Incremental Constrained Discriminant Component Analysis

Amin Allahyar[1] and Hadi Sadoghi Yazdi[1,2]

[1]*Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran*
[2]*Center of Excellence on Soft Computing and Intelligent Information Processing, Ferdowsi University of Mashhad*
Amin.Allahyar@stu.um.ac.ir, h-sadoghi@um.ac.ir

Abstract

Recently, a constrained Linear Discriminant Analysis (LDA) algorithm is introduced and gained popularity. However, this algorithm is not applicable in the environment with large amount of data points or when the data point arrive in a sequential manner. In this paper, we aim to propose an incremental version of this algorithm called Incremental Constrained Discriminant Component Analysis (ICDCA) to reduce the computational cost of this algorithm in large datasets. The ICDCA updates the within class scatter matrix and between class scatter matrix instead of calculating it from scratch. This change significantly reduces the computational cost of feature extraction process while keep the accuracy of such features as close as possible to offline version of this algorithm. In the end the effectiveness of ICDCA is compared to other recently proposed incremental LDA. To ensure the reliability of these experiments, they are repeated with several UC I data set. In these comparisons, advantage of ICDCA in the accuracy and speed is demonstrated.

Keywords: *Feature Extraction, Linear Discriminant Analysis, Large Datasets, Incremental Learning, Scatter Matrix Updating*

1. Introduction

The artificial intelligence field gained quite popularity in the recent decades. It has a very broad application in the real world problems. It fundamentally aimed to give the computers the ability to learn. Regularly, this is done by formulating it in an optimization problem. These algorithms are categorized as machine learning problems. The machine learning consist of two main field of research. The first category is clustering algorithm which is basically and unsupervised approach to learn. It tries to generate groups from the given data point. The data points in these groups should be as similar as possible while has a considerable amount of dissimilarity with data points in the other groups. The second category is classification. The aim in classification is to provide a discriminator function using the given training samples. This function should be capable of categorizing the unseen samples in the correct category. However, because the computational cost, these algorithms cannot operate in the high dimensional space. The solution to this problem is Dimension Reduction (DR).

The main aim in dimension reduction is to recover the intrinsic representations of data located in the high dimensional space and reveal it in a lower dimensional space. Although the Linear Discriminant Analysis (LDA) concept is quite old subject in dimension reduction problem, it remained popular until today because its simplicity and effectiveness. Basically, the LDA algorithm searches for a projection matrix which transfer the data in high dimensional space to the lower dimension. This projection should be done such that the data in the same class remain close while the data in different class has considerable amount of

distance from each other. The effect of this projection is demonstrated in Figure 1. The LDA has found many application in machine learning algorithms including Gene Expression Data [?], Image Processing [?], Face Recognition [?] and *etc.*

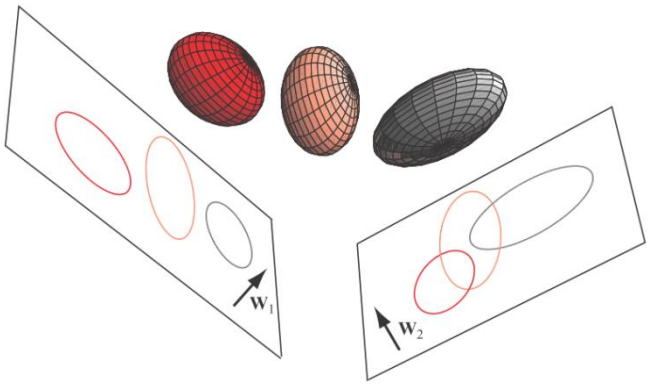


Figure 1. The projection vector and its effect when the right direction is selected [1]

However, the LDA algorithm suffers from high computational cost because it has a cubic complexity with dimension of data points. This is because it requires the inverse of scatter matrix calculation and further the Eigen-vector analysis of multiplication of both scatter matrixes. This family of LDA is regularly called *Offline LDA*. However, in many cases especially in real world applications, it is hard to store entire dataset (i.e. storage limit) and perform LDA (computation limit) [2]. Furthermore in various situations, data arrive continuously and form an infinite data stream. A rudimentary approach is to add new data to the old dataset and find the discriminating direction in the new updated dataset. It is evident that such approach only works if infinite amount of memory and computation power be at hand. Even if that's the case the system is actually discard the knowledge acquired from old dataset and learns it from scratch, which is not ideal [3]. Incremental LDA was introduced to mend such problems.

In the recent years, many incremental LDA algorithms proposed in the literature. Because of the difficulty in incremental solving of standard LDA, some authors proposed algorithms to incrementally approximate the solution [4, 5] or used time consuming approaches to reach exact solutions [3]. Recently the close relationship of LDA and Multivariate Linear Regression (MLR) is demonstrated by Ye [6]. Using this fact, instead of solving the high order $O(d^3)$ eigenvector problem for each iteration, where d is the dimension of data points, Liu proposed a method to incrementally solve MLR problem with order $O(min(n,d) \times d)$ where n indicates number of data points [2]. An extension of Liu's method is investigated by Wang [7] with the same order of complexity. In both of these algorithms, the whole dataset is needed to calculate the solution at every step. Unfortunately in the incremental learning problems, number of data points n is assuming to be infinite [3]. Furthermore LDA is regularly used in face recognition [8-10], microarray gene expression data analysis [11] and information retrieval [12] where number of dimensions is very large. In such situations, the complexities of these algorithms make them impractical. At last it is known that MLR will deviate from global optimal solution when number of classes m is higher than 2. This problem is demonstrated By Ye in (section 4 of his paper) [6] and Hasti, *et al.*, (Page 105 of his book) [13]. It should be noted that the standard LDA problem is a non-

convex problem, so there is no close form solution for this problem. Hence some authors suggested methods where the cost function is transformed into simpler and yet inexact convex problem and solved. These conversions and it's diversion from optimal solution is extensively investigated by Wang in [14].

Another closely related problem which recently got considerable amount of attention is learning the projection matrix from constrains which we call them *Constrained LDA* [15]. The solution to this problem has a wide variety of applications in *Semi-Supervised Learning* [16, 17]. There are two types of constrains in this problem, *Must-Link* and *Cannot-Link*. The must-link constrain between two data point indicates that they should be close as possible. On the other hand, cannot-link constrain indicates that the projection matrix should keep these data away from each other [18]. It is clear that amount of available domain knowledge exists in constrains is weaker compared to a situation where label of data point is at hand. The fact can be assured as labels of data can be converted to constrain between data points but the vice versa is not always possible. To find the best possible projection matrix from these constrains, the task is formulated as constrain optimization problem. The resulting cost function is very close to standard LDA problem. However to our knowledge, in contrast to standard incremental LDA where widely studied in the past, all of the studies in constrain LDA are done in situation when data and their corresponding constrain is given in advance. Therefore these algorithms are not applicable in the problems with stream data and constrains.

In this paper we focus on online constrain discriminant analysis, nevertheless because of close relationship and similarity of cost functions in standard LDA and constrained LDA; it is straightforward to use our algorithm on situation where labels of each data point are available. Thus we aim to formulate these two problem with similar notation as much as possible. We show that similar to incremental PCA, the incremental learning of LDA can be seen as small perturbation of cost function matrix. Then inspired by work of Liu [19] we propose our method to incrementally solve the LDA problem. Beside the fact that our proposed algorithm is the first solution to online constrain LDA, it has smaller order of complexity from all other incremental algorithm for standard LDA problem, so it can be used in such problems. In addition, because of the special characteristics of proposed method, it can not only used when new data points is added to the problem, it can be used when some data is deleted from the problem as well. At last it is applicable when some data changes over time, therefore it support the *Concept Drift* properties too.

This paper is organized as follows: In Section 2 we will discuss about preliminaries in discriminant analysis and its close problem constrained LDA. Also we describe the ordinary routines for solving such problems. In Section 3 we discuss the related work in incremental LDA. Furthermore because our proposed algorithm is actually an incremental version of a recently proposed batch LDA algorithm [15], we also express their solving routine in this section. Section 4 is dedicated to our proposed algorithm. In Section 5 we express the order of complexity corresponding to proposed algorithm and its comparison with other incremental LDA methods. Experimental result will be demonstrated in Section 6 and conclusion is given in Section 7.

2. Preliminary

First we define our notation in this paper. Scripted letters such as $\mathcal{C}$ and $\mathcal{M}$ represent sets. Capital letters like X and W are matrixes while bold lower case letters show column vectors, *e.g.*, $\boldsymbol{x}$ and $\boldsymbol{u}$. Lower case letters indicate scalars, *e.g.*, n and d. Similar

to popular notation we use subscripts to index elements in matrixes or vectors. For example x_i is i-th element of vector $\boldsymbol{x}$ and $\boldsymbol{w}_j$ is the j-th column vector in matrix W. Furthermore we will show new updated objects with an ‘ sign. For example $\boldsymbol{x}'$ indicates the updated version of $\boldsymbol{x}$ vector. The vector norm $\|.\|$ is the l_2 norm so by definition $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$. Let $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\} \in \mathbb{R}^{d\times n}$ be given d dimensional samples where n can be equal to infinity in the incremental learning. Also there are m classes available so the whole dataset is divided into m set where $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_m\}$ and depend on their labels, each data point belong to one of these classes. The aim in LDA problem is to find an optimal projection matrix $W^* \in \mathbb{R}^{r\times d}$ where r is the desired dimensionality of data after projection.

As discussed in previous section, in contrary to constrained LDA, the standard LDA has been extensively studied in the past. Fortunately there exists a close relationship between these two families of feature extraction and one can easily use a defined cost function to solve both family of LDA. We first consider unconstrained environment where label of each data point is given. Each sample $\boldsymbol{x}_i$ has a corresponding class label l_i. Then the between-class scatter matrix S_b and within-class scatter matrix S_w are determined by (1):

(1)

$$S_b = \sum_{k=1}^{m} n_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T$$

$$S_w = \sum_{k=1}^{m} \sum_{\boldsymbol{x}_i \in \mathcal{X}_k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T$$

Where n_k indicate number of samples that belongs to class $\mathcal{X}_k$. In this formula, $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}$ indicate mean vector of class k and global mean vector of whole dataset respectively. These two matrixes can be intuitively described as follows. The S_b matrix will measure how far mean vectors corresponding to each class is scattered around input space. The dispersion of class means is quantized by sum of the distances between each class mean vector and the global mean vector. Likewise the S_w matrix measures the spreading of data points around its class mean vector. Mathematically S_w is sum of covariance matrixes calculated from data points belonging to each class. By these expressions it is clear that the LDA will have a suitable result if the distribution of each class in input space follows a uni-modal distribution [20].

In the constrained linear discriminant analysis, instead of labels, the relationship between two pair of data point is given as domain knowledge. To increase the effectiveness of constrains in this environment, it is a common approach to consider closure of the must-link constrains [21]. In this technique, the must-link constrains build a connected component in such way that no data point in same group is bounded with a cannot-link constrain [22]. These groups are called *discriminative sets* and will be represented by $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_c\}$ where regularly $c \gg m$.

Similar to labeled environment the between and within scatter matrixes can be defined as (2) [23]:

(2)

$$S_b = \sum_{k=1}^{c} n_{\mathcal{D}_k}(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T$$

$$S_w = \sum_{k=1}^{c} \sum_{\boldsymbol{x}_i \in \mathcal{D}_k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T$$

From this point, both constrained and standard LDA problems can be solved in a similar way. Using these definitions, the standard LDA cost function proposed by Fisher [24] is as follows.

(3)

$$J_f(W) = \frac{tr(W^T S_b W)}{tr(W^T S_w W)}$$

We are interested in W such that given S_b and S_w the cost function $J_f(W)$ acquires its maximum value. In the other word, we are looking for a solution to following optimization algorithm.

$$W^* = \underset{W}{\operatorname{argmax}}\, J_f(W)$$

This optimization problem is typically non-convex. More generally there is no close form solution for general trace ratio problem, Therefore these problems are regularly altered into the simpler yet inexact ratio trace problem which is convex and has a close form solution [14]. This transformation can be formulated as (4):

(4)

$$W^* = \underset{W}{\operatorname{argmax}}\, tr(\frac{W^T S_b W}{W^T S_w W}) = \underset{W}{\operatorname{argmax}}\, \frac{|W^T S_b W|}{|W^T S_w W|}$$

The cost function in (4) can be easily solved directly by finding the solution to generalized eigenvector problem of the form (5) [20]:

(5)

$$S_b U = S_w U \Lambda$$

The eigenvectors corresponding to largest eigenvalues are the transformation matrix that maximize the given cost function in (4). But it should be noted that rank of S_b is at most $m - 1$ [1], where m indicate number of classes. So only $m - 1$ of these vectors has discriminating information [25]. Furthermore if S_w be non-singular, formula (4) can be rewritten as standard eigenvector problem by multiplying S_w^{-1} from left as follows.

$$S_b U = S_w U \Lambda$$

$$S_w^{-1} S_b U = S_w^{-1} S_w U \Lambda = I U \Lambda$$

$$S_w^{-1} S_b U = U \Lambda$$

The singularity of S_w is the most known difficulty in LDA and is widely studied in the literatures [26-28]. For comprehensive discussion on these complications please refer to [29].

3. Related Work

As expressed before, an incremental learning algorithm should preserve the past knowledge while incorporating new knowledge every time new information is given without keeping large volume of data points in the storage. To accomplish this, many incremental algorithms has been introduced in the past. Approximation of LDA solution

was investigated by some authors [4, 30]. In order to calculate approximate projection matrix in Rubner, *et al.*, algorithm, a neural network with two layer is used where each layer is actually a PCA network [30]. Mao's algorithm used a similar topology and aims to simultaneously diagonalize S_w and S_t [4]. In the end, output of such a network would be $S_w^{-1}S_t$. This algorithm suffers from slow convergence especially when number of dimension increases. In addition, the algorithm assumes that total mean $\boldsymbol{\mu}$ and class mean $\boldsymbol{\mu}_k$ $\forall k \in \{1,2,..m\}$ is given in advance. Application of neural network in incremental LDA problem is also perused by Chatterjee et al. [5]. The main goal in this algorithm is calculation of square inverse of covariance or correlation matrix [31].

Pang *et al*. suggested a way for exact updating of the within-class scatter matrix but does not solve the time consuming steps of calculating the inverse of S_w and the Eigen-space model resulted by updating of scatter matrixes [3]. In addition, through his algorithm, the between-class scatter matrix should be calculated from scratch with addition of each data point. Despite the computational cost for Eigen-space updating and inverse of S_w and S_b calculation, it was a good idea to update scatter matrixes instead of reconstruction from scratch [31]. Our proposed method is related to this algorithm as we also update the exact scatter matrixes. However we update both S_w and S_b. Furthermore in contrary to Pang's algorithm we do not assume that the data is zero mean and the mean vector impact on updating process taken into account implicitly [3]. Another trick to escape from singularity of S_w is maximizing S_b in projected subspace by QR decomposition method which is introduced by Ye, *et al.*, [32]. The problem with this algorithm is that it might eliminate separability information in first projection of data into subspace [31]. Zhao et al. proposed GSVD-ILDA which is basically an incremental version of LDA/GSVD introduced by Ye et al. [28]. In this algorithm an approximation trick is used in the fast SVD technique to reduce the computational complexity. Hence the computational cost increases if we want to reach a high accuracy solution of the problem.

Kim, *et al.*, suggested a method to update the Eigen system [33, 34]. This is the closest approach to our proposed method as the Eigen system is updated as new sample becomes available. First problem in this algorithm is that the significant eigenvector corresponding to largest eigenvalues for S_b, S_t and $S_t^{-1}S_b$ is stored. So these matrixes will be approximated in every step and may get distance from actual matrixes if number of updating iteration increases. This problem is mentioned by Liu et al [2]. Furthermore as Kim stated, the complexity of updating algorithm is $O\left(\left(rank(S_i) + rank\left(\Delta_{S_i}\right) + 1\right)^3 \times d\right)$ $i \in \{b,t\}$. But in the incremental learning $rank(S_t)$ is close to maximum possible or $rank(S_t) \approx \min(d,n)$. So in the worst case, the complexity of this algorithm will be close to other offline LDA algorithms with $O(\min(n,d)^3)$. Furthermore Kim's only studied the problem when new data point is added. Instead of such method, we update scatter matrixes without any approximation, thus eliminate the problem of large error when number of iteration increase. Then we incrementally calculate Eigen space of cost function. At last the order of our algorithm is $O(\min(r+7,d)^3)$ where regularly $r = m-1$. Therefore, it is very efficient in situations where number of classes is small.

Because the close relationship between our algorithm and an offline LDA method, we briefly describe this algorithm as follows. The relation originates from ICDCA cost function. Previously this cost function introduced in the literature and it's equivalency with standard LDA proved [15, 35]. However it was used for offline or batch learning of LDA. We chose this cost function because it has some appropriate properties which

will help to update Eigen-spaces very fast, efficient and accurate. These properties will be revealed subsequently in the section 4 when we propose our algorithm.

To maintain the integrity, here we provide the equivalency evidence which is proved by Guo, *et al.*, [35]. In (3) the cost function of LDA is defined as:

$$J_f(W) = \frac{tr(W^T S_b W)}{tr(W^T S_w W)}$$

Following Gau, *et al.*, work we are looking for a matrix where maximize this cost function. In the other word:

$$J(W^*) = \underset{W}{\operatorname{argmax}} \; J_f(W) = \lambda^* \tag{6}$$

Where λ^* correspond to maximum possible value for $J(W^*)$. It is given as a corollary by Liu, *et al.*, that following equality holds [36]:

$$J_f(W) = \frac{tr(W^T S_b W)}{tr(W^T S_w W)} = \frac{\sum_{s=1}^{r} \boldsymbol{w}_s^T S_b \boldsymbol{w}_s}{\sum_{s=1}^{r} \boldsymbol{w}_s^T S_w \boldsymbol{w}_s} \tag{7}$$

$$J(W^*) = \underset{W}{\operatorname{argmax}} \; J_f(W) = \underset{\substack{w_i^T w_j = 0 \\ w_i^T w_j = 1}}{\operatorname{argmax}} \frac{\sum_{s=1}^{r} \boldsymbol{w}_s^T S_b \boldsymbol{w}_s}{\sum_{s=1}^{r} \boldsymbol{w}_s^T S_w \boldsymbol{w}_s} = \frac{\sum_{s=1}^{r} \boldsymbol{w}_s^{*T} S_b \boldsymbol{w}_s^*}{\sum_{s=1}^{r} \boldsymbol{w}_s^{*T} S_w \boldsymbol{w}_s^*} = \lambda^* \tag{8}$$

$$i, j = 1,2, \dots, r \; and \; i \neq j$$

Where $\boldsymbol{w}_s$ and $\boldsymbol{w}_s^*$ is s-th column vector of W and W^* respectively and r is desired dimension of data in feature space. Guo showed that if and only if (8) holds, we have the following equality [35]:

$$\sum_{s=1}^{r} \boldsymbol{w}_s^{*T} (S_b - \lambda^* S_w) \boldsymbol{w}_s^* = 0$$

And for any other W we have:

$$\sum_{s=1}^{r} \boldsymbol{w}_s^T (S_b - \lambda^* S_w) \boldsymbol{w}_s \leq 0$$

In addition, under the previous assumption, we have:

$$\lambda < \lambda^* \text{ if and only if } \underset{\substack{\boldsymbol{w}_i^T \boldsymbol{w}_j = 0 \\ \boldsymbol{w}_i^T \boldsymbol{w}_j = 1}}{\operatorname{argmax}}_{W} \sum_{s=1}^{r} \boldsymbol{w}_s^T (S_b - \lambda S_w) \boldsymbol{w}_s > 0$$

$$\lambda > \lambda^* \text{ if and only if } \underset{\substack{\boldsymbol{w}_i^T \boldsymbol{w}_j = 0 \\ \boldsymbol{w}_i^T \boldsymbol{w}_j = 1}}{\operatorname{argmax}}_{W} \sum_{s=1}^{r} \boldsymbol{w}_s^T (S_b - \lambda S_w) \boldsymbol{w}_s < 0$$

Therefore the problem of (3) is converted to (9):

(9)

$$J(W^*) = \underset{\substack{W \\ \boldsymbol{w}_i^T \boldsymbol{w}_j = 0 \\ \boldsymbol{w}_i^T \boldsymbol{w}_j = 1}}{\operatorname{argmax}} \sum_{s=1}^{r} \boldsymbol{w}_s^T (S_b - \lambda S_w) \boldsymbol{w}_s$$

Where λ needs to be determined such that $trace(S_b - \lambda^* S_w) = 0$ [15]. After finding the optimal λ^*, the optimal value W^* can be obtained by selecting the first r eigenvectors corresponding to the largest eigen values of matrix $V = S_b - \lambda^* S_w$. It worth noting that using these theorems Guo proposed a binary search algortihm to find the optimal λ^* in offline LDA. Similar approach is pursued by Xiang, *et al.*, [15] to solve this cost function. Furthermore a Newton's method is proposed by Wang [14] to find λ^*.

In the next section, we first show an efficient exact update for mean vectors and scatter matrixes. The later part made possible by using special definition for scatter matrixes. The equivalency of such definitions for scatter matrixes is also investigated and proved. After that we demonstrate our updating algorithm for Eigen-space which is very fast and efficient, thus make it suitable for incremental learning. It is inspired by pioneer work of Lie, *et al.*, [19]. Another property of proposed algorithm is that it can be used to not only addition of new point, but deletion of old samples is possible. It also can be used when the existing data points move in time, therefore support *concept drift* concept. It should be noted that, many form of cost function for LDA is proposed in the literature [20]. Although we used a specific cost function constructed by $V = S_b - \lambda^* S_w$ any combination of scatter matrixes can be used. In the other word, any combination of $S_i - \lambda S_j$ where $i, j \in \{b, w, t\}$ is valid and can be used as cost function. More generally, as far as the cost function of LDA stays symmetric, it can be incrementally solved by our proposed method. The symmetric requirement is also discussed more clearly in the next section.

4. Proposed Method

In this section we propose our incremental linear discriminative analysis algorithm called *Incremental Constrained Discriminant Component Analysis* (ICDCA). As discussed, our algorithm is inspired by Liu's work [19]. The within scatter matrix is defined as follows.

$$S_w = \sum_{k=1}^{c} \sum_{\boldsymbol{x}_i \in \mathcal{D}_k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T$$

In this formulation, we can easily extract one class. For example assume we want to extract class q. This can be written as follows.

$$S_w = \sum_{\substack{k=1 \\ k \neq q}}^{c} \sum_{\boldsymbol{x}_i \in \mathcal{D}_k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T + \sum_{\boldsymbol{x}_i \in \mathcal{D}_q} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_q\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_q\right)^T$$

This extraction can be further persuaded to extract one data point p. Let the new the new incoming data point $\boldsymbol{x}_p$ belongs to class q. Then the updated version of within scatter matrix can be defined as (10).

(10)

$$S_w = \sum_{\substack{k=1 \\ k \neq q}}^{c} \sum_{\boldsymbol{x}_i \in \mathcal{D}_k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T + \sum_{\substack{\boldsymbol{x}_i \in \mathcal{D}_q \\ i \neq p}} (\boldsymbol{x}_i - \boldsymbol{\mu}_q)(\boldsymbol{x}_i - \boldsymbol{\mu}_q)^T + (\boldsymbol{x}_p - \boldsymbol{\mu}_q)(\boldsymbol{x}_p - \boldsymbol{\mu}_q)^T$$

Therefore with arrival of each data point p from class q, the within scatter matrix can be updated as (11).

(11)

$$S'_w = S_w + (\boldsymbol{x}_p - \boldsymbol{\mu}_q)(\boldsymbol{x}_p - \boldsymbol{\mu}_q)^T$$

Where S'_w is the updated version of within scatter matrix after arrival of new data point. Therefore this scatter matrix does not need to be calculated from scratch. This will greatly reduce the computational cost. Next, inspired by Liu's work we can extract the eigenvectors of new updated within scatter matrix S'_w. Let $\boldsymbol{\varphi}_i$ represents the i-th eigenvetor of within scatter matrix. So the S_w scatter matrix can be calculated as (12).

(12)

$$S_w = \lambda_1 \boldsymbol{\varphi}_1 \boldsymbol{\varphi}_1^T + \lambda_2 \boldsymbol{\varphi}_2 \boldsymbol{\varphi}_2^T + \cdots + \lambda_d \boldsymbol{\varphi}_d \boldsymbol{\varphi}_d^T$$

If we substitute (11) into (12), we have the updated formula for S'_w as (13).

(13)

$$S'_w = \lambda_1 \boldsymbol{\varphi}_1 \boldsymbol{\varphi}_1^T + \lambda_2 \boldsymbol{\varphi}_2 \boldsymbol{\varphi}_2^T + \cdots + \lambda_d \boldsymbol{\varphi}_d \boldsymbol{\varphi}_d^T + (\boldsymbol{x}_p - \boldsymbol{\mu}_q)(\boldsymbol{x}_p - \boldsymbol{\mu}_q)^T$$

Now we can define two matrix for eigen-vectors.

$$G = \left[\sqrt{\lambda_1}\boldsymbol{\varphi}_1 \quad \sqrt{\lambda_2}\boldsymbol{\varphi}_2 \quad \ldots \quad \sqrt{\lambda_r}\boldsymbol{\varphi}_r \quad (\boldsymbol{x}_p - \boldsymbol{\mu}_q)\right] \in \mathbb{R}^{d \times d}$$

$$H^T = \left[\sqrt{\lambda_1}\boldsymbol{\varphi}_1^T \quad \sqrt{\lambda_2}\boldsymbol{\varphi}_2^T \quad \ldots \quad \sqrt{\lambda_r}\boldsymbol{\varphi}_r^T \quad (\boldsymbol{x}_p - \boldsymbol{\mu}_q)^T\right] \in \mathbb{R}^{d \times d}$$

Using these matrixes, we can have the updated version of within scatter matrix.

$$S'_w = GH^T$$

Using the Eigen-vector merging algorithm proposed by Hall et .al [37], we can combine the Eigen vectors of both scatter matrix.

5. Experimental Result

In order to show the effectiveness of our proposed algorithm, we apply ICDCA in the environment where label of data points is available and compared its results with some other recently proposed incremental LDA algorithms including: Kim's [33, 34], Zhao's [31] and Pang's [3] method. Furthermore we measure the speed of each algorithm and represent them in their particular subsection.

5.1. Setup

In this experiment, we used an Intel Quad-Core 2.5 GHZ computer with 4GB RAM on windows 7 64bit. Also Matlab 2012a is used as simulation software. We used four

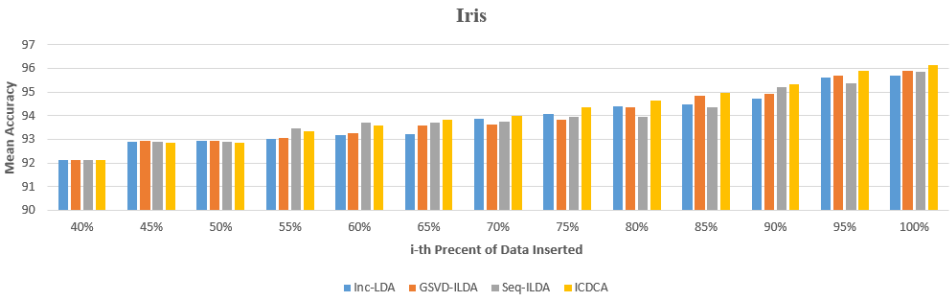
UCI repository[1] data set including: Iris, Protein, Diabetes and Breast-W. In addition, we investigate the accuracy and speed of ICDCA with two high dimensional real world dataset: COIL-20 and ORL. COIL-20 include gray image of 20 objects which is taken from 75 different angles. These samples is reduced to size 32×32. ORL dataset contains gray images of 40 persons. Each person has 10 shots, each with different expressions and facial details. As the source image has dimensionality of 112×92, the input data has 10304 dimension. Properties of these datasets along with generated discriminant set are given in Table 1.

Table 1. Properties of data sets used for experiments. As previously defined, n is number of datapoints, d is data dimension, m indicates number of class, r is the desired number of extracted feature and small and large is number of discriminant set described in 6.1

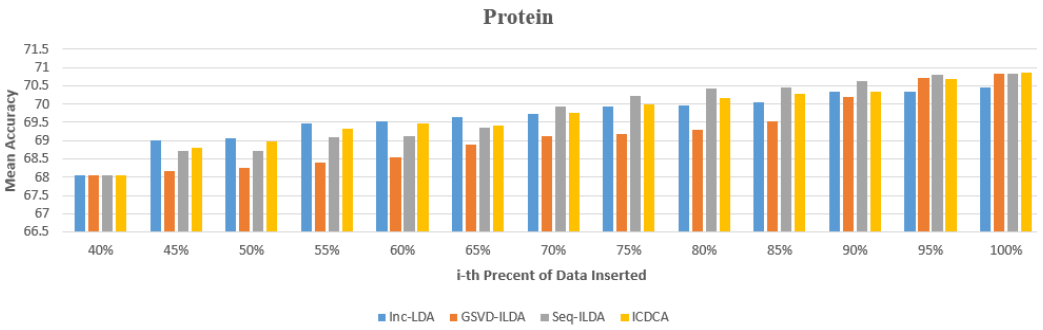
	n	d	m	r
Iris	150	4	3	2
Protein	16	20	6	5
Diabetes	768	8	2	2
Breast-W	683	10	2	2
COIL	1440	256	20	15
ORL	400	10304	40	40

5.2. Experiments on Accuracy

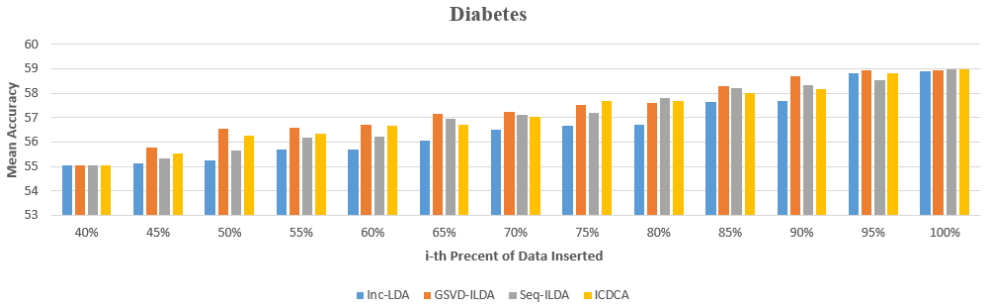
It was discussed in the previous section that to our knowledge there is no online constrained LDA. Thus, we compared the result of ICDCA with offline algorithms. However, ICDCA is also applicable where label of data points is at hand and many algorithms are proposed to solve online LDA problem. It seems reasonable to compare ICDCA with these algorithms in labeled environment. Therefore, in order to show the effectiveness of our proposed algorithm, we apply ICDCA in the environment where label of data points is available and compare its results with some other recently proposed incremental LDA algorithms including: Kim's (Inc-LDA) [33, 34], Zhao's (GSVD-ILDA) [31] and Pang's (Seq-ILDA) [3] method. In this experiment, 40% of data points is randomly selected as initialization and are given to standard LDA algorithm. Then, remaining data points gradually fed to each above algorithms. In each step, accuracy of projected dataset using KNN classifier with K=1 is measured and reported in Figure 2.



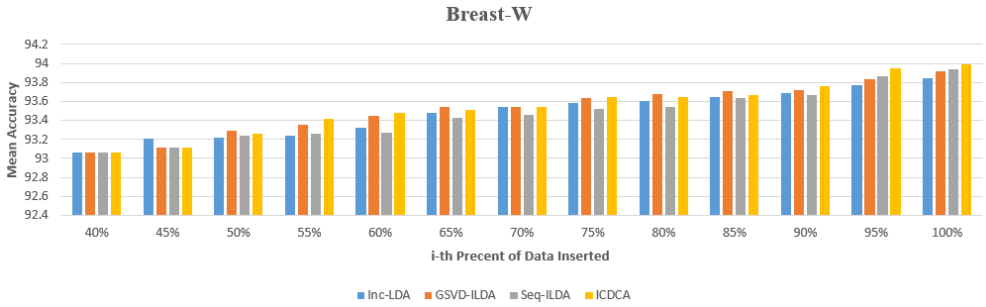
[1] Available at http://www.ics.uci.edu/mlearn/MLRepository.html.


Protein
Mean Accuracy
71.5
71
70.5
70
69.5
69
68.5
68
67.5
67
66.5
40%
45%
50%
55%
60%
65%
70%
75%
80%
85%
90%
95%
100%
i-th Precent of Data Inserted
Inc-LDA
GSVD-ILDA
Seq-ILDA
ICDCA

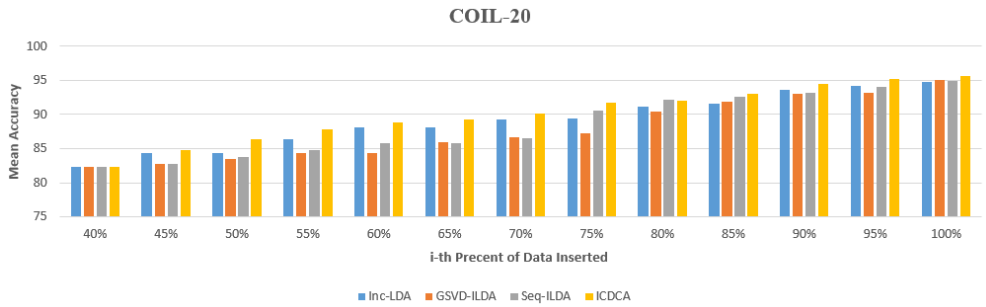
b)


Diabetes
Mean Accuracy
60
59
58
57
56
55
54
53
40%
45%
50%
55%
60%
65%
70%
75%
80%
85%
90%
95%
100%
i-th Precent of Data Inserted
Inc-LDA
GSVD-ILDA
Seq-ILDA
ICDCA

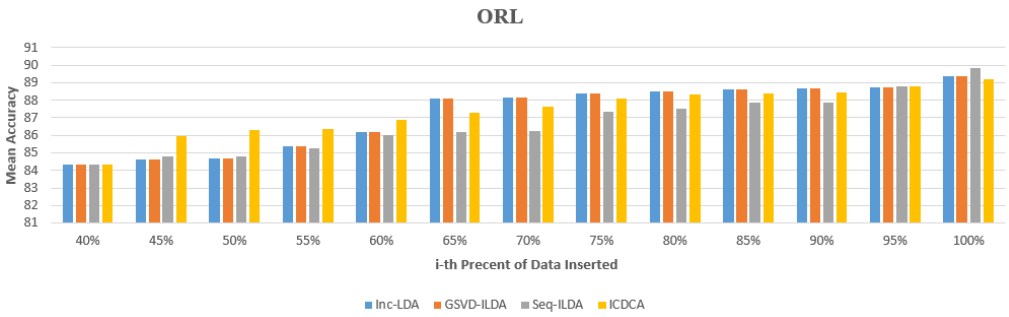
c)


Breast-W
Mean Accuracy
94.2
94
93.8
93.6
93.4
93.2
93
92.8
92.6
92.4
40%
45%
50%
55%
60%
65%
70%
75%
80%
85%
90%
95%
100%
i-th Precent of Data Inserted
Inc-LDA
GSVD-ILDA
Seq-ILDA
ICDCA

d)


COIL-20
Mean Accuracy
100
95
90
85
80
75
40%
45%
50%
55%
60%
65%
70%
75%
80%
85%
90%
95%
100%
i-th Precent of Data Inserted
Inc-LDA
GSVD-ILDA
Seq-ILDA
ICDCA

e)



f)

Figure 2. Mean Accuracy Comparison when data points inserted gradually between incremental LDA algorithms in different data sets including: a) Iris, b) Protein, c) Diabetes, d) Breast-W, e) COIL-20, f) ORL

5.2. Speed Comparison

In this experiment, we aim to compare computational cost of incremental algorithms. The algorithms and datasets in comparison is similar to previous experiment. Table 2 represents the result of running each algorithm for each insertion of data point from 40% to 100%. This comparison shows that ICDCA is more efficient in the environment with high dimensional data and small number of classes.

Table 2. Demonstration of time cost per each iteration in second unit

	Inc-LDA	GSVD-ILDA	Seq-ILDA	ICDCA
Iris	**0.12**	0.15	0.20	0.27
Protein	0.38	**0.31**	0.90	1.25
Diabetes	0.25	0.26	0.30	1.52
Breast-W	0.26	0.27	0.29	0.85
COIL	4.75	**4.31**	10.62	21.63
ORL	10.21	**9.63**	61.51	84.47

6. Conclusion

In this paper, an incremental algorithm called ICDCA is proposed. In ICDCA, we reduced the computational cost by modifications to Xiang's offline method. 1) Rewriting scatter matrix formulation which made us capable of proposing an updating routine thus eliminate the need to compute them from scratch. To show the efficiency of ICDCA, The time complexity of recently proposed incremental LDA problem is compared with ICDCA. In addition, Experiments using publicly available databases have been performed to evaluate the performance of ICDCA. We compared ICDCA with other online labeled LDA algorithms including: Kim's (Inc-LDA) [33, 34], Zhao's (GSVD-ILDA) [31] and Pang's (Seq-ILDA) [3] method. Furthermore the computational cost of ICDCA and other incremental algorithms is compared. In this experiment, ICDCA successfully update the projection direction as new data points arrives.

References

[1] A. R. Webb, K. D. Copsey and G. Cawley, Statistical pattern recognition: Wiley, **(2011)**.
[2] L. P. Liu, Y. Jiang and Z. H. Zhou, "Least square incremental linear discriminant analysis", in Data Mining, 2009. ICDM'09. Ninth IEEE International Conference, **(2009)**, pp. 298-306.
[3] S. Pang, S. Ozawa and N. Kasabov, "Incremental linear discriminant analysis for classification of data streams", Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, vol. 35, **(2005)**, pp. 905-914.
[4] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection", Neural Networks, IEEE Transactions, vol. 6, **(1995)**, pp. 296-317.
[5] C. Chatterjee and V. P. Roychowdhury, "On self-organizing algorithms and networks for class-separability features", Neural Networks, IEEE Transactions, vol. 8, **(1997)**, pp. 663-678.
[6] J. Ye, "Least squares linear discriminant analysis", Proceedings of the 24th international conference on Machine learning, **(2007)**, pp. 1087-1093.
[7] Q. Wang and L. Zhang, "Least squares online linear discriminant analysis", Expert Systems with Applications, vol. 39, **(2012)**, pp. 1510-1517.
[8] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection", Pattern Analysis and Machine Intelligence, IEEE Transactions, vol. 19, **(1997)**, pp. 711-720.
[9] D. L. Swets and J. J. Weng, "Using discriminant eigenfeatures for image retrieval", Pattern Analysis and Machine Intelligence, IEEE Transactions, vol. 18, **(1996)**, pp. 831-836.
[10] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces", Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference, **(1991)**, pp. 586-591.
[11] S. Dudoit, J. Fridlyand and T. P. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data", Journal of the American statistical association, vol. 97, **(2002)**, pp. 77-87.
[12] M. W. Berry, S. T. Dumais and G. W. O'Brien, "Using linear algebra for intelligent information retrieval", SIAM review, vol. 37, **(1995)**, pp. 573-595.
[13] T. Haste and R. Tibshirani, "J. Friedman The elements of statistical learning (2nd)", ed: Springer, **(2009)**.
[14] H. Wang, S. Yan, D. Xu, X. Tang and T. Huang, "Trace ratio vs. ratio trace for dimensionality reduction", in Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference, **(2007)**, pp. 1-8.
[15] S. Xiang, F. Nie and C. Zhang, "Learning a Mahalanobis distance metric for data clustering and classification", Pattern Recognition, vol. 41, **(2008)**, pp. 3600-3612.
[16] S. Basu, I. Davidson and K. L. Wagstaff, "Constrained clustering: Advances in algorithms, theory, and applications", Chapman & Hall/CRC, **(2009)**.
[17] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning", Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 3, **(2009)**, pp. 1-130.
[18] S. Basu, M. Bilenko and R. J. Mooney, "Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering", in Proceedings of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, **(2003)**, pp. 42-49.
[19] X. Liu, T. Chen and S. M. Thornton, "Eigenspace updating for non-stationary process and its application to face recognition", Pattern Recognition, vol. 36, **(2003)**, pp. 1945-1959.
[20] K. Fukunaga, "Introduction to statistical pattern classification", Academic Press, San Diego, California, USA, vol. 1, **(1990)**, pp. 2.
[21] K. Wagstaff, C. Cardie, S. Rogers and S. Schrödl, "Constrained k-means clustering with background knowledge", Machine Learning-International Workshop then Conference, **(2001)**, pp. 577-584.
[22] S. Basu, A. Banerjee and R. Mooney, "Semi-supervised clustering by seeding", Machine Learning-International Workshop then Conference, **(2002)**, pp. 19-26.
[23] S. C. H. Hoi, W. Liu, M. R. Lyu and W. Y. Ma, "Learning distance metrics with contextual constraints for image retrieval", in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference, **(2006)**, pp. 2072-2078.
[24] R. A. Fisher, "The use of multiple measurements in taxonomic problems", Annals of Human Genetics, vol. 7, **(1936)**, pp. 179-188.
[25] J. Ye and T. Xiong, "Computational and theoretical analysis of null space and orthogonal linear discriminant analysis", The Journal of Machine Learning Research, vol. 7, **(2006)**, pp. 1183-1204.
[26] L. -F. Chen, H. -Y. M. Liao, M.-T. Ko, J. -C. Lin and G. -J. Yu, "A new LDA-based face recognition system which can solve the small sample size problem", Pattern Recognition, vol. 33, **(2000)**, pp. 1713-1726.
[27] J. Ye, "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems", Journal of Machine Learning Research, vol. 6, **(2006)**, pp. 483.

[28] J. Ye, R. Janardan, C. H. Park and H. Park, "An optimization criterion for generalized discriminant analysis on undersampled problems", Pattern Analysis and Machine Intelligence, IEEE Transactions, vol. 26, **(2004)**, pp. 982-994.
[29] D. Chu, S. T. Goh and Y. Hung, "Characterization of all solutions for undersampled uncorrelated linear discriminant analysis problems", SIAM Journal on Matrix Analysis and Applications, vol. 32, **(2011)**, pp. 820-844.
[30] J. Rubner and P. Tavan, "A self-organizing network for principal-component analysis", EPL (Europhysics Letters), vol. 10, **(2007)**, pp. 693.
[31] H. Zhao and P. C. Yuen, "Incremental linear discriminant analysis for face recognition", Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, vol. 38, **(2008)**, pp. 210-221.
[32] J. Ye, Q. Li, H. Xiong, H. Park, R. Janardan and V. Kumar, "IDR/QR: an incremental dimension reduction algorithm via QR decomposition", Knowledge and Data Engineering, IEEE Transactions, vol. 17, **(2005)**, pp. 1208-1222.
[33] T. K. Kim, S. F. Wong, B. Stenger, J. Kittler and R. Cipolla, "Incremental linear discriminant analysis using sufficient spanning set approximations", Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference, **(2007)**, pp. 1-8.
[34] T. -K. Kim, B. Stenger, J. Kittler and R. Cipolla, "Incremental Linear Discriminant Analysis Using Sufficient Spanning Sets and Its Applications", International Journal of Computer Vision, vol. 91, **(2011)**, pp. 216-232, 2011/01/01.
[35] Y. -F. Guo, S. -J. Li, J. -Y. Yang, T. -T. Shu and L. -D. Wu, "A generalized Foley–Sammon transform based on generalized fisher discriminant criterion and its application to face recognition", Pattern Recognition Letters, vol. 24, **(2003)**, pp. 147-158.
[36] K. Liu, Y. -Q. Cheng and J. -Y. Yang, "A generalized optimal set of discriminant vectors", Pattern Recognition, vol. 25, **(1992)**, pp. 731-739.
[37] P. Hall, D. Marshal and R. Martin, "Merging and Splitting Eigenspace Models", Pattern Analysis and Machine Intelligence, IEEE Transactions, vol. 22, **(2000)**, pp. 1042-1049.

Authors

Hadi Sadoghi Yazdi is currently an Associate Professor of Computer Science and Engineering at Ferdowsi University of Mashhad (FUM). He received his B.S. degree in Electrical Engineering from FUM in 1994, and received his M.S. and Ph.D. degrees in Electrical Engineering from Tarbiat Modares University in 1996 and 2005, respectively. Dr. Sadoghi Yazdi has received several awards including Outstanding Faculty Award and Best System Design Award in 2007. His research interests are in the areas of Pattern Recognition, Machine Learning, Machine Vision, Signal Processing, Data Mining and Optimization.