Neurocomputing 89 (2012) 89–95

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Incremental threshold learning for classifier selection

Yanwei Pang [a], Junping Deng [a], Yuan Yuan [b,*]

[a] School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China
[b] State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China

ARTICLE INFO

Article history:
Received 3 July 2011
Received in revised form
2 November 2011
Accepted 18 January 2012
Communicated by D. Zhang
Available online 28 February 2012

Keywords:
Incremental learning
Threshold-based classifier
Classifier fusion
Object detection
Pattern recognition

ABSTRACT

Threshold-based classifier is a simple yet powerful pattern classification tool, which has been frequently used in applications of object detection and recognition. A threshold-based classifier is usually associated with a unique one-dimensional feature. A properly selected threshold and a binary sign corresponding to the feature govern the classifier. However, the learning process is usually done in a batch manner. The batch algorithms are not suitable for sequentially incoming data because of the limitation of storage and prohibitive computation cost. To deal with sequentially incoming data, this paper proposes an incremental algorithm for incrementally learning the threshold-based classifiers. The proposed method can not only incrementally model the features but also estimate the threshold and training error in a close form. The effectiveness of the proposed algorithm is evaluated in the applications of gender recognition, face detection, and human detection.

1. Introduction

Both incremental learning and batch learning learn knowledge and establish model from data. But the batch learning requires that all the data are available before learning takes place, which makes it unsuitable for incrementally incoming data due to limitation of storage and computation cost. Contrary to batch learning, incremental learning learns knowledge sample by sample and thus does not require access to previously seen data [7,5]. Therefore, incremental learning has received a great deal of attention [7,5,21,1,2,41] in the fields of classification, dimensionality reduction, object tracking, object detection and recognition, etc. Incremental backpropagation neural network has been widely used for supervised classification [23] while the state-of-the-art self-organizing incremental neural network (SOINN) was designed to realize the unsupervised learning task [24]. Conformal prediction makes full use of past experience to determine precise levels of confidence in new prediction, which can be used with any method of point prediction for classification as well as regression [25]. Incremental singular value decomposition (SVD) was applied in computer vision and audio feature extraction [22]. As an efficient on-line dimensionality reduction method, candid covariance-free incremental principal component analysis (CIPCA) [26] has been applied for face recognition [26] and scene segmentation [27]. Sequential Monte Carlo method, also known as particle filter and as a famous incremental learning algorithm, is among the most successful object tracking methods [28].

In this paper we focus on incrementally learning of multiple threshold-based classifiers. Usually, a threshold-based classifier corresponds to a unique feature and the classifier is determined by thresholding the feature according a certain criterion. The criterion is often related to the training error. Threshold-based classifiers have been widely used for object detection and recognition. For example, Viola's face detection algorithm [16] uses Haar-like rectangle features and the difference value between two rectangles, together with a proper threshold and a sign, form a base classifier. If the difference is larger than the leaned threshold, the pattern in question is classified as nonface, and face otherwise. Yang et al. [6] proposed to employ the difference of Gabor features and compute the optimal threshold to recognize faces. In addition to Gabor features, local binary pattern (LBP) [13] features have also been widely used in threshold-based object recognition [12,14,15]. Rodriguez and Marcel proposed to adopt pixel-based classifiers for face verification [8]. The features in their method are simply raw pixel values. If the pixel value exceeds a learned threshold, a decision is made to claim that the face is either client or imposter.

The above threshold-based classifiers are all selected by the well-known Adaboost algorithm [9]. But Adaboost is a batch learning algorithm. At each iteration of the algorithm, all the data samples have to be re-assigned proper weights [16,17]. Therefore, these threshold-based multi-classifier selection algorithms are not suitable for sequentially incoming data. As an improved version of Adaboost, LEARN++ is towards incremental learning [10,7]. But the input mode of LEARN++ is subset by subset instead of sample by sample. By subset we mean that the input is a small database instead of a single sample. LEARN++ utilizes traditional Adaboost algorithm to train on the small database once available.

* Corresponding author.
E-mail addresses: yuany@opt.ac.cn, yuan369@hotmail.com (Y. Yuan).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2012.01.012

In this paper, we propose a truly incremental threshold-based classifier learning algorithm which has two distinct characteristics:

1. It is a sample-by-sample incremental learning algorithm. Once a sample comes, the algorithm updates a properly designed model of features and then updates a pre-defined number of threshold-based classifiers without re-training on previous data. All the knowledge of previous data is encoded in the model.
2. Both the threshold and training error of each classifier are optimally determined in a close form. Traditional methods choose the threshold by exhaustively searching in the range of the values of a feature. It starts from the minimum value and then increases the threshold by a predefined step. At each step the training error or weighted training error is recorded. The optimal threshold is the one with minimum error. This strategy is sensitive to the threshold step. On one hand, if the threshold step is too large, the optimal threshold can seldom be found. Hence it leads to a poor classifier. On the other hand, if the threshold step is too small, the selection process of the optimal threshold is too time-consuming. By contrast, the proposed algorithm can compute the optimal threshold at one time in an arbitrary accuracy.

The remainder of this paper is organized as follows: Section 2 gives a brief description of threshold-based classifier and two traditional batch algorithms. We present the proposed incremental algorithm in Section 3. The experimental results are reported in Section 4. Finally, conclusions are drawn in Section 5.

2. Batch algorithm

2.1. Threshold-based classifier

This paper is concerned with threshold-based classifier. In this subsection, we describe the form of the threshold-based classifier.

The threshold-based classifier $f_i(x_i;s_i,\theta)=s_i\times x_i-\theta_i$ is associated with a feature indexed by i. x_i is value of feature i [11]. The classifier has two parameters: s_i and θ_i with $s_i\in\{+1,-1\}$ being a sign and θ_i being a proper threshold. Without loss of generality, it is assumed to be a two-class problem where the two classes are named positive class and negative class. The estimated class label $l_{est}(x_i)$ is either $+1$ or -1 where '$+1$' stands for positive class and '-1' stands for negative class. The corresponding decision rule is given by

$$\begin{cases} l_{est}(x_i)=+1, & \text{if } f_i(x_i;s_i,\theta_i)=s_i\times x_i-\theta_i>0 \\ l_{est}(x_i)=-1, & \text{if } f_i(x_i;s_i,\theta_i)=s_i\times x_i-\theta_i\le 0 \end{cases}. \tag{1}$$

If $s_i=1$, the decision rule becomes

$$\begin{cases} l_{est}(x_i)=+1, & x_i>\theta_i \\ l_{est}(x_i)=-1, & x_i\le\theta_i \end{cases}. \tag{2}$$

If $s_i=-1$, the decision rule becomes

$$\begin{cases} l_{est}(x_i)=+1, & x_i<\theta_i \\ l_{est}(x_i)=-1, & x_i\ge\theta_i \end{cases}. \tag{3}$$

Because a single feature and the corresponding classifier are hard to result in a satisfying classification performance, it is usually beneficial to combine multiple features and classifiers. A widely used combination strategy is majority voting which can be formulated as

$$l_{est}(x_i)=\text{sign}\left[\sum_i \text{sign}\{f_i(x_i;s_i,\theta_i)\}\right], \tag{4}$$

where $\text{sign}(\cdot)$ is the sign function. Successful combination of multiple classifiers relies on the premise that the classifiers to be combined are diverse and can complement each other.

2.2. Straightforward batch algorithm

Both s_i and θ_i are learned from the training data. In this subsection, we briefly describe how to determine the two parameters in a widely used straightforward style.

The idea of the algorithm is as follows. Given the entire data, the criterion for computing the free parameters is according to the minimum training error rate. The range of feature i is divided into a lot of levels. Each level is considered as a candidate threshold. For each candidate threshold, the training error rate is recorded. The optimal threshold is the one with the minimum training error rate.

Specifically, suppose that the free parameter θ_i is in the range $[\theta_{\min},\theta_{\max}]$ and the pre-defined (fixed) step is θ_{step}. First, the sign s_i is set to 1. With this configuration, find the best $\theta_1(i)\in[\theta_{\min},\theta_{\max}]$ that gives rise to the minimum training error rate $e_1(i)$. Then the sign s_i is set to -1, and also find the best $\theta_{-1}(i)\in[\theta_{\min},\theta_{\max}]$ that gives rise to the minimum training error rate e_{-1i}. The optimal sign and threshold are those with the minimum error rate $e(i)=\min\{e_1(i),e_{-1}(i)\}$. The procedure is shown in Algorithm 1.

Algorithm 1. Straightforward batch learning algorithm.

Input: Training samples $\mathbf{y}_j$, $j=1,\dots,N$. $K=\theta_{\max}/\theta_{step}$. M: the number of entire features.
Output: The first m optimal classifiers where $m\le M$.
1. **for** each feature i, $i=1,\dots,M$
2. Let $s_i=1$
3. **for** $\theta_1^k=\theta_{\min}+k\times\theta_{step}$ to $\theta_{\max}$, $k=1,\dots,K$
4. **for** each training sample y_j, $j=1,\dots,N$
5. Extract the ith feature x_{ij} of y_j
6. Estimate the class label $l_{est}=sign(x-\theta_1^k)$
7. Count the number of misclassified samples: $e_{1k}(i)\leftarrow e_{1k}(i)+I(l_{est}\ne l_{true}(\mathbf{y}_j))$
8. **end for**
9. **end for**
10. $k^*=\arg\min_k\{e_{1k}(i)|k=1,\dots,K\}$,
11. $e_1(i)=e_{1k^*}(i)$, $\theta_{1i}=\theta_1^{k^*}$,
12. Let $s_i=-1$, repeat line 3 to line 11 except replacing θ_1^k with θ_{-1}^k, $e_1(i)$ with $e_{-1}(i)$, and '$x-\theta_1^k$' with 'θ_{-1}^k-x'
13. **If** $e_1(i)<e_{-1}(i)$ **then** $s_i\leftarrow 1$ and $\theta_i=\theta_{1i}$, **else**
14. $s_i\leftarrow -1$ and $\theta_i=\theta_{-1i}$
15. **end if**
16. $e(i)=\min\{e_1(i),e_{-1}(i)\}$
17. Sort the classifiers in ascending order w.r.t. to $e(i)$
18. Reserve the first m classifiers.
19. **end for**

In Algorithm 1, I('*statement*')$=1$ if the '*statement*' is true, I('*statement*')$=0$ otherwise. $l_{true}(\mathbf{y})$ is the true (underlying) class label of $\mathbf{y}$.

However, the simple algorithm is not suitable for sequential data because it has to compute the training error which cannot be obtained without requiring access to all the training samples.

2.3. Adaboost-based algorithm

The Adaboost algorithm is also a batch learning algorithm. It is powerful to select (learn) and combine weak classifiers into a stronger one [16,9].

The Adaboost algorithm is distinguished from the above straightforward batch algorithm in the three aspects. Firstly, each component classifier is selected according to weighted training error rate rather than usual training error. Secondly, each sample is assigned a weight which is large if it is difficult to be correctly classified by the existing classifier. The weights need to be normalized so that they can be regarded as a distribution. Thirdly, the classifiers are combined in a more elegant manner. The less the weighted training error is, the more the classifier contribute.

However, the traditional Adaboost algorithm is not suitable for incremental learning. It has to maintain a distribution of all the samples which reflects the difficulty for the currently selected classifiers to correctly recognize the samples. The parameters, s and θ, of a classifier are determined by counting the classification error on entire samples. Once a new sample is available, the training process has to be repeated and the previous training results cannot be reused. It also should be noted that the computation of the threshold s and sign θ is similar to that of the straightforward algorithm. They are both determined by trial-error manner.

Recently, online Adaboost algorithms have been developed [32,33,36–40]. Javed proposed an online version of Adaboost algorithms [36]. However, it needs to train a strong classifier offline and then the weak classifiers can be updated online. Grabner et al. developed a true incremental Adaboost algorithm. It does apply online boosting not directly to the weak classifiers but to feature selectors [32,33].

3. Proposed incremental algorithm

Both the aforementioned two algorithms belong to batch learning. Once a new data sample comes, they have to be rerun (i.e. re-train). Thus, previously learned knowledge is forgotten (discarded). This is contrasted with the proposed incremental algorithm where the models and classifiers are incrementally tuned when a novel sample is available. The proposed algorithm models features in a parametric form and the threshold parameter can be determined annalistically.

3.1. Statistical modeling of the features

The threshold-based classifier $f_i(x_i;s,\theta)$ works in many applications because of the discriminating ability of the one-dimensional feature x_i. The fact that a threshold can be used to separate two classes implies that the overlap between distributions of the features of positive class ω_1 and negative class ω_2, is relatively small. Therefore, it is reasonable to model the features of different classes with different Gaussian distributions [34]. In addition, many successful object detection algorithms (e.g. [35]) on the assumption of Gaussian distribution also give the evidence of the reasonability and feasibility of the Gaussian distribution. The distribution of the features is determined by only two parameters: mean and variance. Because the dimensionality of the feature is one, variance instead of covariance is used together with mean to describe the uni-variable Gaussian distribution. The mean and variance are given by

$$u_i = \frac{1}{N}\sum_{x \in \omega_i} x, \tag{5}$$

$$\sigma_i = \frac{1}{N}\sum_{x \in \omega_i} (x-u_i)^2. \tag{6}$$

The class-conditional probability is

$$p(x|\omega_i) = \frac{1}{\sigma_i\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{x-u_i}{\sigma_i}\right)^2\right). \tag{7}$$

The minimum-error-rate classification can be achieved using the discriminant functions [18]

$$\begin{aligned} g_i(x) &= \ln[p(\omega_i|x)] = \ln p(x|\omega_i) + \ln P(\omega_i) \\ &= -\frac{1}{2}\left(\frac{x-u_i}{\sigma_i}\right)^2 - \frac{1}{2}\ln 2\pi - \frac{1}{2}\ln\sigma_i^2 + \ln P(\omega_i). \end{aligned} \tag{8}$$

It is assumed in this paper that the prior probability $P(\omega_i)$ is uniform. Note that the assumption of uniform prior probability may not strictly hold, but the method depending on this assumption does work in practice. The boundary between the two distributions is formed by the point that follows the equation:

$$g(x) = g_1(x) - g_2(x) = 0. \tag{9}$$

Substituting (8) into (9) yields

$$g(x) = (\sigma_1^2-\sigma_2^2)x^2 + (2\sigma_2^2u_1 - 2\sigma_1^2)x + \sigma_1^2u_2^2 - \sigma_2^2u_1^2 + \sigma_1^2\sigma_2^2\ln\left(\frac{\sigma_2^2}{\sigma_1^2}\right) = 0. \tag{10}$$

If $\sigma_1 = \sigma_2$, then the threshold is given by

$$\theta = x = \frac{(u_1+u_2)}{2}. \tag{11}$$

If $\sigma_1 \neq \sigma_2$, then solution is

$$\theta = \frac{-(\sigma_2^2u_1 - \sigma_1^2u_2) \pm \sqrt{(\sigma_2^2u_1-\sigma_1^2u_2)^2 - (\sigma_1^2-\sigma_2^2)(\sigma_1^2u_2^2 - \sigma_2^2u_1^2 + \sigma_1^2\sigma_2^2\ln(\sigma_2^2/\sigma_1^2))}}{(\sigma_1^2-\sigma_2^2)} \tag{12}$$

The symbol "$\pm$" in (12) makes θ possible to have two different values. Only if the squared term is zero, θ has a unique value. Fig. 1(a) illustrates the situation. But when this squared term is nonzero, then θ has two different values: θ_1 and θ_2. In this case, the threshold value that is assigned larger class-conditional probability $p(|\omega_i)$ is selected while the other is discarded. As illustrated in Fig. 1(b), θ_2 is selected while θ_1 is discarded.

The sign parameter s is easy to be determined by simply comparing the values of u_1 and u_2:

$$s = \begin{cases} 1 & u_1 < u_2 \\ -1 & \text{otherwise} \end{cases}. \tag{13}$$

Eqs. (11) and (12) show that the threshold in our algorithm is determined optimally in a close form. This is contrasted with

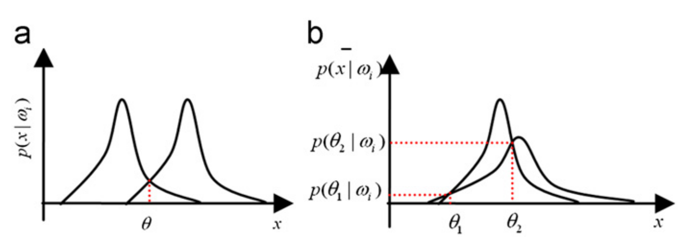


Fig. 1. Eq. (12) has a unique threshold value in (a) and two different solutions in (b). In (b), because θ_2 is assigned the larger class-conditional probability, θ_2 is selected.

Algorithm 1 and Adaboost algorithm where the thresholds are determined by exhaustively searching in the range of the values of a feature with a predefined step. These algorithms are much sensitive to the threshold step and time-consuming if the threshold is small. The proposed incremental algorithm does not involve the threshold step and can obtain the optimal threshold at one time in arbitrary accuracy.

3.2. Incremental modeling

It is observed from Eqs. (11) and (12) in the above subsection that threshold θ is a function of means u_1 and u_2 and variances σ_1 and σ_2. However, observing (5) and (6), one can find that u_i and σ_i are computed only when all the training samples are available. To make it tailored to sequentially incoming samples, we propose to incrementally compute u and σ (for the sake of notational simplicity, hereinafter the subscript i is omitted):

$$u(n)=(1-\eta)u(n-1)+\eta x(n), \tag{14}$$

$$\sigma^2(n)=(1-\eta)\sigma^2(n-1)+\eta(x(n)-u(t))^2, \tag{15}$$

where $0\le\eta<1$ is a small learning rate, $u(0)=x(0)$, and $\sigma^2(0)=0$.

Eqs. (14) and (15) are frequently used in background modeling [19]. It is easy to prove that (14) and (15) approach (5) and (6), respectively, when the number of samples is infinite and the learning rate approaches zero. Similarly, it can be proved that (15) approaches (6) as the number of data samples is infinite and the learning rate is zero.

Given a feature, Eqs. (11) and (12) show that the threshold can be determined by the mean and variance while Eqs. (14) and (15) show that the mean and variance can be incrementally undated. Because there are a large number of features, it is required to select a few of good features. The smaller the training error, the better the corresponding feature. To incrementally compute the training error $P(error)$, we adopt Bhattacharyya bound (a special case of the Chernoff bound) to approximate the training error. The Chernoff bound is given as follows [18]:

$$P(error)\le P^{\beta}(\omega_1)P^{1-\beta}(\omega_2)\int p^{\beta}(\mathbf{x}|\omega_1)p^{1-\beta}(\mathbf{x}|\omega_2)d\mathbf{x}$$
$$=P^{\beta}(\omega_1)P^{1-\beta}(\omega_2)\exp^{-k(\beta)}, \tag{16}$$

where

$$k(\beta)=\frac{\beta(1-\beta)}{2}\frac{(u_+-u_-)^2}{\beta\sigma_++(1-\beta)\sigma_-}+\frac{1}{2}\ln\frac{\beta\sigma_++(1-\beta)\sigma_-}{2\sqrt{\sigma_+^{\beta}\sigma_-^{1-\beta}}}. \tag{17}$$

Assume that the prior probabilities of positive (marked by '+') and negative classes (marked by '−') are equal, the error bound e (Bhattacharyya bound) for the classifier is given by

$$e=\exp(-k(1/2)), \tag{18}$$

$$k(1/2)=\frac{1}{4}\frac{(u_+-u_-)^2}{\sigma_++\sigma_-}+\frac{1}{2}\ln\frac{\sigma_++\sigma_-}{2\sqrt{\sigma_+\sigma_-}}, \tag{19}$$

where (u_+, σ_+) and (u_-, σ_-) are the mean and squared variance of the positive class (ω_1) and negative classes (ω_2), respectively. From Eqs. (18) and (19), one can see that the error bound is completely governed by four mean-variance parameters: u_+, σ_+, u_-, and σ_-. Because the four parameters can be obtained in an incremental manner, the error bound can also be computed incrementally. The error bound can be computed quickly without having to really classify the training samples by the threshold-based classifier.

The threshold-based classifiers are sorted in ascending order according to their error bounds. Finally, the first m classifiers are selected.

The procedure of the proposed algorithm is described in Algorithm 2.

Algorithm 2. The proposed incremental algorithm.

Input: Sequentially coming training samples $\mathbf{y}_j$, $j=1,\ldots,N$. M: The number of entire features. η: the learning rate.
Output: The first m optimal classifiers: $f_j(x_j,\theta_j,s_j)$, $j=1,\ldots,m$ where $m\le M$

1. **for** each available sample y_i and its class label $l(\mathbf{y}_i)$, $i=1,\ldots,N$.
2. Extract the feature value x_{ij} of b, $j=1,\ldots,M$.
3. **if** $l(\mathbf{y}_i)=1$ **then**
4. $u_+(n)=(1-\eta)u_+(n-1)+\eta x(n)$,
5. $\sigma_+^2(n)=(1-\eta)\sigma_+^2(n-1)+\eta(x(n)-u_+(t))^T(x(n)-u_+(n))$.
6. **if** $l(\mathbf{y}_i)=-1$ then
7. $u_-(n)=(1-\eta)u_-(n-1)+\eta x(n)$,
8. $\sigma_-^2(n)=(1-\eta)\sigma_-^2(n-1)+\eta(x(n)-u_-(t))^T(x(n)-u_-(n))$
9. **if** $\sigma+=\sigma-$, then the threshold is given by
10. $\theta=x=\frac{(u_1+u_2)}{2}$.
11. **if** $\sigma1\ne\sigma2$, then the threshold is
$$\theta=\frac{-(\sigma_2u_1-\sigma_1u_2)\pm\sqrt{(\sigma_2u_1-\sigma_1u_2)^2-(\sigma_1-\sigma_2)(\sigma_1u_2^2-\sigma_2u_1^2+\sigma_1\sigma_2\ln(\sigma_2/\sigma_1))}}{(\sigma_1-\sigma_2)}$$
12. Estimate the error: $e_j=\exp(-k(1/2))$.
13. Sort error e_j in ascending order and return the first m classifiers $f_j(x_j,\theta_j,s_j)$ with $e_1<e_2<\cdots<e_M$.
14. **end for**

The sorting process can be regarded as the processes of both classifier selection and feature selection. Obviously, the classifier selection is accompanied with feature selection in the threshold-based classifier learning algorithm. Compared with traditional methods, the proposed methods can incrementally compute the threshold and error bound once a new sample is available.

4. Experimental Results

The proposed method has been applied and evaluated in gender recognition, face detection, and human detection which are all two-class pattern classification problems. For gender recognition and face detection, the basic features extracted from images are local binary pattern (LBP) which has been widely used in visual pattern classification and has achieved great success [12–15] in these applications. For human detection, we employed Histogram of Oriented Gradients (HOG) [29,30].

The process of computing basic LBP feature is illustrated in Fig. 2. Fig. 2(a) shows the pixel intensities under a 3×3 mask. The intensity in the center is used as a threshold. All the pixels except the central one are binarized by the threshold. If a pixel intensity is larger than the threshold, it set to 1, and 0 otherwise. The threshold result is shown in Fig. 2(b). Finally, in Fig. 2(c), the result is encoded by a binary string as well as the corresponding decimal value.

In our experiments, uniform LBP was employed. It is an important extension of original LBP. Uniform LBP is the binary pattern that contains at most two bitwise transitions from 1 to 0 or vice versa when the bit pattern is considered circular [12,13].

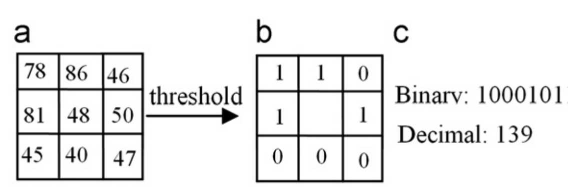


Fig. 2. Process of computing basic LBP feature. (a) A 3×3 mask. The entries are pixel values. The value of central pixel is used as a threshold. All the other 8 pixels are binarized by the threshold. The threshold result is shown in (b). (c) The result is encoded by a binary string and decimal value.

Uniform LBP is capable of describing texture microstructures such as edges. Refer to [12,13] for details.

HOG is a dense version of SIFT [30,31]. Refer to [29,30] for details.

4.1. Gender recognition

Gender recognition based on the face images has received a fair amount of attention in the computer vision literature. In our experiments, a subset of the FERET face data set [20] was used to evaluate the gender recognition performance of the proposed method. In the subset, there are 657 images with 380 male and 277 female face images. Among them, 461 images were used for training while the remaining 196 images for testing. We randomly sampled the data in five times and thus we obtain five different training and testing sets. The images were normalized to 48×48 pixels. The recognition performance is measured by error rate which is defined as the number of incorrectly classified images divided by the number of total images.

Fig. 3 shows the error rates with respect to classifier (feature) number. It is observed that the error rates of the proposed algorithm approach those of the batch algorithm (Section 2.2) to a great extent. The recognition performance arrives at the best point when about 50 classifiers are selected.

To investigate the incremental property of the proposed method, we let the number of selected classifiers/features fixed to 50 and show how the error rates of the proposed algorithm vary with increasing of the number of training samples. The results are shown in Fig. 4. As can be seen in the figure, the error rate decreases as more samples are available.

The third column of Table 1 shows the training time and storage required by the proposed algorithm and the batch one. The algorithms were run on a 2.0 GHz CPU and 2.0 GB RAM computer with the Matlab platform. It is clear that the proposed algorithm is 15 times faster than the traditional algorithm and the storage cost is 42.5 times economical than the traditional algorithm.

4.2. Face detection

Face detection can be regarded as a face against non-face classification problem. We use the MITEx face database to evaluate our threshold-based face detector. The training set consists of 800 face images and 1100 non-face images while the testing set consists of 897 faces and 1276 nonfaces. Both of the face images and the non-face images are of size 20×20.

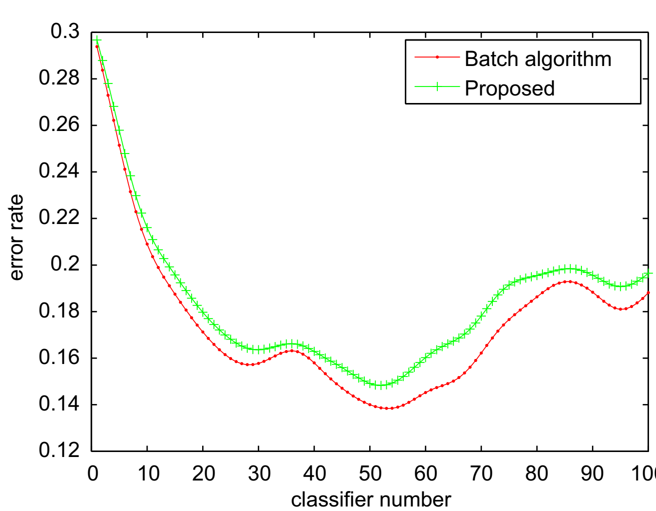


Fig. 3. Error rates of gender recognition on the FERET face database.

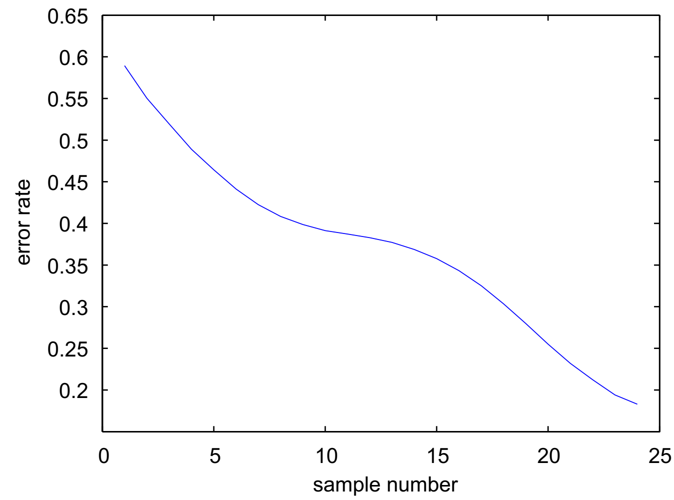


Fig. 4. Incremental performance on the FERET database: the classification error rates of the proposed method as the training samples come sequentially (i.e. sample by sample).

Table 1
Comparison of the computational cost of the proposed algorithm and the batch one.

Algorithms	Gender recognition	Face detection	Human detection
Training time (s)			
Proposed algorithm	12	55	376
Batch algorithm	180	900	4500
Storage (KByte)			
Proposed algorithm	20	3.5	35
Batch algorithm	850	583	4368

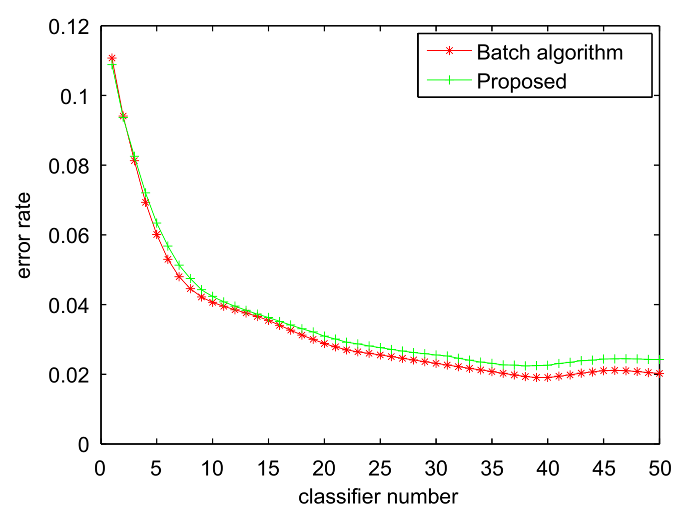


Fig. 5. Error rates of face detection on the MITEx face database.

The detection error rates are shown in Fig. 5 from which one can find that in most of regions the detection error rates decrease as the number of classifiers increases. The proposed incremental algorithm performs almost as good as the traditional batch algorithm.

Fig. 6 shows that the detection performance improves as the samples come sequentially. Note that 33 classifiers/features are used in Fig. 6. Roughly speaking, the detection performance is consistently enhanced as more samples are incrementally available.

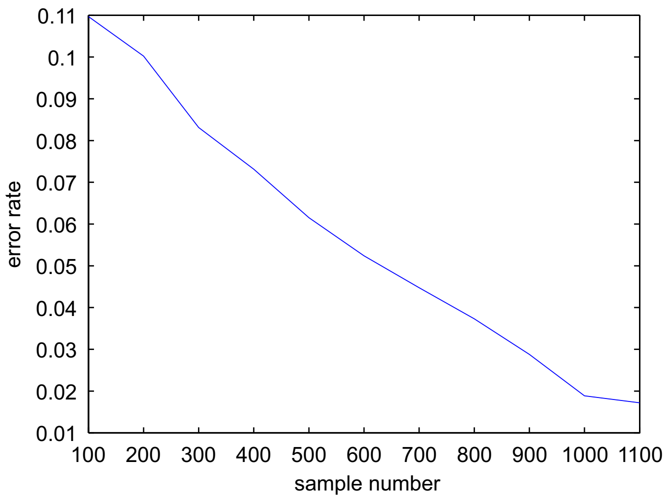


Fig. 6. Incremental performance on the MITEx face database: the detection error rates of the proposed method drop as the samples come sequentially.

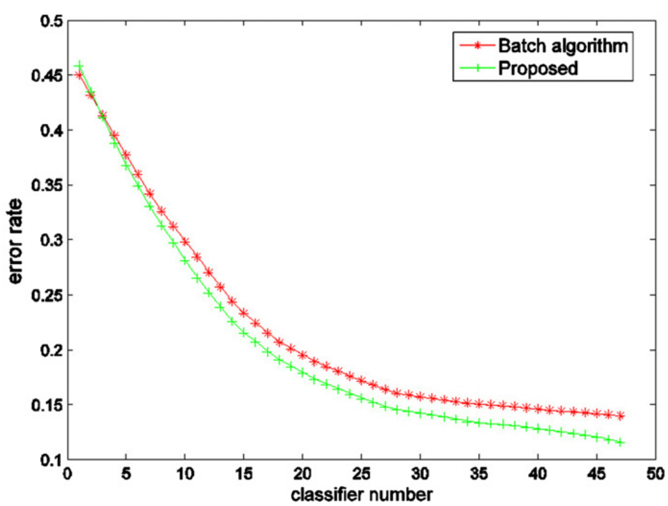


Fig. 7. Error rates of human detection on the INRIA human database.

The forth column of Table 1 gives the training time and storage required by the proposed algorithm and the batch one. For such face detection task, the proposed incremental algorithm is 16.36 times faster than the traditional algorithm and the storage cost is 166 times economical than the traditional algorithm.

4.3. Human detection

The INRIA data set was used to evaluate the propose method in application of human detection. Human detection is also a two-class problem where the positive class is human and the negative class is nonhuman. The positive training set has 1208 images where the negative one has 12,180 images [29,30]. The testing data set includes 288 positive images and 453 negative ones. The 3780-dimensional HOG feature vectors are extracted from the images. The description of HOG features can be found from [29,30]. The training images are of size 64×128. Fig. 7 shows the human detection performance. Though the incremental algorithm is not better than the batch one, their difference is small.

From Fig. 8 we can see that the error rate of human detection tends to decrease as the training images are sequentially available. The knowledge conveyed by previous training images is reserved but such training images are not necessary to be

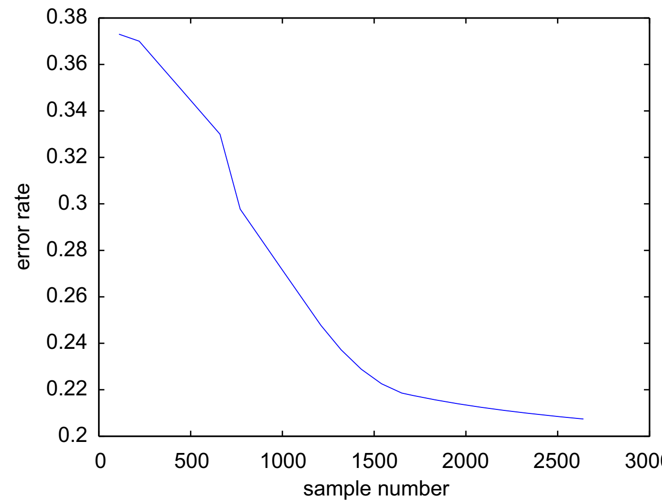


Fig. 8. Incremental performance on the INRIA human database: the detection error rates of the proposed method drop as the samples come sequentially.

remained. Once new images are available, previous knowledge is updated by the knowledge encoded in the new images.

The training time and storage required by the proposed algorithm and the batch one for human detection application are given in the fifth column of Table 1. From the table, it is observed that the proposed incremental algorithm is 11.96 times faster than the traditional algorithm and the storage cost is 124.8 times economical than the traditional algorithm.

The experimental results on the gender, face, and human databases clearly demonstrate the effectiveness and efficiency of the proposed algorithm. The proposed method is more suitable for real-time and large-scale applications.

5. Conclusions

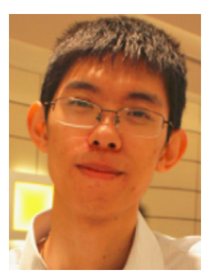
In this paper, we have presented an incremental threshold-based classifier learning algorithm. A threshold-based classifier is uniquely associated to an one-dimensional feature. The one-dimensional feature is modeled by the univariate Gaussian distribution. The parameters of the distributions of positive class and negative class are incrementally updated. These parameters directly determine the threshold-based classifier. The training error is analytically determined by the parameters and so it can be computed quickly without having to really classify the training samples. The algorithm is designed for two-class problem. In our ongoing research, we generalize our algorithm to multi-class (multi-label) task [3,4].

References

[1] Annalisa Franco, Dario Maio, Davide Maltoni, Incremental template updating for face recognition in home environments, Pattern Recognition 43 (8) (2010) 2891–2903.
[2] Jyrko Correa-Morris, Dustin L. Espinosa-Isidrón, Denis R. Álvarez-Nadiozhin, An incremental nested partition method for data clustering, Pattern Recognition 43 (7) (2010) 2439–2455.
[3] Tingting Mu, Asoke K. Nandi, Multiclass classification based on extended support vector data description, IEEE Trans. Syst. Man Cybern.—Part B: Cybern. 39 (5) (2009) 1206–1216.
[4] Jinghui Tang, Xian-shen Hua, Meng Wang, Zhiwei Gu, Guo-jun Qi, Xiuqing Wu, Correlative linear neighborhood propagation for video annotation, IEEE Trans. Syst. Man Cybern.—Part B: Cybern. 39 (2) (2009) 409–416.
[5] Haitao Zhao, Pong Chi Yuen, Incremental linear discriminant analysis for face recognition, IEEE Trans. Syst. Man Cybern.—Part B: Cybern. l38 (1) (2008) 210–221.
[6] Peng Yang, Shiguang Shan, Wen Gao, Stan Z. Li, Dong Zhang, Face recognition using ada-boosted gabor features, in: Proceedings of the 6th International

Conference on Automatic Face and Gesture Recognition, Seoul, Korea, May 17–19, 2004.
[7] Davi Parikh, Robi Polikar, An ensemble-based incremental learning approach to data fusion, IEEE Trans. Syst. Man Cybern.—Part B: Cybern. 37 (2) (2007) 437–450.
[8] Y. Rodriguez, S. Marcel, Boosting pixel-based classifier for face verification, in: Proceedings of the Biometric Authentication Workshop of the 8th European Conference on Computer Vision, Prague, May 11–14, 2004.
[9] Yoav Freund, Robert E. Schapire, A decision-theoretic generalization of online learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[10] Robi Polikar, Lalita Udpa, Satish S. Udpa, Vasant Honavar, Learn++: an incremental learning algorithm for supervised neural networks, IEEE Trans. Syst. Man Cybern.—Part C: Appl. Rev. 31 (4) (2001) 497–508.
[11] Chris Mesterharm, Using linear-threshold algorithms to combine multi-class sub-experts, in: Proceedings of International Conference on Machine Learning, Washington, USA, August 21–24, 2003.
[12] Timo Ahonen, Abdenour Hadid, Matti Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2037–2041.
[13] Timo Ojala, Matti Pietikainen, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[14] Guangcheng Zhang, Xiangsheng Huang, San Z. Li, Yangsheng Wang, Xihong Wu, Boosting local binary pattern (LBP)-based face recognition, Lect. Notes Comput. Sci. 3338 (2004) 179–186.
[15] Caifeng Shan, Tommaso Gritti, Learning discriminative LBP-histogram bins for facial expression recognition, in: Proceedings of the British Machine Vision Conference, Leeds, UK, September 1–4, 2008.
[16] Paul Viola, Michael J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[17] Jie Chen, Shiguang Shan, Shengye Yan, Xilin Chen, Wen Gao, Modification of the adaboost-based detector for partially occluded faces track, in: Proceedings of the International Conference on Pattern Recognition, Hong Kong, China, August 20–24, 2006.
[18] Richard O. Duda, Peter E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[19] C. Stauffer, W.E.L. Grimson, Adaptive background mixture models, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, Ft. Collins, CO, USA, June 23–25, 1999.
[20] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rivzi, Patrick J. Rauss, The FERET evaluation methodology for face-recognition algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 22 (10) (2000) 1090–1104.
[21] Jimeng Sun, Dacheng Tao, Spiros Papadimitriou, Philip S. Yu, Christos Faloutsos, Incremental tensor analysis: theory and applications, ACM Trans. Knowl. Discovery Data 2 (3) (2008).
[22] M.E. Brand, Incremental singular value decomposition of uncertain data with missing values, in: Proceedings of the European Conference on Computer Vision, Comenhagen, Denmark, May 28–31, 2002.
[23] Keem Siah Yap, Chee Peng Lim, Izham Abidin, A hybrid ART-GRNN online learning neural network with a ε-insensitive loss function, IEEE Trans. Neural Networks 19 (9) (2008) 1641–1646.
[24] Shen Furao, Tomotaka Ogura, Osamu Hasegawa, An enhanced self-organizing incremental neural network for online unsupervised learning, Neural Networks 20 (2007) 893–903.
[25] Glenn Shafer, Vladimir Vovk, A tutorial on conformal prediction, J. Mach. Learn. 9 (2008) 371–421.
[26] Juyang Weng, Yilu Zhang, Wey-Shiuan Hwang, Candid covariance-free incremental principal component analysis, IEEE Trans. Pattern Anal. Mach. Intell. 25 (8) (2003) 1034–1040.
[27] Yanwei Pang, Yuan Yuan, Scene segmentation based on IPCA for visual surveillance, Neurocomputing 72 (2009) 2450–2454.
[28] M. Isard, A. Blake, Condensation—conditional density propagation for visual tracking, Int. J. Comput. Vis. 29 (1) (1998) 5–28.
[29] Yanwei Pang, Yuan Yuan, Xuelong Li, Jing Pan, Efficient HOG human detection, Signal Process. 91 (2011) 773–781.
[30] Navneet Dalal, Bill Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 886–893.
[31] D.G. Lowe, Distinctive image features for scale-invariant key points, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[32] H. Grabner, H. Bischof, On-line boosting and vision, in: Proceedings of the CVPR, vol. 1, 2006, pp. 260–267.
[33] Helmut Grabner, Michael Grabner, Horst Bischof, Real-time tracking via online boosting, BMVC, 2006.
[34] Feiping Nie, Shiming Xiang, Yangqing Jia,Changshui Zhang, Shuicheng Yan, Trace ratio criterion for feature selection, in: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI), Chicago, 2008.
[35] Baback Moghaddam, Alex Pentland, Probabilistic visual learning for object representation, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 696–710.
[36] O. Javed, S. Ali, M. Shah, Online detection and classification of moving objects using progressively improving detectors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 695–700.
[37] Chang Huang, Haizhou Ai, Takayoshi Yamashita, Shihong Lao, Masato Kawade, Incremental learning of boosted face detector, in: Proceedings of the IEEE Conference on Computer Vision, 2007.
[38] Rong Liu, Jian Cheng, Hanqing Lu A robust boosting tracker with minimum error bound in a co-training framework, in: Proceedings of the IEEE Conference on Computer Vision, 2009.
[39] Takuya Kidera, S. Ozawa, S. Abe, An incremental learning algorithm of ensemble classifier systems, in: Proceedings of the IEEE International Joint Conference on Neural Network, 2006.
[40] Andreas Opelt, Axel Pinz, Andrew Zisserman, Incremental learning of object detectors using a visual shape alphabet, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 3–10.
[41] Hsin-Hsiung, Yi-Ren Yeh, An iterative algorithm for robust kernel principal component analysis, Neurcomputing 74 (18) (2011) 3921–3930.

Yanwei Pang (M'07–SM'09) received his Ph.D. degree in Electronic Engineering from the University of Science and Technology of China (USTC) in 2004. Currently, he is an Associate Professor with the School of Electronic Information Engineering, Tianjin University, Tianjin, China. His research interests include computer vision, web search and mining, pattern recognition, and digital image processing, in which he has published about 60 scientific papers including ten IEEE Transaction papers. Dr. Pang has served as Program Committee Member for more than 20 international conferences. He is on the Editorial Board of the *International Journal of Computer Mathematics*. He was elected as the New Century Excellent Talent in University, Ministry of Education, China. He is a senior member of the IEEE.



Junping Deng received the B.S. degree in Dalian University of Technology, Dalian, China in 2008, the M.S. degree in Tianjin University, Tianjin, China, 2010. He is currently pursuing the Ph.D. degree in Ritsumeikan University, Japan. His research interests include computer vision, tensor space learning, sparse representation and medical image processing.

Yuan Yuan is a professor at Chinese Academy of Sciences. Her research interests include visual information processing and Image/Video content analysis.