVC, 1998. 5

[13] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997. 1

[14] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000. 5

[15] M. A. Turk and A. Pentland. Eigenface for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991. 1

[16] Z.-H. Zhou and X.-Y. Liu. On multi-class cost-sensitive learning. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 567–572, Boston, MA, 2006. 1, 3

[17] J. Zhu and T. Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3):427–443, 2004. 4