One-pass Incremental Membership Authentication by Face Classification

Shaoning Pang[1], Seiichi Ozawa[2] and Nikola Kasabov[1]

[1] Knowledge Engineering & Discover Research Institute
Auckland University of Technology, Private Bag 92006, Auckland 1020, New Zealand
Email: spang@aut.ac.nz, nkasabov@aut.ac.nz
[2] Graduate School of Science and Technology, Kobe University, Kobe 657-8501, Japan
Email: ozawa@eedept.kobe-u.ac.jp

Abstract. Real membership authentication applications require machines to learn from stream data while making a decision as accurately as possible whenever the authentication is needed. To achieve that, we proposed a novel algorithm which authenticated membership by a one-pass incremental principle component analysis(IPCA) learning. It is demonstrated that the proposed algorithm involves an useful incremental feature construction in membership authentication, and the incremental learning system works optimally due to its performance is converging to the performance of a batch learning system.

1 Introduction

Membership authentication problem [1] can be depicted as follows. Consider a certain human group G with N members, which is the universal set. If there exists an arbitrary subgroup M such that $M \subset G$ and $|M| < N/2$, then it is a membership group, and the remaining peoples $\bar{M} = G - M$ makes a non-membership group. Thus, membership authentication problem is to distinguish the membership class M from the non-membership class $\bar{M}$ in the human group.

Obviously, the membership authentication by face classification can be simply treated as a 2-class classification problem, in which either member or non-member is judged by the system as a human tries to get authentication. However, in the real situation of online membership authentication, the difficulties of the problem are as follows, (1)Both the size of membership/non-membership group and the members in the membership group/non-membership group are dynamically changed. (2)There are little within-class similarities, either for the membership class or for the nonmembership class. (3)Face images for training are provided as a stream dataset, which means that except for an initial dataset, other datasets are added incrementally.

To deal with the first three points of difficulty, our previous work [1] proposed an SVM ensemble method for membership authentication in dynamic face groups. To achieve a dynamical authentication, we performed the membership authentication in terms of binary classification without revealing the individual's identity. To deal with the classification difficulty from little within-class similarity, we used a SVM ensemble [2] combining several binary SVM classifiers. Here, towards solving the third difficulty, we introduce a algorithm of one-pass incremental membership authentication(IMA).

2 Incremental Membership Authentication: Problem Definition

Given $M = \{\boldsymbol{x}_i\}$ is a membership face group of G, and $\bar{M} = G - M$ is the non-membership face group, then membership authentication by face classification can be depicted by a typical binary classification problem as,

$$f(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in M, \\ -1 & \text{otherwise.} \end{cases} \tag{1}$$

Suppose f can be implemented by a specific classification model C with a set of parameters $W = \{w_1, w_2, ...w_\xi\}$, where ξ is determined by the used model in practice. In this paper, we simply use a K-NN classification model, then G represents the K-NN computation, and w is the prototypes (reference centers).

For the batch learning of Eq.(1), the whole training dataset is given before learning is started. The learning procedure can be illustrated as the following formula,

$$C(w_1, w_2, ...w_\xi) \leftarrow Train(M, \bar{M}). \tag{2}$$

In contrast to the batch learning mode, incremental learning samples are provided as a stream of data. It follows that the incremental learning must be a procedure of recursive learning that can memorize the new coming faces, while not forget the faces that are already learned before. Thus the incremental authentication can be formulated as a recursive computation,

$$\begin{gathered} C(w_1^t, w_2^t, ...w_\xi^t) \leftarrow \\ Train(C(w_1^{(t-1)}, w_2^{(t-1)}, ..., w_\xi^{(t-1)}), M_t, \bar{M_t}) \end{gathered} \tag{3}$$

where t identifies the current time, M_t and $\bar{M_t}$ denote current training set of membership and non-membership, respectively.

It is possible to apply batch learning method to learn stream data. But that requires system always to keep the previous training data and perform a repeating learning from scratch. The continually accumulating data might cause a memory disaster for the system.

3 One-pass Incremental Learning

3.1 Incremental Principal Component Analysis (IPCA)

Since the original PCA is not suited for incremental learning purposes, Hall and Martin devised a method to update eigenvectors and eigenvalues in an incremental way [4].

Assume that N training samples $\boldsymbol{x}_i \in R^n$ $(i = 1, \cdots, N)$ have been presented so far, and an eigenspace model $\Omega = (\boldsymbol{x}, \boldsymbol{U}, \boldsymbol{\Lambda}, N)$ is constructed by calculating the eigenvectors and eigenvalues from the covariance matrix of $\boldsymbol{x}_i$, where $\boldsymbol{x}$ is a mean input vector, $\boldsymbol{U}$ is a $n \times k$ matrix whose column vectors correspond to the eigenvectors, and $\boldsymbol{\Lambda}$ is a $k \times k$ matrix whose diagonal elements correspond to the eigenvalues. Here, k is the number of dimensions of the current eigenspace that is often determined such that a specified fraction of energy in the eigenvalue spectrum is retained.

Let us consider the case that the $(N+1)$th training sample $\boldsymbol{y}$ is presented. The addition of this new sample will lead to the changes in both of the mean vector and covariance matrix; therefore, the eigenvectors and eigenvalues should also be recalculated. The mean input vector $\boldsymbol{x}$ is easily updated as follows:

$$\boldsymbol{x}' = \frac{1}{N+1}(N\boldsymbol{x} + \boldsymbol{y}) \tag{4}$$

Therefore, the problem is how to update the eigenvectors and eigenvalues.

When the eigenspace model Ω is reconstructed to adapt to a new sample, we must check if the dimensions of the eigenspace should be changed or not. If the new sample has almost all energy in the current eigenspace, the dimensional augmentation is not needed in reconstructing the eigenspace. However, if it has some energy in the complementary space to the current eigenspace, the dimensional augmentation cannot be avoided. This can be judged from the norm of the following residue vector $\boldsymbol{h}$:

$$\boldsymbol{h} = (\boldsymbol{y} - \boldsymbol{x}) - \boldsymbol{U}\boldsymbol{g} \tag{5}$$

where $\boldsymbol{g} = \boldsymbol{U}^T(\boldsymbol{y} - \boldsymbol{x})$. When the norm of the residue vector $\boldsymbol{h}$ is larger than a threshold value η, it must allow the number of dimensions to increase from k to $k+1$, and the current eigenspace must be expanded in the direction of $\boldsymbol{h}$. Otherwise, the number of dimensions remains the same.

It has been shown that the eigenvectors and eigenvalues should be updated based on the solution of the following intermediate eigenproblem [4]:

$$\left(\frac{N}{N+1}\begin{bmatrix} \boldsymbol{\Lambda} & \boldsymbol{0} \\ \boldsymbol{0}^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2}\begin{bmatrix} \boldsymbol{g}\boldsymbol{g}^T & \gamma\boldsymbol{g} \\ \gamma\boldsymbol{g}^T & \gamma^2 \end{bmatrix}\right)\boldsymbol{R} = \boldsymbol{R}\boldsymbol{\Lambda}' \tag{6}$$

where $\gamma = \boldsymbol{h}^T(\boldsymbol{y} - \boldsymbol{x})$, $\boldsymbol{R}$ is a $(k+1) \times (k+1)$ matrix whose column vectors correspond to the eigenvectors obtained from the above intermediate eigenproblem, $\boldsymbol{\Lambda}'$ is the new eigenvalue matrix, and $\boldsymbol{0}$ is a k-dimensional zero vector. Using this solution $\boldsymbol{R}$, we can calculate the new $n \times (k+1)$ eigenvector matrix $\boldsymbol{U}'$ as follows:

$$\boldsymbol{U}' = [\boldsymbol{U},\ \boldsymbol{h}]\boldsymbol{R} \tag{7}$$

where

$$\boldsymbol{h} = \begin{cases} \boldsymbol{h}/\|\boldsymbol{h}\| & \text{if } \|\boldsymbol{h}\| > \eta \\ \boldsymbol{0} & \text{otherwise.} \end{cases} \tag{8}$$

As we can see from Eq. (7), $\boldsymbol{R}$ operates as a rotation of the eigenvectors; we will call $\boldsymbol{R}$ a rotation matrix. Note that if $\boldsymbol{h} = \boldsymbol{0}$, $\boldsymbol{R}$ degenerates into a $n \times k$ matrix; that is, the dimensions of the updated eigenspace remains the same as those of the previous eigenspace.

3.2 Evolving Clustering Method (ECM)

ECM is a fast one-pass algorithm for dynamic clustering of an input stream of data, where there is no predefined number of clusters. This algorithm is a distance-based clustering method where the cluster centers (called "prototypes") are determined online such that the maximum distance, *MaxDist*, between an input $\boldsymbol{x}_i$ and the closest prototype cannot be larger than a threshold value, *Dthr*. The ECM algorithm can be seen in [3].

3.3 The Proposed Method

As stated in [4], IPCA is utilized for reducing the dimensions of input data and constructing an appropriate feature space (i.e., eigenspace) based on an incoming data stream. In IPCA, depending on input data, the following two operations are carried out: eigen-axes rotation and dimensional augmentation of a feature space. On the other hand, ECM can evolve the prototypes which correspond to the representative points in the feature space constructed by IPCA. Hence, when the rotation and dimensional augmentation are carried out, all prototypes must be modified so as to keep consistency between the old and new eigenspaces.

When only the rotation is needed, each prototype $\boldsymbol{p}_j$ is easily updated by multiplying by the rotation matrix $\boldsymbol{R}$ as follows:

$$\boldsymbol{p}_j^{new} = \boldsymbol{R}^T \boldsymbol{p}_j, \text{ for } j = 1, \cdots, l. \tag{9}$$

where l is the number of prototypes. On the other hand, when the dimensional augmentation as well as the rotation is needed, the prototypes are not easy to be updated because the current prototypes have already lost information in the augmented complementary eigenspace. There are at least two methods to update prototypes. One is to approximate them as follows:

$$\boldsymbol{p}_j^{new} \simeq \boldsymbol{R}^T [\boldsymbol{p}_j^T, \ 0]^T, \text{ for } j = 1, \cdots, l \tag{10}$$

where $[\boldsymbol{p}_j^T, \ 0]^T$ is a $k+1$ dimensional column vector which is given by adding a zero element to the current prototype $\boldsymbol{p}_j$. Another method is to define the following 2-fold prototypes $(\boldsymbol{p}_j, \boldsymbol{p}_j^*)$, where $\boldsymbol{p}_j^*$ is the original vector in the input space associated with the prototype $\boldsymbol{p}_j$; i.e., $\boldsymbol{p}_j = \boldsymbol{U}^T \boldsymbol{p}_j^*$ is hold for these prototypes. When the dimensional augmentation occurs, the new prototypes are updated using $\boldsymbol{p}_j^*$ instead of $\boldsymbol{p}_j$ as follows:

$$\boldsymbol{p}_j^{new} = \boldsymbol{R}^T \boldsymbol{U}^T \boldsymbol{p}_j^*, \text{ for } j = 1, \cdots, l \tag{11}$$

Note that the new prototypes obtained by Eq. (10) are only an approximation to those obtained by Eq. (11). However, if the approximation error could be ignored, the first method is a good choice in the terms of memory costs because it does not require keeping the $\boldsymbol{p}_j^*$ prototype. Let us assume that a small number of initial training data is given to form an initial eigenspace. We propose the following unsupervised one-pass incremental learning algorithm,

Step 0: Calculate the eigenvector matrix $\boldsymbol{U}$ and eigenvalue matrix $\boldsymbol{\Lambda}$ from the covariance matrix of initial training data. Calculate the projection of training data $\boldsymbol{x}_i$ into the eigenspace to obtain the feature vectors $\tilde{\boldsymbol{x}}_i$. Apply ECM to the feature vectors, and keep the selected two-fold prototypes $(\boldsymbol{p}_j, \boldsymbol{p}_j^*)$.

Step 1: Apply IPCA to a new training sample $\boldsymbol{x}_i$ and the current eigenspace model $\Omega = (\boldsymbol{x}, \boldsymbol{U}, \boldsymbol{\Lambda}, N)$ as follows:

1. Solve an intermediate eigenproblem in Eq. (6) to obtain a rotation matrix $\boldsymbol{R}$ and an eigenvalue matrix $\boldsymbol{\Lambda}'$.
2. Update the mean input vector $\boldsymbol{x}$ and eigenvector matrix $\boldsymbol{U}'$ based on Eqs. (4) and (7), respectively.
3. Increase the total number of training samples N by one.

Step 2: Add $\boldsymbol{x}_i$ to a training set S, and increase the size $|S|$ of the training set by one.

Step 3: If $|S|$ is larger than a certain value B, go to Step 4. Else, go back to Step 1.

Step 4: If the dimensional augmentation is not needed, update all the current prototypes $\boldsymbol{p}_j$ based on Eq. (9). Else, update them based on Eq. (11).

Step 5: For all training samples $\boldsymbol{x}_i$ in S, calculate the feature vectors $\tilde{\boldsymbol{x}}_i$ as follows:

$$\tilde{\boldsymbol{x}}_i = \boldsymbol{U}'^T \boldsymbol{x}_i, \text{ for } i = 1, \cdots, B. \tag{12}$$

Step 6: Initialize the training set S.

Step 7: Apply ECM to the feature vectors $\tilde{\boldsymbol{x}}_i$ $(i = 1, \cdots, B)$, and keep the selected two-fold prototypes $(\boldsymbol{p}_j, \boldsymbol{p}_j^*)$.

Step 8: Go back to Step 1.

When a query sample is presented for classification purpose, the distances to all prototypes are calculated, and then the k nearest neighbor (k-NN) method is applied to determine the class.

4 Experiment Results and Discussions

4.1 Training

We perform various experiments to evaluate the proposed membership authentication technique using a benchmark MPEG-7 face dataset, which consists of 1,355 face images of 271 persons. (5 different face images per person are taken). We randomly selected a certain number of persons among 271 persons and assigned them as the membership group. The remaining persons are assigned as the non-membership group. To simulate the procedure of incremental membership

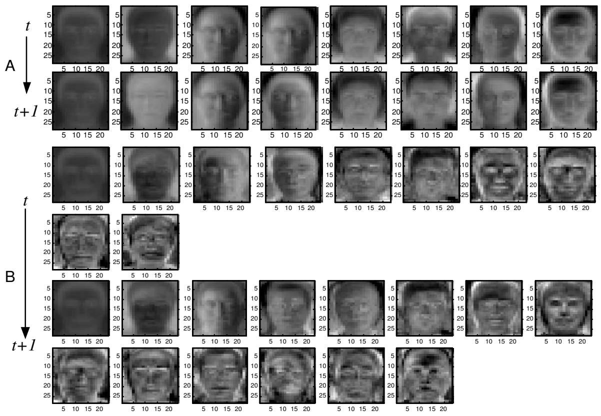

Fig. 1. Example of eigenspace variations.

authentication, we used a data division as follows: at each step, 4 out of 5 face images of each person are used for training, and the remaining 1 face for test. The learning procedure is composed of an initial learning phase and numbers of incremental learning stages. For the convenience of performance evaluation, except for the initial training dataset where we used 5% of 271 persons (i.e., 20 images), all the other stage dataset are set as 5 persons for incremental learning at each stage, where both the percentage and the stage size may vary. Here, selecting a small percentage is just for a more distinct difference between PCA and IPCA on feature deriving.

We implemented the proposed one-pass incremental learning algorithm on the above simulated stream data. We observed that rotation and augmentation happen alternately and frequently during the incremental learning process. As rotation happening, eigenvectors are rotated. Fig. 1 (A) is an example of eigenspace rotation. As we can see, the eigenfaces are changed between two incremental learnings, but the number of eighenfaces are kept the same. Whereas as augmentation happening, since it is usually accompanied by the happening of rotation, the eigenfaces are not only rotated, but also increased in the number. Fig. 1 (B) is an example of eigenspace augmentation with rotation.

4.2 Comparison of Results

Fig.2 shows the time courses of the authentication accuracy in the case of 50 persons are randomly selected as members. Where we carried out a comparison among three methods, the proposed one-pass IMA, IMA with PCA and SVM ensemble [1], and the results were represented by three curves in the figure respectively. As can be seen, the performance of incremental membership authentication with IPCA is stable even if the incremental learning for new member/non-member proceeds. While with no incremental feature construction

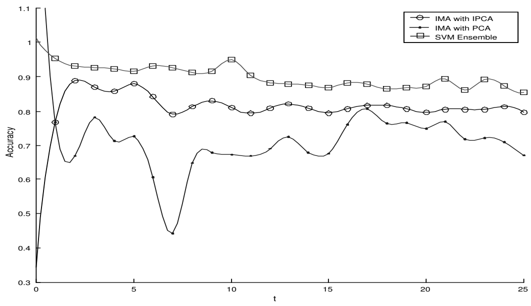

Fig. 2. Authentication comparison among IMA-IPCA, IMA-PCA and SVM ensemble

(using PCA), the system does not have such stability. As it is shown in the figure, some fluctuations appear in the curve of IMA-PCA. However, the performance of IMA-IPCA is slightly lower than that of the SVM ensemble in which the learning is conducted in a batch mode by using the whole of the training dataset. We can see that the curves of IMA-IPCA and SVM ensemble in the figure become closer as the learning is extended over time. It indicates that the proposed incremental system is convergent, and it can perform classification with the same accuracy as the batch system do, as long as a incremental learning is carried on.

5 Conclusions

In this paper, we proposed an algorithm of one-pass incremental membership authentication method by face classification. The algorithm processes a face stream efficiently in one pass, and learns to make a decision as accurately as possible whenever the authentication is needed. We compared the proposed IPCA feature-derived incremental authentication with the PCA feature-derived authentication and the traditional batch-modelling authentication, the performance of the proposed algorithm is found to be converging to the performance of batch learning as the learning procedure proceeds, which imply us a very optimal incremental learning of face membership authentication.

References

1. S. N. Pang, D. Kim, and S. Y. Bang, "Membership authentication in the dynamic group by face classification using SVM ensemble," Pattern Recognition Letters, Vol. 24, pp. 215-225, 2003.
2. S. N. Pang, D. Kim, and S. Y. Bang, "Fraud detection using support vector machine ensemble," Proc. ICORNIP2001, pp. 1344-1349, 2001.
3. N. Kasabov, Evolving connectionist systems: Methods and applications in bioinformatics, brain study and intelligent machines, Springer-Verlag, 2002.
4. P. Hall and R. Martin, "Incremental eigenanalysis for classification," British Machine Vision Conference, Vol. 1, pp. 286-295, 1998.