A METHOD FOR UPDATING A 2 DIMENSIONAL LINEAR DISCRIMINANT ANALYSIS (2DLDA) CLASSIFIER ENGINE

FIELD OF INVENTION

The present invention relates broadly to a method for updating a 2 dimensional linear discriminant analysis (2DLDA) classifier engine, to a 2 dimensional linear discriminant analysis (2DLDA) classifier engine and to a method and system for selecting samples from a pool comprising a plurality unlabeled samples.

BACKGROUND

Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are popular tools for dimension reduction and feature extraction. However, one common difficulty in using the PCA and LDA is that whenever one additional sample is presented, the system has to discard the training acquired in the past and repeat the learning process from the beginning. On the other hand, the ability to update the discriminant eigenspace with light computation instead of full re-training is crucial to using PCA and LDA in real-time face recognition, monitoring or surveillance applications.

Incremental learning can provide a way out of the above problem. In such learning scheme, the learning system, on being presented with new training data, updates its learning without having to reuse the current training samples. To achieve this, several existing methods perform incremental PCA (IPCA) learning by updating the eigenspace models. However, all these methods have only considered adding exactly one new sample to an eigenspace model at a time. One improved approach is a method of merging and splitting eigenspace models that allows a chunk of new samples to be learned in a single step. Incremental tensor PCA has also been applied in object tracking applications. One such method uses an on-line tensor subspace learning algorithm which models the appearance changes of a target by incrementally learning a low-order tensor eigenspace representation through adaptively updating the sample mean and eigenbasis.

A common aspect of existing LDA-based algorithms is the use of Singular Value Decomposition (SVD). Due to the difficulty of designing an incremental solution for the eigenvalue problem on the product of scatter matrices in LDA, there has been little work on designing incremental LDA algorithms. Recently, some incremental linear discriminant analysis (ILDA) methods have been used for the classification of data streams, for example, an ILDA which uses QR decomposition rather than SVD. Another method focuses on specific features that best discriminate the data so far, and an ILDA is based on directly updating the between-class scatter matrix and within-class scatter matrix. However it should be noted that the above ILDA methods use data that is presented in vectorised form. Thus, a limitation of the above methods is that the chunk size (the number of samples to be added each round of update) cannot be too large because the memory cost may become relatively too high. Although the large chunk size problem can be solved by partitioning a big chunk into several smaller ones and perform ILDA on these smaller chunks, the computational complexity may still be high.

In a typical incremental learning approach, an initial model is built using an initial set of sample(s) which may not be needed thereafter, and then the model is updated using a new sample, which gets discarded after the update. To cater for the case where wrong classification is caused by an initial bias model, an existing ILDA combines discriminative and reconstructive information. Also, an incremental LDA has been used for face recognition where generalized SVD is adopted. However, a difficulty for ILDA modelling, compared with previous IPCA modelling, is that all class data of a complete training dataset may not be presented at every incremental learning stage.

In addition, it should be appreciated that conventional PCA or LDA are based on vectors. Thus, when dealing with images, one must firstly convert the image matrices into image vectors, and then compute the covariance matrix of these vectors and finally extract the optimal projections from the covariance matrix. Using this strategy, Eigenfaces and Fisherfaces have been introduced into face recognition applications and achieved good performance. However, face images may be high-dimensional patterns. For example, an image of 112 × 92 pixel size forms a 10,304-dimension vector. Since the resulting image vectors are of high dimension, LDA usually encounters the Small Sample Size (SSS) problem in which the within-class scatter matrix becomes singular.

Various conventional methods have attempted to solve the SSS problem, e.g. PCA plus LDA has been used in ILDA. One important conventional approach is the two dimensional linear discriminant analysis (2DLDA) method. 2DLDA is an extension of LDA. The key difference between classical LDA and the 2DLDA is in the representation of data. While classical LDA uses the vectorized representation, 2DLDA works with data in matrix representation. In addition, 2DLDA has asymptotically minimum memory requirements, and lower time complexity than LDA, which is desirable for large face datasets. Also, 2DLDA uses the image matrix to calculate the between-class scatter matrix and the within-class scatter matrix. As the dimension of between-class and within-class scatter matrix may be much lower compared to the number of training samples, 2DLDA implicitly avoids the singularity problem encountered in classical LDA, i.e. the problem of the within-class scatter matrix becoming singular can be solved.

The first approach that attempts employing patterns in a matrix form in pattern recognition was applied to character recognition. There has also been a unilateral 2DLDA. More recently, a 2DLDA + LDA method for face recognition has been introduced. Also, there are some further extensions on 2DLDA to solve the small sample size problem, for example, a bilateral 2DLDA as described in [H. Kong, L. Wang, E.K. Teoh, J.-G. Wang and R. Venkateswarlu, A Framework of 2D Fisher Discriminant Analysis: Application to Face Recognition with Small Number of Training Samples, Proc. *CVPR* 2005, pp. 1083-1088], and a real-time 3D face recognition system based on the 2DLDA features as described in [J.-G. Wang, H. Kong, E. Sung, W.-Y. Yau, E. K. Teoh, Fusion of Appearance Image and Passive Stereo Depth Map for Face Recognition Based on the Bilateral 2DLDA, *EURASIP Journal on Image and Video Processing*, Volume 2007 (2007), Article ID 38205, 11 pages doi:10.1155/2007/38205], the contents of which are hereby incorporated by reference. However, all the 2DLDA methods discussed above do not consider incremental subspace analysis. Thus, the computational complexity may still be very high.

Meanwhile, a difficulty that has been noted in age categorization using face images is that the database is highly incomplete. In order to collect photos of a person, the subject may be required to scan his/her photos captured during the past at his/her different ages. On the other hand, there are a lot of unlabeled face images. Although the age range can be roughly estimated by humans from a face image, the labeling process can cost much time and require significant experience because incorrect labeling can happen depending on the subjective nature of the observer, the quality of the face images, the viewpoint, scenery, or simply the fact that somebody looks younger/older than his/her actual age etc.

So far, there has been relatively little literature on automatic age estimation compared to other facial image processing applications such as face recognition, facial gender recognition. Early algorithms are computationally expensive and thus not suitable for real-time applications. In addition, most of the conventional methods for age estimation are intended for accurate estimation of the actual age. However, it is difficult to accurately estimate an actual age from a face image because age progression is person-specific and the aging subspace is obtained based on a largely incomplete database. Also, for some applications such as digital signage, it is unnecessary to obtain the precise estimates of the actual age.

Active learning is a mechanism which aims to optimize the classification performance while minimizing the number of needed labeled samples. A key challenge in active learning is to minimize the selection of samples from the unlabeled pool to be labeled in order to fully learn the complete data available. The classification error of a sample typically serves as the criteria for selecting the samples. In order to select the most informative samples, the incorrectly classified samples are ordered using their classification errors. The idea for selecting a sample is that the worst samples (with the biggest error) should be added to the training samples and a new classifier will be learned using the new training database. However, as the data are unlabeled, one cannot tell which data is incorrect. Moreover, due to the large data set and high dimension of the feature, training with the complete data set is usually not feasible.

A need therefore exists to provide method for updating a 2 dimensional linear discriminant analysis (2DLDA) classifier engine and a 2 dimensional linear discriminant analysis (2DLDA) classifier engine that seek to address at least one of the above problems.

SUMMARY

In accordance with a first aspect of the present invention, there is provided a method for updating a 2 dimensional linear discriminant analysis (2DLDA) classifier engine for feature recognition, the method comprising the steps of:

providing one or more sample images to the classifier engine, the classifier engine comprising a plurality of classes derived from a plurality of training images and a mean matrix of all images;

updating the mean matrix of all images based on the sample images;

updating a between-class scatter matrix based on the sample images; and

updating a within-class scatter matrix based on the sample images.

The updating of the between-class scatter matrix may comprise updating a mean matrix of each class to which at least one of the sample images belongs prior to updating the between-class scatter matrix.

The updating of the within-class scatter matrix may comprise updating a mean matrix of each class to which at least one of the sample images belongs prior to updating the within-class scatter matrix.

In accordance with a second aspect of the present invention, there is provided a 2 dimensional linear discriminant analysis (2DLDA) classifier engine for feature recognition, the classifier engine comprising:

a plurality of classes derived from a plurality of training images;

a mean matrix of all images;

means for receiving one or more sample images;

means for updating the mean matrix of all images based on the sample images;

means for updating a between-class scatter matrix based on the sample images; and

means for updating a within-class scatter matrix based on the sample images.

In accordance with a third aspect of the present invention, there is provided a method for selecting samples from a pool comprising a plurality of unlabeled samples for active learning, the method comprising the steps of:

applying a 2 dimensional linear discriminant analysis (2DLDA) classifier engine to the unlabeled samples;

sorting the unlabeled samples in the pool according to their distances to a respective nearest neighbour,

selecting the sample with the furthest nearest neighbour for labeling; and

updating the 2DLDA classifier engine based on the labeled sample.

The updating of the 2DLDA classifier engine may comprise the method of the first aspect.

The method of the third aspect may be applied to face recognition.

The method of the third aspect may be applied to face age recognition.

The face age recognition may comprise determining whether a face belongs to one of the groups consisting of children, teen age, adult and senior adult.

In accordance with a fourth aspect of the present invention, there is provided a system for selecting samples from a pool comprising a plurality of unlabeled samples for active learning, the system comprising:

means for applying a 2 dimensional linear discriminant analysis (2DLDA) classifier engine to the unlabeled samples;

means for sorting the unlabeled samples in the pool according to their distances to a respective nearest neighbour;

means for selecting the sample with the furthest nearest neighbour for labeling;

means for updating the 2DLDA classifier engine based on the labeled sample;

means for obtaining the highest accuracy while labeling least unlabeled samples; and

means for removing the labeled sample from the pool.

In accordance with a fifth aspect of the present invention, there is provided a computer storage medium having stored thereon computer code means for instructing a computing device to execute a method for updating a 2 dimensional linear discriminant analysis (2DLDA) classifier engine for feature recognition, the method comprising the steps of:

providing one or more sample images to the classifier engine, the classifier engine comprising a plurality of classes derived from a plurality of training images and a mean matrix of all images;

updating the mean matrix of all images based on the sample images; and

updating a between-class scatter matrix based on the sample images.

In accordance with a sixth aspect of the present invention, there is provided a computer storage medium having stored thereon computer code means for instructing a computing device to execute a method for selecting samples from a pool comprising a plurality of unlabeled samples for active learning, the method comprising the steps of:

applying a 2 dimensional linear discriminant analysis (2DLDA) classifier engine to the unlabeled samples;

sorting the unlabeled samples in the pool according to their distances to a respective nearest neighbour,

selecting the sample with the furthest nearest neighbour for labeling; and

updating the 2DLDA classifier engine based on the labeled sample.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention will be better understood and readily apparent to one of ordinary skill in the art from the following written description, by way of example only, and in conjunction with the drawings, in which:

Fig. 1 (a) and Fig. 1 (b) show graphs of face recognition accuracy against the number of the incremental learning stages on the Olivetti Research Laboratory (ORL) and Extended Multi Modal Verification for Teleservices and Security applications (XM2VTS) databases respectively when new samples are sequentially added to a face recognition system according to an example embodiment.

Figure 2(a) and 2(b) show graphs of classification accuracy against number of new samples using chunk Incremental 2DLDA on the ORL and XM2VTS databases respectively according to an example embodiment.

Figures 3(a) and 3(b) show graphs comparing the execution times between batch 2DLDA and sequential Incremental 2DLDA against the number of learning

stages using ORL and XM2VTS databases respectively according to an example embodiment.

Figures 4(a) and 4(b) show graphs comparing the execution times between batch 2DLDA and chunk Incremental 2DLDA of various chunk sizes against the number of new samples using ORL and XM2VTS databases respectively according to an example embodiment.

Figures 5(a) and 5(b) show graphs comparing the memory costs of sequential ILDA and sequential Incremental 2DLDA using ORL and XM2VTS databases respectively according to an example embodiment.

Figures 6(a) and 6(b) show graphs comparing the memory costs of chunk ILDA and chunk Incremental 2DLDA using ORL and XM2VTS databases respectively according to an example embodiment.

Figure 7 shows a diagram illustrating the Furthest Nearest Neighbour method according to an example embodiment.

Figure 8 shows a flow chart illustrating a method for active learning according to an example embodiment.

Figures 9(a)-9(d) show sample images of the four age groups respectively according to an example embodiment.

Figure 10 shows sample unlabeled images in the pool according to an example embodiment.

Figure 11 shows a graph of classification accuracy versus the number of the selected samples according to an example embodiment.

Figure 12 shows a flow chart illustrating a method for updating a 2 dimensional linear discriminant analysis (2DLDA) classifier engine according to an example embodiment.

Figure 13 shows a schematic diagram of a computer system for implementing the method and system for updating a 2 dimensional linear discriminant analysis (2DLDA) classifier engine according to an example embodiment.

Figure 14 shows a flow chart illustrating a method for selecting samples from a pool comprising a plurality of unlabeled samples for active learning, according to an example embodiment.

DETAILED DESCRIPTION

Embodiments of the present invention can provide an exact solution of Incremental 2DLDA for updating the discriminant eigenspace as bursts of new class

data come in sequentially. Thus, the between-class and within-class matrix can be updated in the example embodiments without much recalculation. Two versions of incremental 2DLDA are described for two cases: add one new sample at one step (sequential) and add more than one new sample at one step (chunk).

Some portions of the description which follows are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as "scanning", "updating", "calculating", "determining", "replacing", "generating", "initializing", "identifying", or the like, refer to the action and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the computer system into other data similarly represented as physical quantities within the computer system or other information storage, transmission or display devices.

The present specification also discloses apparatus for performing the operations of the methods. Such apparatus may be specially constructed for the required purposes, or may comprise a general purpose computer or other device selectively activated or reconfigured by a computer program stored in the computer. The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose machines may be used with programs in accordance with the teachings herein. Alternatively, the construction of more specialized apparatus to perform the required method steps may be appropriate. The structure of a conventional general purpose computer will appear from the description below.

In addition, the present specification also implicitly discloses a computer program, in that it would be apparent to the person skilled in the art that the individual steps of the method described herein may be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the disclosure contained herein. Moreover, the computer program is not intended to be limited to any particular control flow. There are many other variants of the computer program, which can use different control flows without departing from the spirit or scope of the invention.

Furthermore, one or more of the steps of the computer program may be performed in parallel rather than sequentially. Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a general purpose computer. The computer readable medium may also include a hard-wired medium such as exemplified in the Internet

system, or wireless medium such as exemplified in the GSM mobile telephone system. The computer program when loaded and executed on such a general-purpose computer effectively results in an apparatus that implements the steps of the preferred method.

Overview of 2DLDA

In the example embodiments, $\{(X_1^1,C_1),...(X_{n_1}^1,C_1),...,(X_1^N,C_N),...,(X_{n_N}^N,C_N)\}$ are image samples from N classes. $X_i^k \in R^{r\times w}$ is the i^{th} ($r\times w$ image matrix) sample of the k^{th} class C_k, for $i = 1...n_k$ and n_k is the number of samples in class C_k, $\bar{x}_k = \frac{1}{n_k}\sum_{i=1}^{n_k} X_i^k$ is defined as the mean matrix of samples of the class C_k, $\mathbf{M} = \sum_{k=1}^{N} \frac{n_k}{T}\bar{\mathbf{x}}_k$ is the mean matrix of all samples, where T is the total number of samples for all N classes.

In addition, the example embodiments make use of the bilateral 2DLDA (B2DLDA) described in [H. Kong, L. Wang, E.K. Teoh, J.-G. Wang and R. Venkateswarlu, A Framework of 2D Fisher Discriminant Analysis: Application to Face Recognition with Small Number of Training Samples, Proc. *CVPR* 2005, pp. 1083-1088] and [J.-G. Wang, H. Kong, E. Sung, W.-Y. Yau, E. K. Teoh, Fusion of Appearance Image and Passive Stereo Depth Map for Face Recognition Based on the Bilateral 2DLDA, *EURASIP Journal on Image and Video Processing*, Volume 2007 (2007), Article ID 38205, 11 pages doi:10.1155/2007/38205] as the 2DLDA to be extended to Incremental 2DLDA. The B2DLDA is a general 2DLDA which finds a pair of (adjoint) discriminant vectors W_l and W_r satisfying:

$$(\mathbf{W}_l,\mathbf{W}_r) = \arg\max_{(\mathbf{W}_l,\mathbf{W}_r)} \frac{\sum_{k=1}^{N}\frac{n_k}{n}\mathbf{W}_l^T(\bar{\mathbf{x}}_k-\mathbf{M})\mathbf{W}_r\mathbf{W}_r^T(\bar{\mathbf{x}}_k-\mathbf{M})^T\mathbf{W}_l}{\frac{1}{n}\sum_{k=1}^{L}\sum_{i=1}^{n_k}\mathbf{W}_l^T(\mathbf{X}_i^k-\bar{\mathbf{x}}_k)\mathbf{W}_r\mathbf{W}_r^T(\mathbf{X}_i^k-\bar{\mathbf{x}}_k)^T\mathbf{W}_l} \quad (1)$$

The optimal W_l and W_r correspond to the eigenvectors of $S_{wl}^{-1}S_{bl}$ and $S_{wr}^{-1}S_{br}$ respectively, where S_{bl} and S_{br} are the left within-class and between-class scatter matrices of the training samples respectively; S_{wl} , S_{wr} are the right within-class and between-class scatter matrices of the training samples respectively. The pseudo-code for the B2DLDA algorithm is given as follows.

Algorithm B2DLDA (W_l, W_r, B_{l1}, B_{l2}, ...B_{ln}, B_{r1}, B_{r2}, ...,B_m) = B2DLDA(I_1, I_2,..., I_n, m_l, m_r)

Input: I_1, I_2, ...I_n, m_l, m_r

% I_i , i=1,2,...n, represent n images, and m_l and m_r are the number of the discriminant components of left and right B2DLDA transforms respectively

Output: W_l, W_r, B_{l1}, B_{l2}, ...B_{ln}, B_{r1}, B_{r2}, ...,B_{rn}

% W_l and W_r are the left and right transformation matrix respectively by the B2DLDA; B_{li} and B_{ri} are the reduced representations of I_i by W_l and W_r respectively

---

1. Compute the mean matrix, $\bar{\mathbf{x}}_i$, of the i^{th} class of each i
2. Compute the global mean matrix, M, of $\{I_i\}$, i=1,2...n
3. Find S_{bl} and S_{wl},

$$S_{bl} = \sum_{i=1}^{N} n_i \cdot (\bar{\mathbf{x}}_i - \mathbf{M})^T (\bar{\mathbf{x}}_i - \mathbf{M}) \quad (2)$$

$$S_{wl} = \sum_{i=1}^{N} \sum_{j=1}^{n_i} (\mathbf{X}_i^j - \mathbf{M}_i)^T (\mathbf{X}_i^j - \mathbf{M}_i) \quad (3)$$

4. Compute the first m_l eigenvectors $\{\phi_i^L\}_{i=1}^{m_l}$ of $S_{wl}^{-1} S_{bl}$

5. $W_l \leftarrow [\phi_1^L, \phi_2^L, \ldots, \phi_{m_l}^L]$ (4)

6. Find S_{br} and S_{wr},

$$S_{br} = \sum_{i=1}^{N} n_i \cdot (\bar{\mathbf{x}}_i - \mathbf{M})(\bar{\mathbf{x}}_i - \mathbf{M})^T \quad (5)$$

$$S_{wr} = \sum_{i=1}^{n} \sum_{j=1}^{n_i} (\mathbf{X}_i^j - \mathbf{M}_i)(\mathbf{X}_i^j - \mathbf{M})^T \quad (6)$$

7. Compute the first m_r eigenvectors $\{\phi_i^R\}_{i=1}^{m_r}$ of $S_{wr}^{-1} S_{br}$

8. $W_r \leftarrow [\phi_1^R, \phi_2^R, \ldots, \phi_{m_r}^R]$ (7)

9. $B_{li} = I_i * W_l$, i =1,...n (8)
$B_{ri} = I_i' * W_r$, i =1...n (9)

Incremental two dimensional linear discriminant analysis (Incremental 2DLDA)

The 2DLDA described above is usually trained on an entire batch of training samples, thus it can also be referred to as Batch 2DLDA. In some situations, however, not all of the training samples can be presented in advance. The method of the example embodiments can improve on the B2DLDA as described above by using incremental learning, which is an effective method that can adapt a low dimensional eigenspace representation to reflect appearance changes of the target, thereby facilitating the recognition task. It can update the current subspace states using only the new samples without the need for the past samples. The method of the example embodiments is herein referred to as incremental two dimensional linear discriminant analysis (Incremental 2DLDA).

In the incremental 2DLDA of the example embodiments, the between-class scatter matrix and the within-class scatter matrix are updated based on the new sample/samples. For evaluation purposes, an initial discriminant eigenspace is learnt using a database of samples by randomly selecting only a part of the available original database as the training samples. The remaining part is treated as new samples. Assuming M is the overall mean matrix of all the training samples, the mean matrix of the class m is $\bar{x}_m$, for m = 1,2,...,N. S_w and S_b are the within- and between- class scatter matrices respectively. The two cases are described as follows.

Sequential incremental 2DLDA

In one embodiment, referred to herein as Sequential incremental 2DLDA, only a new sample is added at each step. Let Y be a new sample which is labeled as belonging to class l_Y, the overall mean matrix is updated in the example embodiment as

$$\mathbf{M}' = (n\mathbf{M} + \mathbf{Y})/(n+1) \quad (10)$$
$$n = n+1$$

If Y is a face image of a new subject, i.e. Y is a sample of a subject which does not appear in the current training set, then

$N = N+1$, $n_{N+1} = 1$

On the other hand, if Y is found from the present trained 2DLDA classifier to be a sample of an existing class (e.g. person), the mean matrix of the class l_Y is updated in the example embodiment as

$$\bar{\mathbf{x}}'_{l_Y} = \frac{1}{n_{l_Y}+1}(n_{l_Y}\bar{\mathbf{x}}_{l_Y} + \mathbf{Y}), \tag{11}$$

$$n'_{l_Y} = n_{l_Y} + 1$$

The between-class scatter matrix is updated as follows

$$\mathbf{S}'_{bl} = \sum_{c=1}^{N} n_c(\bar{\mathbf{x}}_c - \mathbf{M}')^T(\bar{\mathbf{x}}_c - \mathbf{M}') \tag{12}$$

$$\mathbf{S}'_{br} = \sum_{c=1}^{N} n_c(\bar{\mathbf{x}}_c - \mathbf{M}')(\bar{\mathbf{x}}_c - \mathbf{M}')^T \tag{13}$$

The within-class scatter matrix is updated similarly.

If Y is a face image of a new subject, a new class is created. The within-class scatter matrix remains unchanged:

$$\mathbf{S}'_{wr} = \mathbf{S}_{wr} \tag{14}$$

$$\mathbf{S}'_{wl} = \mathbf{S}_{wl} \tag{15}$$

else

$$\mathbf{S}'_{wl} = \sum_{il} + \frac{n_{l_Y}}{n_{l_Y}+1}(\mathbf{Y} - \bar{\mathbf{x}}_{l_Y})^T(\mathbf{Y} - \bar{\mathbf{x}}_{l_Y}) \tag{16}$$

$$\mathbf{S}'_{wr} = \sum_{ir} + \frac{n_{l_Y}}{n_{l_Y}+1}(\mathbf{Y} - \bar{\mathbf{x}}_{l_Y})(\mathbf{Y} - \bar{\mathbf{x}}_{l_Y})^T \tag{17}$$

where

$$\sum_{il} = \sum_{c=1}^{N}\sum_{ic} = \sum_{c=1}^{N}\sum_{i}(\mathbf{X}_i^c - \bar{\mathbf{x}}_c)^T(\mathbf{X}_i^c - \bar{\mathbf{x}}_c) \tag{18}$$

$$\sum_{ir} = \sum_{c=1}^{N}\sum_{ic} = \sum_{c=1}^{N}\sum_{i}(\mathbf{X}_i^c - \bar{\mathbf{x}}_c)(\mathbf{X}_i^c - \bar{\mathbf{x}}_c)^T \tag{19}$$

Once S_w and S_b are updated using the new sample, the feature extraction can be done in the same way as with 2DLDA.

The pseudo-code of the Incremental 2DLDA algorithm according to the example embodiment is given in Algorithm Incremental 2DLDA[a] below. When a new sample is added, the algorithm updates the discriminant eigenspace based on the new sample.

Algorithm Incremental 2DLDA[a] ($\mathbf{M}'$, $\bar{\mathbf{x}}'_k$, S_b', S_w') = Incremental 2DLDA ({Y,l_Y}, $\mathbf{M}$, $\bar{\mathbf{x}}_k$, S_b, S_w, n_k, n, N)

Input: Y, l_Y, $\mathbf{M}$, $\bar{\mathbf{x}}_k$, k=1,2,...N; S_b, S_w, n_k, k=1,2,...N

%Y : new training sample

% l_Y: the label of Y
% n: number of the samples in the database
% N: number of the classes in the database
% $\mathbf{M}$ and $\bar{\mathbf{x}}_k$ (k=1,2,...,N) are the mean matrix of the all training samples and the mean matrix of the k-th class respectively;
% $\mathbf{S}_b$ and $\mathbf{S}_w$ are the between-class scatter matrix and within-class scatter matrix respectively;
% n_k: the number of the samples for the k-th class

Output: $\mathbf{M}'$, $\bar{\mathbf{x}}_k'$, k=1,2,...; $\mathbf{S}'_b$, $\mathbf{S}'_w$

%$\mathbf{M}'$ and $\bar{\mathbf{x}}_k'$ (k=1,2,...,N or 1,2,...N+1) are the updated mean matrix of the all training samples and the updated mean matrix of the k-th class respectively;
% $\mathbf{S}_b'$ and $\mathbf{S}_w'$ are the updated between-class scatter matrix and the updated within-class scatter matrix respectively;

---

1. Update the mean matrix of all samples

$$\mathbf{M}'=(n\mathbf{M}+\mathbf{Y})/(n+1) \quad (20)$$

$$n=n+1$$

2. Update the between-class scatter matrix

$$\mathbf{S}'_{bl}=\sum_{c=1}^{N} n'_c(\bar{\mathbf{x}}_c-\mathbf{M}')^T(\bar{\mathbf{x}}_c-\mathbf{M}') \quad (21)$$

$$\mathbf{S}'_{br}=\sum_{c=1}^{N} n'_c(\bar{\mathbf{x}}_c-\mathbf{M}')(\bar{\mathbf{x}}_c-\mathbf{M}')^T \quad (22)$$

If Y is a face image of a new subject, N=N+1, n'_{N+1} =1
If Y is not a face image of a new subject,

$$\bar{\mathbf{x}}'_{l_Y}=\frac{1}{n_{l_Y}+1}(n_{l_Y}\bar{\mathbf{x}}_{l_Y}+\mathbf{Y}) \quad (23)$$

$$n'_{l_Y}=n_{l_Y}+1$$

3. Update the within-class scatter matrix

If Y is a face image of a new subject,

$$\mathbf{S}'_{wl}=\mathbf{S}_{wl} \quad (24)$$

$$\mathbf{S}'_{wr}=\mathbf{S}_{wr} \quad (25)$$

N=N+1
n_{N+1} =1

If Y is not face image of a new subject,

$$\mathbf{S}'_{wl}=\mathbf{S}_{wl}+\frac{n_{l_Y}}{n_{l_Y}+1}(\mathbf{Y}-\bar{\mathbf{x}}_{l_Y})^T(\mathbf{Y}-\bar{\mathbf{x}}_{l_Y}) \quad (26)$$

$$\mathbf{S}'_{wr}=\mathbf{S}_{wr}+\frac{n_{l_Y}}{n_{l_Y}+1}(\mathbf{Y}-\bar{\mathbf{x}}_{l_Y})(\mathbf{Y}-\bar{\mathbf{x}}_{l_Y})^T \quad (27)$$

$$\bar{\mathbf{x}}'_{l_Y}=\frac{1}{n_{l_Y}+1}(n_{l_Y}\bar{\mathbf{x}}_{l_Y}+\mathbf{Y}) \quad (28)$$

$$n'_{l_Y} = n_{l_Y} + 1$$

The proof of Equation (27) is as follows.

Assume C_{l_Y} denotes the set of the training samples with label l_Y.

$$S'_{wr} = \sum_{X \in \{C_{l_Y}, Y\}} (X - \bar{x}'_{l_Y})(X - \bar{x}'_{l_Y})^T$$

$$= \sum_{X \in C_{l_Y}} (X - \bar{x}'_{l_Y})(X - \bar{x}'_{l_Y})^T + (Y - \bar{x}'_{l_Y})(Y - \bar{x}'_{l_Y})^T$$

$$= \sum_{X \in C_{l_Y}} (X - \bar{x}_{l_Y} - \frac{Y - \bar{x}_{l_Y}}{n_{l_Y} + 1})(X - \bar{x}_{l_Y} - \frac{Y - \bar{x}_{l_Y}}{n_{l_Y} + 1})^T + (Y - \bar{x}_{l_Y} - \frac{Y - \bar{x}_{l_Y}}{n_{l_Y} + 1})(Y - \bar{x}_{l_Y} - \frac{Y - \bar{x}_{l_Y}}{n_{l_Y} + 1})^T$$

$$= S_{wr} + \sum_i \left[\frac{(Y - \bar{x}_{l_Y})(Y - \bar{x}_{l_Y})^T}{(n_{l_Y} + 1)^2} - (X_i^{l_Y} - \bar{x}_{l_Y}) \frac{(Y - \bar{x}_{l_Y})^T}{n_{l_Y} + 1} - \frac{(Y - \bar{x}_{l_Y})}{n_{l_Y} + 1} (X_i^{l_Y} - \bar{x}_{l_Y})^T \right] +$$

$$\frac{n_{l_Y}^2}{(n_{l_Y} + 1)^2} (Y - \bar{x}_{l_Y})(Y - \bar{x}_{l_Y})^T$$

As $\sum_i (X_i^{l_Y} - \bar{x}_{l_Y}) = 0$,

$$S'_{wr} = S_{wr} + \frac{n_{l_Y}(n_{l_Y} + 1)}{(n_{l_Y} + 1)^2} (Y - \bar{x}_{l_Y})(Y - \bar{x}_{l_Y})^T = S_{wr} + \frac{n_{l_Y}}{n_{l_Y} + 1} (Y - \bar{x}_{l_Y})(Y - \bar{x}_{l_Y})^T$$

Hence,

$$S'_{wr} = S_{wr} + \frac{n_{l_Y}}{n_{l_Y} + 1} (Y - \bar{x}_{l_Y})(Y - \bar{x}_{l_Y})^T$$

Similarly,

$$S'_{wl} = S_{wl} + \frac{n_{l_Y}}{n_{l_Y} + 1} (Y - \bar{x}_{l_Y})^T (Y - \bar{x}_{l_Y})$$

Chunk incremental 2DLDA

In another embodiment, referred to herein as Chunk incremental 2DLDA, multiple samples are added at each step. The Sequential incremental 2DLDA described above is therefore a special case of the Chunk incremental 2DLDA.

It will be appreciated that if more than one new sample is provided, it may be more efficient to use all of them to retrain the Incremental 2DLDA rather than to retrain it with a single new sample at a time. In the example embodiment, assume t new samples are given and their labels, $Y=\{\{Y_1, l_{Y1}\}, \{Y_2, l_{Y2}\}, \ldots, \{Y_t, l_{Yt}\}\}$. Without loss of generality, assume there are q_m new samples, $\{Y_l\}_{Y_l \in Y \cap l = m}$ which belong to the m^{th} class. The mean matrix of the m^{th} class is updated in the example embodiment as follows:

$$\bar{x}'_m = (n_m \bar{x}_m + \sum_{Y_l \in Y \cap l = m} Y_l) / (n_m + q_m) \quad (31)$$

$n_m = n_m + q_m$

The updated overall mean matrix is

$$\mathbf{M}' = \frac{n\mathbf{M} + \sum_{i=1}^{t} \mathbf{Y}_i}{n+t} \tag{32}$$

The between-class scatter matrix is updated by

$$\mathbf{S}_{bl}' = \sum_{c=1}^{N} n_c'(\bar{\mathbf{x}}_c' - \mathbf{M}')^T(\bar{\mathbf{x}}_c' - \mathbf{M}') \tag{33}$$

$$\mathbf{S}_{br}' = \sum_{c=1}^{N} n_c'(\bar{\mathbf{x}}_c' - \mathbf{M}')(\bar{\mathbf{x}}_c' - \mathbf{M}')^T \tag{34}$$

The within-class scatter matrix is updated by:

$$\mathbf{S}_{wl}' = \sum_{c=1}^{N}\sum_{c}' = \sum_{c=1}^{N}\left(\sum_{c} + \frac{n_c q_c^2}{(n_c+q_c)^2}(\bar{\mathbf{y}}_c - \bar{\mathbf{x}}_c)^T(\bar{\mathbf{y}}_c - \bar{\mathbf{x}}_c) + \right. \tag{35}$$
$$\left. \frac{n_c^2}{(n_c+q_c)^2}\sum_{\mathbf{Y}_k \in \mathbf{Y} \cap l_k = c}(\mathbf{Y}_k - \bar{\mathbf{x}}_c)^T(\mathbf{Y}_k - \bar{\mathbf{x}}_c) + \frac{q_c(q_c+2n_c)}{(n_c+q_c)^2}\sum_{\mathbf{Y}_k \in \mathbf{Y} \cap l_k = c}(\mathbf{Y}_k - \bar{\mathbf{y}}_c)^T(\mathbf{Y}_k - \bar{\mathbf{y}}_c)\right)$$

$$\mathbf{S}_{wr}' = \sum_{c=1}^{N}\sum_{c}' = \sum_{c=1}^{N}\left(\sum_{c} + \frac{n_c q_c^2}{(n_c+q_c)^2}(\bar{\mathbf{y}}_c - \bar{\mathbf{x}}_c)(\bar{\mathbf{y}}_c - \bar{\mathbf{x}}_c)^T + \right. \tag{36}$$
$$\left. \frac{n_c^2}{(n_c+q_c)^2}\sum_{\mathbf{Y}_k \in \{\mathbf{Y}\} \cap l_k = c}(\mathbf{Y}_k - \bar{\mathbf{x}}_c)(\mathbf{Y}_k - \bar{\mathbf{x}}_c)^T + \frac{q_c(q_c+2n_c)}{(n_c+q_c)^2}\sum_{\mathbf{Y}_k \in \mathbf{Y} \cap l_k = c}(\mathbf{Y}_k - \bar{\mathbf{y}}_c)(\mathbf{Y}_k - \bar{\mathbf{y}}_c)^T\right)$$

where $\bar{\mathbf{y}}_c$ is the mean matrix of the new samples of the class c.

If the samples belong to a new subject, assume l_{N+1} of L new samples belong to the N+1 class, the between class scatter matrix is updated:

$$\mathbf{S}_{bl}' = \sum n_c'(\bar{\mathbf{x}}_c - \mathbf{M}')^T(\bar{\mathbf{x}}_c - \mathbf{M}') + q_{N+1}(\bar{\mathbf{y}} - \mathbf{M}')^T(\bar{\mathbf{y}} - \mathbf{M}') \tag{37}$$

$$\mathbf{S}_{br}' = \sum n_c'(\bar{\mathbf{x}}_c - \mathbf{M}')(\bar{\mathbf{x}}_c - \mathbf{M}')^T + q_{M+1}(\bar{\mathbf{y}} - \mathbf{M}')(\bar{\mathbf{y}} - \mathbf{M}')^T \tag{38}$$

The within-class scatter matrix is updated:

$$\mathbf{S}_{wl}' = \mathbf{S}_{wl} + \sum_{(N+1)l} \tag{39}$$

$$\mathbf{S}_{wr}' = \mathbf{S}_{wr} + \sum_{(N+1)r} \tag{40}$$

where $\sum_{(N+1)l}$ and $\sum_{(N+1)r}$ are the left and right scatter matrix of the $(N+1)^{th}$ class and:

$$\sum_{(N+1)l} = \sum_{\mathbf{Y}_k \in \mathbf{Y} \cap l_k = c}(\mathbf{Y}_k - \bar{\mathbf{y}}_c)^T(\mathbf{Y}_k - \bar{\mathbf{y}}_c) \tag{41}$$

$$\sum_{(N+1)r} = \sum_{\mathbf{Y}_k \in \mathbf{Y} \cap l_k = c}(\mathbf{Y}_k - \bar{\mathbf{y}}_c)(\mathbf{Y}_k - \bar{\mathbf{y}}_c)^T \tag{42}$$

The proof of Equations (35) and (36) is given as follows:

According to the definition of the between-class scatter matrix,

$$\sum_{c}' = \sum_{i=1} n_c(\mathbf{X}_c^i - \mathbf{M}')(\mathbf{X}_c^i - \mathbf{M}')^T + \sum_{i=1} q_c(\mathbf{Y}_i - \mathbf{M}')(\mathbf{Y}_i - \mathbf{M}')^T$$

Replacing $\mathbf{M}'$ in the above equation with (32) and using the fact that

$$\sum_{i=1}^{q_c}(\overline{\mathbf{x}}\overline{\mathbf{y}}^T - \overline{\mathbf{x}}\mathbf{Y}_i^T) = \sum_{i=1}^{q_c}(\overline{\mathbf{y}}\overline{\mathbf{x}}^T - \overline{\mathbf{y}}\mathbf{X}_i^c) = \sum_{i=1}^{q_c}(\overline{\mathbf{y}}\overline{\mathbf{y}}^T - \mathbf{Y}_i\overline{\mathbf{y}}) = 0$$

in the above equation, the following is obtained:

$$\sum\nolimits_c' = \sum[(\mathbf{X}_i^c - \mathbf{M})(\mathbf{X}_i^c - \mathbf{M})^T - \frac{q_c}{q_c + n_c}(\overline{\mathbf{y}} - \mathbf{M})][(\mathbf{X}_i^c - \mathbf{M})(\mathbf{X}_i^c - \mathbf{M})^T - \frac{q_c}{q_c + n_c}(\overline{\mathbf{y}} - \mathbf{M})]^T$$

$$+\frac{1}{(n_c + q_c)^2}\sum_{i=1}^{} q_c[n_c(\mathbf{Y}_i - \mathbf{M}) + q_c(\mathbf{Y}_i - \overline{\mathbf{y}})][n_c(\mathbf{Y}_i - \mathbf{M}) + q_c(\mathbf{Y}_i - \overline{\mathbf{y}})]^T$$

$$= \sum_{i=1}^{n_c}(\mathbf{X}_i^c - \mathbf{M})(\mathbf{X}_i^c - \mathbf{M})^T + \frac{q_c^2}{(q_c + n_c)^2}\sum_{i=1}^{n_c}(\overline{\mathbf{y}} - \mathbf{M})(\overline{\mathbf{y}} - \mathbf{M})^T + \frac{1}{(n_c + q_c)^2}[n_c^2\sum_{i=1}^{} q_c(\mathbf{Y} - \mathbf{M})(\mathbf{Y} - \mathbf{M})^T$$

$$+(2n_c q_c + q_c^2)\sum_{i=1}^{q_c}(\mathbf{Y}_i\,\mathbf{Y}_i^T - \mathbf{Y}_i\overline{\mathbf{y}}^T)$$

$$= \sum_{i=1}^{n_c}(\mathbf{X}_i^c - \mathbf{M})(\mathbf{X}_i^c - \mathbf{M})^T + \frac{q_c^2}{(q_c + n_c)^2}\sum_{i=1}^{n_c}(\overline{\mathbf{y}} - \mathbf{M})(\overline{\mathbf{y}} - \mathbf{M})^T$$

$$+\frac{1}{(n_c + q_c)^2}[n_c\sum_{i=1}^{} q_c(\mathbf{Y}_i - \mathbf{M})(\mathbf{Y}_i - \mathbf{M})^T + q_c(q_c + 2n_c)\sum_{i=1}^{q}(\mathbf{Y}_i - \overline{\mathbf{y}})(\mathbf{Y}_i - \overline{\mathbf{y}})^T$$

$$= \sum\nolimits_c' + \frac{n_c q_c^2}{(n_c + q_c)^2}(\overline{\mathbf{y}}_c - \overline{\mathbf{x}}_c)(\overline{\mathbf{y}}_c - \overline{\mathbf{x}}_c)^T + \frac{n_c^2}{(n_c + q_c)^2}\sum_{\mathbf{Y}_k \in \mathbf{Y} \cap l_k = c}(\mathbf{Y}_k - \overline{\mathbf{x}}_c)(\mathbf{Y}_k - \overline{\mathbf{x}}_c)^T$$

$$+\frac{q_c(q_c + 2n_c)}{(n_c + q_c)^2}\sum_{\mathbf{Y}_k \in \mathbf{Y} \cap l_k = c}(\mathbf{Y}_k - \overline{\mathbf{y}}_c)(\mathbf{Y}_k - \overline{\mathbf{y}}_c)^T)$$

The pseudo-code of the Chunk Incremental 2DLDA algorithm according to the example embodiment is given in the Algorithm Incremental 2DLDA^b.

Algorithm Incremental 2DLDA^b ($\overline{\mathbf{x}}_k'$, $\mathbf{S}_b'$, $\mathbf{S}_w'$) = Incremental 2DLDA ($\{\mathbf{Y}_1, l_{y1}\}, \{\mathbf{Y}_2, l_{y2}\}, \ldots \{\mathbf{Y}_t, l_{yt}\}$, m, $\mathbf{M}$, $\overline{\mathbf{x}}_k$, $\mathbf{S}_b$, $\mathbf{S}_w$, n_k, $\mathbf{M}'$, n, N)

Input: $\{\mathbf{Y}_i, l_{Yi}\}_{i=1\ldots t}$, $\mathbf{M}$, $\overline{\mathbf{x}}_k$, k=1,2,...N; $\mathbf{S}_b$, $\mathbf{S}_w$, n_k, k=1,2,...N

%$\{\mathbf{Y}_i, l_{Yi}\}$, i=1,2,...,t: new training samples and labels;

% n: number of the samples in the database

% N: number of the classes in the database

% $\mathbf{M}$ and $\overline{\mathbf{x}}_k$ (k=1,2,...,N) are the mean matrix of the all training samples and the mean matrix of the k-th class respectively;

% $\mathbf{S}_b$ and $\mathbf{S}_w$ are the between-class scatter matrix and within-class scatter matrix respectively;

% n_k: the number of the samples for the k-th class

Output: $\mathbf{M}'$, $\overline{\mathbf{x}}_k'$, k=1,2,...; $\mathbf{S}'_b$, $\mathbf{S}'_w$

%$\mathbf{M}'$ and $\bar{\mathbf{x}}_k'$ (k=1,2,...,N or 1,2,...N+1) are the updated mean matrix of the all training samples and the updated mean matrix of the k-th class respectively;
% $\mathbf{S}_b'$ and $\mathbf{S}_w'$ are the updated between-class scatter matrix and the updated within-class scatter matrix respectively;

1. Update the mean matrix of all samples

$$\mathbf{M}'=(n*\mathbf{M}+\Sigma\mathbf{Y})/(n+t) \quad (43)$$
$$n=n+t$$

2. Update the mean matrix of the classes

$$\bar{\mathbf{x}}_m' = (n_m\bar{\mathbf{x}}_m + l_m\bar{\mathbf{y}}_m)/(n_m + l_m) \quad (44)$$

3. Update the between-class scatter matrix

$$\mathbf{S}_{bl}' = \sum_{c=1}^{N} n_c'(\bar{\mathbf{x}}_c' - \mathbf{M}')^T(\bar{\mathbf{x}}_c' - \mathbf{M}') \quad (45)$$

$$\mathbf{S}_{br}' = \sum_{c=1}^{N} n_c'(\bar{\mathbf{x}}_c' - \mathbf{M}')(\bar{\mathbf{x}}_c' - \mathbf{M}')^T \quad (46)$$

4. Update the within-class scatter matrix

$$\mathbf{S}_{wl}' = \sum_{c=1}^{N}\sum_{c}' = \sum_{c=1}^{N}\left(\sum_{c} + \frac{n_c q_c^2}{(n_c+q_c)^2}(\bar{\mathbf{y}}_c - \bar{\mathbf{x}}_c)^T(\bar{\mathbf{y}}_c - \bar{\mathbf{x}}_c) + \right.$$
$$\left. \frac{n_c^2}{(n_c+q_c)^2}\sum_{\mathbf{Y}_l\in\mathbf{Y}\cap l_k=c}(\mathbf{Y}_l - \bar{\mathbf{x}}_c)^T(\mathbf{Y}_l - \bar{\mathbf{x}}_c) + \frac{q_c(q_c+2n_c)}{(n_c+q_c)^2}\sum_{\mathbf{Y}_k\in\mathbf{Y}\cap l_k=c}(\mathbf{Y}_k - \bar{\mathbf{y}}_c)^T(\mathbf{Y}_k - \bar{\mathbf{y}}_c)\right) \quad (47)$$

$$\mathbf{S}_{wr}' = \sum_{c=1}^{N}\sum_{c}' = \sum_{c=1}^{N}\left(\sum_{c} + \frac{n_c q_c^2}{(n_c+q_c)^2}(\bar{\mathbf{y}}_c - \bar{\mathbf{x}}_c)(\bar{\mathbf{y}}_c - \bar{\mathbf{x}}_c)^T + \right.$$
$$\left. \frac{n_c^2}{(n_c+q_c)^2}\sum_{\mathbf{Y}_k\in\mathbf{Y}\cap l_k=c}(\mathbf{Y}_k - \bar{\mathbf{x}}_c)(\mathbf{Y}_k - \bar{\mathbf{x}}_c)^T + \frac{q_c(q_c+2n_c)}{(n_c+q_c)^2}\sum_{\mathbf{Y}_k\in\mathbf{Y}\cap l_k=c}(\mathbf{Y}_k - \bar{\mathbf{y}}_c)(\mathbf{Y}_k - \bar{\mathbf{y}}_c)^T\right) \quad (48)$$

Advantageously, the Incremental 2DLDA according to the example embodiments inherits the advantages of the 2DLDA and the Incremental LDA (ILDA). Based on the Incremental 2DLDA, the small sample size problem can be avoided as well, since the within-class scatter matrix is not singular. It does not have to redo the entire training when a new sample is added. In addition, while the present formulation can provide an exact solution, the existing ILDA gives only approximate updates and thus it may suffer from numerical instability. The experimental results show that the Incremental 2DLDA can produce the same performance as with batch 2DLDA while saving more computation time and memory than the latter, as discussed in detail below. As will be understood by a person skilled in the relevant art, in order to do incremental learning, both ILDA and Incremental 2DLDA need to maintain one between-class scatter matrix and one within-class scatter matrix of every class. However, for Incremental 2DLDA, the size of the between-class scatter matrix and within-class scatter matrix can be much smaller than the ones of ILDA, so Incremental 2DLDA can overcome the limitations of the number of the classes or chunk size in ILDA.

In the conventional 2DLDA algorithm, most of the computation occurs at the steps of computing the mean of each class, overall mean, eigenvalues and eigenvectors, within-class scatter matrix and between-class scatter matrix, and the computation is based on the number of training samples. The computation times may become significant when the number of samples is very large (e.g. in the thousands or tens of thousands). In contrast, the Incremental 2DLDA algorithm according to the example embodiments has most of the computation occurring at the step of updating within-class scatter matrix and between classes scatter matrix, and the computation is based on the number of the classes. The analysis of the computational complexity for both 2DLDA and Incremental 2DLDA algorithm is listed in Table 1. It can be seen from Table 1 that the first part (eigen computation) of the computation complexity of 2DLDA and incremental 2DLDA are the same ($O(l^3)$). The second part (computational time for computing between-class and within-class scatter matrices) of the complexity of the conventional 2DLDA (i.e. $O(nrw^2)$) increases with the increase of the training samples size n, while the second part of the computational requirement of the Incremental 2DLDA according to the example embodiment (i.e. $O(Nrw^2)$) only depends on the number of classes N. For a database with q samples of each subject, the time cost of the Incremental 2DLDA is about $1/q$-th compared to 2DLDA. In general, in order to achieve higher accuracy in face recognition, $q>1$. Thus, the Incremental 2DLDA according to the example embodiments can be computationally more efficient than 2DLDA.

Another advantage of Incremental 2DLDA over the conventional 2DLDA and ILDA is that it requires much less memory than the latter two. The 2DLDA algorithm needs to load all training samples when a new image is added, requiring memory of $(n\times r\times w\times 2)$ bytes (with float type data) while the Incremental 2DLDA algorithm only incurs memory size of $(r\times w)$ bytes for loading one new image and $(N+3)\times(r\times w\times 2)$ bytes for remembering the mean of every classes, overall mean of the training samples, within-class matrix and between-class matrix. Thus, the conventional 2DLDA uses about (n/N) times more memory than Incremental 2DLDA

Also, in ILDA, the image is represented as a 1D vector with size $r\times w$, and the size of the covariance matrix is $(r\times w)^2$. In order to do incremental learning, both ILDA and Incremental 2DLDA need to maintain one between-class scatter matrix and one within-class scatter matrix of every class. For an application with N classes, the memory requirement in ILDA to maintain $\mathbf{S}_b$ and $\mathbf{S}_w$ is $N\times 2\times(r\times w)^2\times 2$ bytes and the memory requirement in Incremental 2DLDA to maintain $\mathbf{S}_b$ and $\mathbf{S}_w$ is $N\times 2\times(r\times w)\times 2$. Thus, the memory cost of the ILDA is about $(r\times w)$ times of Incremental 2DLDA. Further, Incremental 2DLDA can solve the limitation of the number of classes or limitation of the chunk size encountered in ILDA. The experimental results using the method of the example embodiments show that it is possible use large chunk size to update the eigenspace quickly.

Table. 1: The computation complexity of the 2DLDA and Incremental 2DLDA

Algorithm	2DLDA	Incremental 2DLDA
Computation Complexity	$O(l^3+nrw^2)$, $l=\max(r,c)$	$O(l^3+Nrw^2)$, $l=\max(r,w)$

The inventors have evaluated the method of the example embodiments in a face recognition application using the publicly available Olivetti Research Laboratory (ORL) and Extended Multi Modal Verification for Teleservices and Security applications (XM2VTS) databases. The ORL database contains ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, with varying lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). The size of the face image is about 112 × 92 pixels. The XM2VTS database contains eight (CDS 0001 and CDS 0006) different frontal images of each of 295 subjects. The images in CDS0001 were taken at the beginning of the head rotation shot. The images in CDS0006 were taken from the middle of the head rotation shot when the subjects had returned their heads to the middle. They are different from those contained in CDS001. The size of each normalized face image is about 101 × 161 pixels.

For the ORL database, one of the ten samples of each subject is randomly selected to form the testing set, and the remaining nine samples of each subject form the training set. At the beginning, 8 images of the respective 15 subjects are used (about 30% of the size of the database) to construct an initial bilateral two-dimensional linear discriminant eigenspace. The testing set comprises 40 images (one image from each subject). Subsequently, the remaining training samples are added to update the discriminant subspace. Once a discriminant subspace is updated, the samples of the testing set are used to test the performance of the updated discriminant model. For face classification, $\mathbf{W}_l$ and $\mathbf{W}_r$ in algorithm B2DLDA are applied to a probe image to obtain the features $\mathbf{B}_l$ and $\mathbf{B}_r$. The $\mathbf{B}_l$ and $\mathbf{B}_r$ are converted to 1D vectors respectively. PCA is adopted to reduce the dimension of the concatenated vectors of $\{\mathbf{B}_l, \mathbf{B}_r\}$ and a nearest neighbour classifier is employed to get the final recognition result. For the XM2VTS database, at the beginning, 6 images of the respective 160 subjects (about 40% of the size of the database) are used to construct an initial bilateral two-dimensional linear discriminant eigenspace. The testing set comprises 295 images (one image from each subject).

In the evaluation, the inventors have implemented the Incremental 2DLDA in Matlab on a personal computer (PC) having a 2.66GHz central processing unit (CPU) and 4GB of Random Access Memory (RAM). In order to show the advantage of Incremental 2DLDA in terms of execution time and memory cost over the batch 2DLDA, both algorithms are run on the same database. Figures 1(a) and 1(b) show graphs of face recognition accuracy against the number of the incremental learning stages on the ORL and XM2VTS databases respectively when new samples are sequentially added to a face recognition system according to an example embodiment. In Figures 1(a) and 1(b), the number of the incremental learning stages is equivalent to the number of samples that has been added to the learning samples for the incremental models. As can be seen from Figs. 1 (a) and 1 (b), the accuracy is improved as the number of the samples is increased.

Figure 2(a) and 2(b) show graphs of classification accuracy against number of new samples using chunk Incremental 2DLDA on the ORL and XM2VTS databases respectively according to an example embodiment. In Figure 2(a), the chunk size is 40, while in Figure 2(b), the chunk size 300. As discussed above, the chunk size is limited in the conventional ILDA because it needs to maintain a within-class scatter matrix and a between-class scatter matrix which are of size $(r \times w)^2$. On

the other hand, the chunk size in the example embodiment can be relatively large because the memory cost for maintain the between-class and within-class scatter matrices are lower.

Figures 3(a) and 3(b) show graphs comparing the execution times between batch 2DLDA and sequential Incremental 2DLDA against the number of learning stages using ORL and XM2VTS databases respectively according to an example embodiment. In Figures 3(a) and 3(b), one new sample is added at each stage. At each stage, the batch 2DLDA learning is executed using the training samples composed by the initial training set and the new sample added at that stage. It can be seen from Figures 3(a) and 3(b) that the execution time based on the sequential Incremental 2DLDA method of the example embodiment (as represented by lines 302 and 312) is much less than that for batch 2DLDA (as represented by lines 304 and 314). In addition, the execution time in the method of the example embodiment remains nearly the same for subsequent new samples, while the execution time for the batch 2DLDA increases rapidly with the increase in the number of the new samples.

Figures 4(a) and 4(b) shows graphs comparing the execution times between batch 2DLDA and chunk Incremental 2DLDA of various chunk sizes against the number of new samples using ORL and XM2VTS databases respectively according to an example embodiment. Similar to Figures 3(a) and 3(b), at each step corresponding to an Incremental 2DLDA step, the batch 2DLDA learning is executed using the training samples composed by the initial training set and the new sample set added at that step. It can be seen from Figures 4(a) and 4(b), the larger the chunk size is, the shorter the execution time using the chunk Incremental 2DLDA method of the example embodiment.

Figures 5(a) and 5(b) show graphs comparing the memory costs of sequential ILDA and sequential Incremental 2DLDA using ORL and XM2VTS databases respectively according to an example embodiment. The samples are resized to about 56×46 pixels in the example embodiment. Figures 6(a) and 6(b) show graphs comparing the memory costs of chunk ILDA and chunk Incremental 2DLDA using ORL and XM2VTS databases respectively according to an example embodiment. In Figures 6(a), the chunk size is 80, while in Figure 6(b), the chunk size is 300. From Figures 5(a)-(b) and 6(a)-(b), it can be seen that the Incremental 2DLDA method of the example embodiment, as represented by lines 502, 512, 602 and 612 uses significantly less memory than the conventional ILDA, as represented by lines 504, 514, 604 and 614.

The comparison of the memory cost for ILDA and Incremental 2DLDA is shown in Table 2. It can be seen from Table 2 that the Incremental 2DLDA method can consumes significantly less memory than ILDA. These results verify the analysis that the Iincremental 2DLDA can overcome the limitation of the number of the classes or the chunk size that is encountered in ILDA. When the number of classes is large or the chunk size is too large, the memory cost of the ILDA can become very high. On the other hand, in the Incremental 2DLDA method of the example embodiment, the memory cost is very low.

Table 2. The memory cost (in Mega bytes) of the ILDA and the INCREMENTAL 2DLDA for the ORL and the XM2VTS databases

Database	Image size	Initial training set	Initial memory cost (ILDA)	Initial memory cost (Incremental 2DLDA)	Memory cost for complete training (ILDA)	Memory cost for complete training (Incremental 2DLDA)
ORL (40 subjects, 10 images/ each subject)	56×48	15 subjects, 8 images/subject	206.7	0.07	536.09	0.20
XM2VTS (295 subjects, 8 images/ each subject)	50×80	160 subjects, 6 images/subject	5041.1	1.24	9771.0	2.11

The method and system of the example embodiments have also been applied to face age recognition. In such application, the Incremental 2DLDA method according to the example embodiments can combine the tasks of active learning and sequential learning for classifying a face image into one of several age categories. The following description provides an example implementation in face age recognition.

Pool-based active learning

It will be appreciated by a person skilled in the relevant art that pool-based learning is a setup for active learning. In active learning, there is a pool of unlabeled points U and a pool of labeled points L. The goal is to iteratively pick the most informative points in U for labeling, obtain the labels from some oracle or teacher, add the labeled points to L, incrementally update the classifier using the newly added samples from U, and then iterate and see how fast the classifier converge to the final solution. An active learner usually has three components: (1) a classifier trained on the current set of labeled data; (2) A querying function that decides which instance in U to query at the next round; and (3) An updated classifier after each query. In the example embodiment, a classifier is trained using a small number of randomly selected labeled examples called the seed set. In addition, the process is repeated until either the evaluation rate arrives at a value or U is an empty set or until the oracle is no longer able to provide labels. During each round of active learning, *n* points are selected for labeling. This is herein referred to as the batch size. One of the main differences in active learners is how one determines whether a point in U will be informative if labeled.

Uncertainty

In the example embodiment, an uncertainty sampling approach is used to perform active learning. Uncertainty sampling typically works by assigning an uncertainty score to each point in U and picking the *n* points with the highest uncertainty scores. These uncertainty scores are based on the predictions of the classifier currently trained on L. The uncertainty sampling method usually relies on probability estimates of class membership for all the examples in the active pool. Margin-based classifiers, e.g. SVM, have been used as a notion of uncertainty in prior art methods where class membership probabilities of the unlabeled examples are first estimated using the distance from the hyperplane for classifiers. The uncertatinty score is inversely proportional to the absolute value of the distance from the hyperplane, where points closer to the hyperplane are more uncertain.

A main difference between an active learner and a passive learner is the querying element, i.e. to choose the next unlabeled instance to query. In the example embodiment, the output of the incremental 2DLDA classifier is the distance of the query sample to the class instead of probabilities. Incremental 2DLDA with nearest neighbour classifier according to the example embodiment may be one of the simplest classification schemes that classifies a data point based on the labels of its neighbors, and can naturally handle multi-class problems. However, incremental 2DLDA does not admit a natural notion of uncertainty in classification, and hence, it is unclear how to estimate the probability of misclassification for a given data point. In the example embodiment, the distance in the subspace is chosen as the uncertainty measurement.

Further, in the example embodiment, the data face that yields the largest distance to the nearest neighbour is selected and the sample selection method is referred to as Furthest Nearest Neighbour (FNN). This means all the unlabelled or uncertain data are tested and for each data, its nearest neighbour is found using the incremental 2DLDA projections as described above and the one which has the furthest nearest neighbour is chosen. This data is deemed to have the highest probability of uncertainty. If the furthest nearest neighbour turns out to be incorrectly classified, this may imply a significant step forward in learning. If it is assumed that the nearer a sample is to a data the higher the probability that the example is classified correctly, then one that is furthest away will have the least probability of being correctly classified. It is desirable to learn this sample.

Figure 7 shows a diagram illustrating the Furthest Nearest Neighbour method according to an example embodiment. Assume there are two classes of samples, marked as "o" and "×" and their feature space distributions are shown in Figure 7. In the example embodiment, "Δ"s represent four unlabeled samples that have been projected to the subspace. The nearest neighbours (A, B, C and D) of the four samples are connected with them respectively. Based on the approach described above, "A" is the first sample to be selected by the FNN selection method because it is the furthest nearest neighbour.

The new classifier is incrementally learned using the added samples, and uncertainty scores are produced for the unlabeled data in the pool. Figure 8 shows a flow chart 800 illustrating a method for active learning according to an example embodiment. At the start, a seed set 802, an evaluation set 804, a pool 806 and a 2DLDA classifier 808 are provided. The samples in the seed set 802 and the

evaluation set 804 are labelled while those in the pool data 806 are initially unlabelled. At step 810, the classifier 808 is evaluated using the evaluation set 804. At step 812, the pool 806 is checked to determine whether it is empty. If it is not, at step, 814, the 2DLDA is applied to the unlabelled data in the pool 806. At step 816, the samples in the pool 806 are sorted according to their distances to the respective nearest neighbours. At step 818, the samples with the furthest distances are labelled by the user. At step 820, the classifier 808 is updated using new labelled samples in the pool 806. At step 822, the labelled samples are removed from the pool 806. On the other hand, if it is determined at the checking step 812 that the pool 806 is empty, at step 824, the algorithm stops.

The inventors have evaluated the method of the example embodiments using the Face and Gesture Recognition Research Network (FG-NET) and MORPHOLOGY (Morph) databases. The FG-NET and Morph databases have been made available for research in areas related to age-progression. The FG-NET database contains 1,002 face images from 82 subjects, with approximately 10 images per subject. In the Morph database, there are 1,724 face images from 515 subjects. Each subject has about 3 aging images. In the example embodiment, four age groups are defined as: children, teen age, adult and senior adult. The age ranges of the four groups in the example embodiment are 0-11, 12-21, 22-60, and 61 and above respectively. In the example embodiment, a database which includes the face images from both the FG-NET and Morph aging databases is used since it is preferably to have a database which has enough face images for each age range mentioned above. The face images are manually grouped into the four classes defined above according to their ground truth, giving a total of 2726 images. Figures 9(a)-9(d) show sample images of the four age groups respectively.

For the unlabeled face data in the pool 806 (Figure 8), a database is collected from three sources: (1) a lab using a web camera; (2) frontal face images in public databases including the Facial Recognition Technology (FERET), the Pose, Expression, Accessories, and Lighting (PEAL) and the Pose, Illumination, and Expression (PIE) databases; and (3) the Internet. There are total of 4000 face images of 1713 persons. In addition, in the example embodiment, a face detector is used to detect the face, and all faces are then geometrically normalized to 88×88 image. Figure 10 shows sample unlabeled images in the pool.

In the example embodiment, an initial classifier is trained using the images from the FG-NET database, and the face images in Morph are used as query images. Half of the FG-NET and Morph database respectively is randomly selected for the seed set. The remainder half is used as the evaluation set. For each round of the active learning, 4 samples from the unlabeled pool with the furthest nearest neighours which are at the top of the pool are selected and labelled by the user. They are then added to the training set and the 2DLDA classifier is updated using the newly added samples.

Figure 11 shows a graph of classification accuracy versus the number of the selected samples according to an example embodiment. As can be seen from line 1102, the error rate can be much lower when only a subset of U has been labeled as opposed to when all of U has been labeled. This phenomenon has been observed in other works on active learning as well. Thus, stopping the labeling process early can

be very useful in reducing overall classification error. One possible reason is that stopping the process early may help to avoid outliers in the training set, i.e. the error rate may increase when additional points are added, for example, if noisy or outlying points are added to L.

In addition, the performance of the method according to the example embodiments is compared with the approach where the query sample is randomly selected, as represented by line 1104 in Figure 11. As can be seen from Figure 11, the method of the example embodiments converges much faster than the random selection approach.

As a further comparison, the reduction in the number of training examples required for FNN to obtain a similar accuracy as the random selection approach is quantified. From Figure 11, for each round of active learning, the number of rounds required to achieve similar accuracy using either method is determined by fixing a value at Y-axis. Table 3 shows the reduction of the number of rounds needed for the FNN and random selection for getting the similar accuracy. As can be seen from Table 3, FNN selection can obtain similar accuracy with random selection using on average 72.5% fewer samples.

Tabel 3

FNN selection	Random selection	Training examples reduction (%)
10	95	89.5
25	150	83.3
140	310	54.8
190	600	68.3
270	810	66.7
		72.52 (average)

Figure 12 shows a flow chart 1200 illustrating a method for updating a 2 dimensional linear discriminant analysis (2DLDA) classifier engine for feature recognition according to an example embodiment. At step 1202, one or more sample images are provided to the classifier engine, the classifier engine comprising a plurality of classes derived from a plurality of training images and a mean matrix of all images. At step 1204, the mean matrix of all images is updated based on the sample images. At step 1206, a between-class scatter matrix is updated based on the sample images. At step 1208, a within-class scatter matrix is updated based on the sample images.

Figure 14 shows a flow chart 1400 illustrating a method for selecting samples from a pool comprising a plurality of unlabeled samples for active learning, according to an example embodiment. At step 1402, a 2 dimensional linear discriminant analysis (2DLDA) classifier engine is applied to the unlabeled samples. At step 1404, the unlabeled samples in the pool are sorted according to their distances to a respective nearest neighbour. At step 1406, the sample with the furthest nearest neighbour is selected for labeling. At step 1408, the 2DLDA classifier engine is updated based on the labeled sample.

The method and system of the example embodiment can be implemented on a computer system 1300, schematically shown in Figure 13. It may be implemented as software, such as a computer program being executed within the computer system 1300, and instructing the computer system 1300 to conduct the method of the example embodiment.

The computer system 1300 comprises a computer module 1302, input modules such as a keyboard 1304 and mouse 1306 and a plurality of output devices such as a display 1308, and printer 1310.

The computer module 1302 is connected to a computer network 1312 via a suitable transceiver device 1314, to enable access to e.g. the Internet or other network systems such as Local Area Network (LAN) or Wide Area Network (WAN).

The computer module 1302 in the example includes a processor 1318, a Random Access Memory (RAM) 1320 and a Read Only Memory (ROM) 1322. The computer module 1302 also includes a number of Input/Output (I/O) interfaces, for example I/O interface 1324 to the display 1308, and I/O interface 1326 to the keyboard 1304.

The components of the computer module 1302 typically communicate via an interconnected bus 1328 and in a manner known to the person skilled in the relevant art.

The application program is typically supplied to the user of the computer system 1300 encoded on a data storage medium such as a CD-ROM or flash memory carrier and read utilising a corresponding data storage medium drive of a data storage device 1330. The application program is read and controlled in its execution by the processor 1318. Intermediate storage of program data maybe accomplished using RAM 1320.

It will be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present invention as shown in the specific embodiments without departing from the spirit or scope of the invention as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

CLAIMS

1. A method for updating a 2 dimensional linear discriminant analysis (2DLDA) classifier engine for feature recognition, the method comprising the steps of:

providing one or more sample images to the classifier engine, the classifier engine comprising a plurality of classes derived from a plurality of training images and a mean matrix of all images;

updating the mean matrix of all images based on the sample images;

updating a between-class scatter matrix based on the sample images; and

updating a within-class scatter matrix based on the sample images.

2. The method as claimed in claim 1, wherein updating the between-class scatter matrix comprises updating a mean matrix of each class to which at least one of the sample images belongs prior to updating the between-class scatter matrix.

3. The method as claimed in claims 1 or 2, wherein updating the within-class scatter matrix comprises updating a mean matrix of each class to which at least one of the sample images belongs prior to updating the within-class scatter matrix.

4. A 2 dimensional linear discriminant analysis (2DLDA) classifier engine for feature recognition, the classifier engine comprising:

a plurality of classes derived from a plurality of training images;

a mean matrix of all images;

means for receiving one or more sample images;

means for updating the mean matrix of all images based on the sample images;

means for updating a between-class scatter matrix based on the sample images; and

means for updating a within-class scatter matrix based on the sample images.

5. A computer storage medium having stored thereon computer code means for instructing a computing device to execute a method for updating a 2 dimensional linear discriminant analysis (2DLDA) classifier engine for feature recognition, the method comprising the steps of:

providing one or more sample images to the classifier engine, the classifier engine comprising a plurality of classes derived from a plurality of training images and a mean matrix of all images;

updating the mean matrix of all images based on the sample images; and

updating a between-class scatter matrix based on the sample images.