Proceedings of ITC-CSCC 2000, Pusan, Korea

Fast algorithm for online linear discriminant analysis

Kazuyuki Hiraoka, Masashi Hamahira, Ken-ichi Hidai,
Hiroshi Mizoguchi, Taketoshi Mishima, and Shuji Yoshizawa
Saitama University
Dept. of Information and Computer Sciences, 255 Shimo-Okubo, Urawa, 338, Japan
Tel: +81-48-852-2111, Fax: +81-48-858-3716
E-mail: hira@ics.saitama-u.ac.jp

Abstract: Linear discriminant analysis (LDA) is a basic tool of pattern recognition, and it is used in extensive fields, e.g. face identification. However, LDA is poor at adaptability since it is a batch type algorithm. To overcome this, a new algorithm of online LDA is proposed in the present paper. It is experimentally shown that the new algorithm is about two times faster than the previously proposed algorithm.

1 Introduction

Linear discriminant analysis (LDA) is a basic tool of pattern recognition, and it is used in extensive fields, e.g. face identification [5][6]. However, LDA is poor at adaptability since it is a batch type algorithm. Namely, LDA is designed in the following manner: (1) all sample images are given at once, (2) the discriminant matrix A is calculated for the sample images, and then (3) identification is performed by use of A. Owing to this design, we have to recalculate A every time when we add new data to update the identification system. This calculation is heavy for high dimensional data such as face images.

When the situation changes gradually or suddenly, one time learning is not sufficient and additional learning is indispensable for adaptability. Thus the identification system must have the ability to learn new data and update itself with small calculations. Such a algorithm that has this ability is called *online* learning algorithm. As we have mentioned above, conventional LDA is not online learning.

To overcome this disadvantage of LDA, the authors has been proposed an online LDA algorithm [8][9][10]. In contrast to the conventional LDA, updating the identification system according to new additional data can be executed with low computational cost by online LDA. Hence online LDA has the ability of adaptation to the change of environment. online LDA also has an advantage that huge matrices never appear in its calculation.

In the present paper, a new algorithm of online LDA is proposed. It is experimentally shown that the new algorithm is about two times faster than the previously proposed algorithm. Namely, the presented algorithm attains the same level of performance by half steps of learning compared with the original one. This advantage is mainly owed to the fact that new algorithm is robust for choice of the learning coefficient η.

2 Fast OLDA algorithm

At every time step $t = 1, 2, 3, \cdots$, a new pair $(\boldsymbol{x}(t), c(t))$ is presented, where $\boldsymbol{x}(t)$ is an N-dimensional data vector, $c(t) \in \{1, \cdots, M\}$ is the class of $\boldsymbol{x}(t)$, and M is the number of classes. Based on this pair, auxiliary variables are updated as follows:

$$
\begin{aligned}
t^c(t) &= t^c(t-1) + \delta(c, c(t)), && (1)\\
\bar{\boldsymbol{x}}(t) &= \left(1 - \frac{1}{t}\right)\bar{\boldsymbol{x}}(t-1) + \frac{1}{t}\boldsymbol{x}(t), && (2)\\
\bar{\boldsymbol{x}}^c(t) &= \\
&\begin{cases} \left(1 - \frac{1}{t^c(t)}\right)\bar{\boldsymbol{x}}^c(t-1) + \frac{1}{t^c(t)}\boldsymbol{x}(t) & (c = c(t)),\\ \bar{\boldsymbol{x}}^c(t-1) & (c \neq c(t)), \end{cases} && (3)\\
\boldsymbol{v}^c(t) &= \bar{\boldsymbol{x}}^c(t) - \bar{\boldsymbol{x}}(t), && (4)\\
\boldsymbol{w}(t) &= \boldsymbol{x}(t) - \bar{\boldsymbol{x}}^{c(t)}(t), && (5)
\end{aligned}
$$

$$
B(t) = \frac{1}{M}\sum_{c=1}^{M} \boldsymbol{v}^c(t)\boldsymbol{v}^c(t)^T, \qquad (6)
$$

where $c = 1, \cdots, M$ and $\delta(c, c(t)) = 1\,(c = c(t)),\ 0\,(c \neq c(t))$. Then $N \times L$ discriminant matrix A is updated:

$$
\begin{aligned}
A(t) = A(t-1) + \eta\Big[&B(t)A(t-1)\\
&- \frac{1}{2}A(t-1)\,A(t-1)^T B(t)\boldsymbol{w}(t)\boldsymbol{w}(t)^T A(t-1)\\
&- \frac{1}{2}A(t-1)\,A(t-1)^T \boldsymbol{w}(t)\boldsymbol{w}(t)^T B(t)A(t-1)\Big],
\end{aligned} \qquad (7)
$$

where the learning coefficient η is a small positive number. This updating rule is slightly different from the original one

$$
\begin{aligned}
A(t) = A(t-1) + \eta\Big[&B(t)A(t-1)\\
&- \frac{1}{2}B(t)A(t-1)\,A(t-1)^T \boldsymbol{w}(t)\boldsymbol{w}(t)^T A(t-1)\\
&- \frac{1}{2}\boldsymbol{w}(t)\boldsymbol{w}(t)^T A(t-1)\,A(t-1)^T B(t)A(t-1)\Big]
\end{aligned} \qquad (8)
$$

in [8][9][10].

The number L of features is less than or equal to $\min(N, M-1)$. Note that the covariance matrix $W(t) = \frac{1}{t}\sum_{\tau=1}^{t} \boldsymbol{w}(\tau)\boldsymbol{w}(\tau)^T$ within classes is replaced with the instantaneous value $\boldsymbol{w}(t)\boldsymbol{w}(t)^T$. Such replacement is justified by the theory of stochastic approximation [4]. As for the initial values, $t^c(0) = 0$, $\bar{\boldsymbol{x}}(0)$ and $\bar{\boldsymbol{x}}^c(0)$ are arbitrary vectors, and $A(0)$ is an arbitrary matrix which satisfies $\mathrm{rank}A(0) = L$.

In both (7) and (8), the learning coefficient η affects the performance of algorithms. In order to obtain fast convergence of the discriminant matrix $A(t)$, we want to set η as larger as possible. However, if η is too large, $A(t)$ can diverge. In section 3, it will be shown that the boundary of "acceptable" η in (7) is larger than that in (8). Thus, we can obtain fast convergence by the algorithm (7).

Note that the right hand side of (7) can be calculated efficiently in the following manner for the case $M \ll N$.

1. Instead of calculating B explicitly, calculate

$$
BA = \frac{1}{M}\sum_{c=1}^{M} \boldsymbol{v}^c\left(\boldsymbol{v}^{cT} A\right). \qquad (9)
$$

2. Calculate $A^T\boldsymbol{w}$ and $A^T B\boldsymbol{w} = (BA)^T\boldsymbol{w}$.

3. Calculate $A(A^T\boldsymbol{w})$ and $A(A^T B\boldsymbol{w})$.

4. Calculate $AA^T B\boldsymbol{w}\boldsymbol{w}^T A = (AA^T B\boldsymbol{w})(A^T\boldsymbol{w})^T$ and $AA^T\boldsymbol{w}\boldsymbol{w}^T BA = (AA^T\boldsymbol{w})(A^T B\boldsymbol{w})^T$.

3 Simulation

In this section, performance of online LDA algorithms is tested for face identification task (Fig. 1). It is experimentally shown that the presented algorithm is about two times faster than the original one. Namely, the presented algorithm attains the same level of performance by half steps of learning compared with the original one.

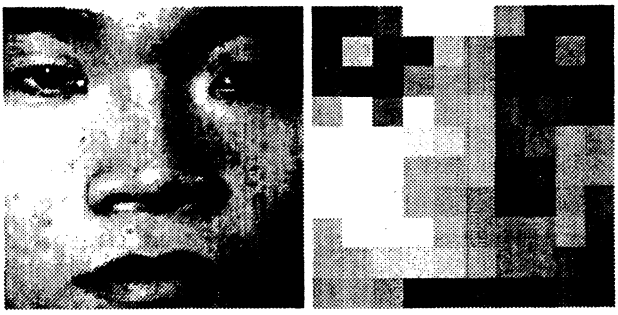

Figure 1: *A sample image for the simulation (left: original, right: reduced to* 10×10*)*

Table 1: Setting of the simulation

task	identification of face images
data vector $\boldsymbol{x}(t)$	face images under various illumination conditions (front view, 256 level gray scale, normalized to $[-1,+1]$)
size of $\boldsymbol{x}(t)$	$N = 10 \times 10 = 100$ (pixels)
number of classes to be identified	$M = 3$ (persons)
number of features	$L = 2$ (= number of columns in A)
initial values of elements in A	random values from the uniform distribution on $[-0.001, +0.001]$
learning coefficient	$\eta = 0.01$ or 0.03
regularization coefficient	$\epsilon = 0.01$ (applied only to the original algorithm (10))
procedure of learning	Face images for learning is presented in a random order.
procedure of evaluation	The ratio of the correct identification is evaluated for face images which are different from the face images for learning.
number of face images for learning	up to 500(images per person) × 3(persons) = 1500
number of face images for evaluation	100(images per person) × 3(persons) = 300
number of trials	100 independent trials with different random seeds

The setting of the simulation is written in Table 1. In the simulation of the original algorithm, the updating rule

$$\begin{aligned} A(t) &= A(t-1) + \eta\Big[B(t)A(t-1) \\ &- \frac{1}{2}B(t)A(t-1)\,A(t-1)^T(\boldsymbol{w}(t)\boldsymbol{w}(t)^T + \epsilon I)A(t-1) \\ &- \frac{1}{2}(\boldsymbol{w}(t)\boldsymbol{w}(t)^T + \epsilon I)A(t-1)\,A(t-1)^T B(t)A(t-1)\Big] \end{aligned} \tag{10}$$

is used instead of (8), where I is the identity matrix and the regularization coefficient ϵ is a small positive number. The term $+\epsilon I$ is useful for stabilization of the algorithm [10].

The result of the simulation is shown in Fig. 2 and Fig. 3. The presented algorithm has the advantage that we can take a large learning coefficient η so as to obtain fast convergence. On the other hand, the original algorithm is superior with regard to the final ratio of correct identification after sufficient learning.

4 Discussion

A key improvement of the presented algorithm is the fact that it is robust for choice of the learning coefficient η. By choosing a large learning coefficient, we can accelerate the learning process. However, as for the result after presentation of many samples, the original OLDA algorithm is superior. Thus, we may be able to obtain a better performance by combining them. Namely, the presented algorithm is used only in early stage. When saturation of the performance is observed, the algorithm is switched to the original one.

This work has been partly supported by CREST of JST (Japan Science and Technology) 279102.

References

[1] E. Oja, "A simplified neuron model as a principal component analyzer", J. Math. Biol., Vol. 15, pp. 267-273, 1982.

[2] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection", IEEE Tr. NN, Vol. 6, No. 2, pp. 296-317, 1995.

[3] C. Chatterjee and V. P. Roychowdhury, "On self-organizing algorithms and networks for class-separability features", IEEE Tr. NN, Vol. 8, No. 3, pp. 663-678, 1997.

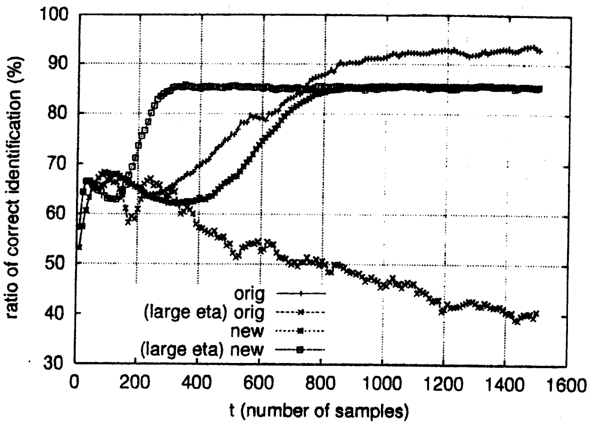

Figure 2: *Mean learning curves of the original algorithm ("orig") and the presented algorithm ("new") for* 100 *independent trials. Horizontal axis: number of presented samples. Vertical axis: percentage of correct discrimination. Two cases* $\eta = 0.01$ *and* $\eta = 0.03$*("large eta") are shown for each algorithm. The presented algorithm has the advantage that we can take a large learning coefficient* η *so as to obtain fast convergence. On the other hand, the original algorithm is superior with regard to the final ratio of correct identification after sufficient learning.*

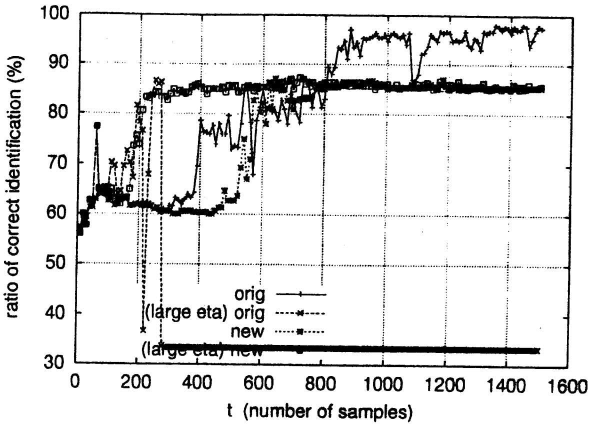

Figure 3: *Learning curves of the original algorithm ("orig") and the presented algorithm ("new") for one trial. Horizontal axis: number of presented samples. Vertical axis: percentage of correct discrimination. Two cases* $\eta = 0.01$ *and* $\eta = 0.03$*("large eta") are shown for each algorithm. The original algorithm fails for the large learning coefficient* $\eta = 0.03$.

[4] M. B. Nevel'son and R. Z. Has'minskiĭ, *Stochastic Approximation and Sequential Estimation*, Nauka, 1968 (in Russian; translated into Japanese by T. Kitagawa and K. Tajima, 1983).

[5] T. Kurita and S. Hayamizu, "Gesture Recognition using HLAC Features of PARCOR Images and HMM based Recognizer", Proc. 3rd int. conference on automatic face and gesture recognition, April, 1998, pp. 422-427.

[6] W. Zhao, et al., "Discriminant Analysis of Principal Components for Face Recognition", Proc. 3rd int. conference on automatic face and gesture recognition, 1998, pp. 422-427. pp. 336-341.

[7] K. Fukunaga, *Statistical Pattern Recognition*, Academic Press, New York, 1989.

[8] K. Hiraoka and M. Hamahira, "On Successive Learning Type Algorithm for Linear Discriminant Analysis", IEICE Technical Report, NC99-73, pp. 85–92, 1999 (in Japanese).

[9] Kazuyuki Hiraoka, Ken-ichi Hidai, Masashi Hamahira, Hiroshi Mizoguchi, Takaomi Shigehara, and Taketoshi Mishima "Derivation of online LDA algorithm and its application for face identification", Fifth Robotics Symposia, pp. 226–231, 2000 (in Japanese).

[10] Kazuyuki Hiraoka, Shuji Yoshizawa, Ken-ichi Hidai, Masashi Hamahira, Hiroshi Mizoguchi, and Taketoshi Mishima, "Convergence Analysis of Online Linear Discriminant Analysis", International Joint Conference on Neural Networks (IJCNN), 2000, *to appear.*