

Stream Data Classification Using Improved Fisher Discriminate Analysis

Chen Ling

Department of Computer Science, Yangzhou University, Yangzhou, China
 State Key Lab of Novel Software Tech, Nanjing University, Nanjing, China
lchen@yzcn.net

Zou Ling-Jun¹, Tu Li²

¹Department of Computer Science, Yangzhou University, Yangzhou, China

² College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Abstract—A modified Fisher discriminate analysis method for classifying stream data is presented. To satisfy the real-time demand in classifying stream data, this method defines a new criterion for Fisher discriminate analysis. Since the new criterion requires less computation and memory space, it is much faster and more suitable for online processing in stream data environment. It can overcome the problem of singular within-class scatter matrix in traditional FDA. Our algorithm speeds up the mining process while maintaining the high classification accuracy and capturing the up-to-date trends in the stream. Experiments on real and synthetic data sets show that our algorithm can improve the classification accuracy and speed for stream data classification.

Index Terms — data mining, classification, Fisher discriminate analysis

I. INTRODUCTION

Advances in data storage technology have led to the ability to store the data for real-time transactions. Such processes lead to data which often grow without limit and are referred to as data streams. Examples of such data stream include network event log, financial data, credit card transactional flows, etc. Unlike traditional data sets, stream data flow in and out of a computer system continuously and with varying update rates. They are temporally ordered, fast changing, massive and potentially infinite. Researches on data stream mining mainly focus on modeling[1], query[2,3], regression[4], clustering[5-7] and classification[8-13].

Classification is an important stream data mining task. The main thrust on data stream mining in the context of classification has been that of one-pass mining. In reality, the nature of the underlying changes in the data stream can impose considerable challenges.

The classification problem of data stream is defined as follows: assuming that a data stream consists of sequence of data records, r_1, r_2, \dots, r_n , where n could be an arbitrarily large integer. Each data record consists of a set of attributes. Let $A = \{A_1, A_2, \dots, A_d\}$ be a set of d attributes

each of which may be symbolic or numeric. A data record takes the form of $r = (a_1, a_2, \dots, a_d, y)$ where a_j is the value of attribute A_j , while y is a discrete class label of r . The goal of stream data classification is to build a classifier based on stream data in order to predict the class label of unknown, coming records with high accuracy. For example, a_1, a_2, \dots, a_d could be a description of a man's health indexes, and y the decision whether the man has diabetes or not. In data stream, it is impossible to use conventional classification algorithms because of its high volume and concept drifts which is the problem that the underlying process behind the stream data may not be static, i.e., it may change over time. It is infeasible to store and process the entire data stream in memory and it is insufficient to build a static model to classify the entire data.

In recent years, classification in the context of data stream has been an active research area. Several online learning methods[8-10] have been proposed for the classification of stream data. They are one-pass incremental learning methods, continuously incorporating new data when they arrive and revise the model. While some of these methods are sensitive to the order of coming data, some models constructed can not balance between accuracy and efficiency, they rely on some costly updating procedures, others do not consider the underlying concept drift.

Several incremental PCA methods[14-16] have been proposed, which use principal components analysis (PCA) for online classification. However, PCA is essentially a technique commonly used for dimensionality reduction. This method chooses dimensionality reducing linear projection that maximizes the scatter of all projected samples. The disadvantage of the approach[17] is that the scatter being maximized is due not only to the between-class scatter that is useful for classification, but also to the within-class scatter that for classification proposes, is unnecessary information.

An ILDA[12] method uses incremental Fisher discriminate analysis for constructing classifiers on data

stream. The method provides two computational approaches: sequential ILDA and Chunk ILDA. However, both of them only produce a single model to represent the entire data stream. It is less effective when the present data have different distribution from historical data. The model built on the entire data stream is not accurate for classifying recent data generated from different concept. In this case, it is reasonable that the model built on a certain amount of recent data. Another problem is that it does not consider the phenomena that the within-class scatter matrix may singular, and traditional Fisher discriminate analysis is not applicable in this case.

In this paper, we propose an improved Fisher discriminate analysis for classifying data stream. We assume that the model construction and testing process are simultaneous which is often the case in real applications when some data are labeled while others not. We continuously learn models that capture the up-to-date model within a sliding window. We use an improved Fisher discriminate analysis to seek directions for efficient discrimination. It works efficiently when the within-class is singular and is more suitable in streaming environment.

The remainder of the paper is organized as follows. In Section 2, we give a brief overview of Fisher discriminate analysis. The method of improved Fisher discriminate analysis is presented in Section 3. Section 4 proposes a method to classify the stream data on a sliding window and describe the framework of our algorithm DFDA. Experimental results are shown and analyzed in Section 5, and Section 6 concludes the paper.

II. FISHER DISCRIMINATE ANALYSIS

Fisher discriminate analysis (FDA) [18,19] is a linear methods for dimensionality reduction, it searches for those vectors in the underlying space that best discriminate among classes. It tries to shape the scatter in order to make it more reliable for classification.

Let us consider a set of n samples $x = \{x_{gk}^{(i)}\}$, where datum is in an n -dimensional space, and belongs to one of the G classes. Here, $g = 1, 2, \dots, G$ is label of the class, $k = 1, 2, \dots, n_g$ is the index of a datum in its class, n_g is the number of data in class g , $i = 1, 2, \dots, n$ is the superscript of the element in vector x_{gk} .

For n samples $x = \{x_{gk}^{(i)}\}$ in G classes, its total-class scatter matrix $s_t = [t_{ij}]$ is defined as

$$s_t = \sum_{g=1}^G \sum_{k=1}^{n_g} (x_{gk} - \bar{x})(x_{gk} - \bar{x})^T \quad (1)$$

Here, x_{gk} is the k th data of class g , $\bar{x} = (\bar{x}^{(1)}, \bar{x}^{(2)}, \dots, \bar{x}^{(n)})$ is the mean of all classes. For S_t , its element t_{ij} can be calculated by:

$$t_{ij} = \sum_{g=1}^G \sum_{k=1}^{n_g} (x_{gk}^{(i)} - \bar{x}^{(i)})(x_{gk}^{(j)} - \bar{x}^{(j)}).$$

The between-class scatter matrix $s_b = [b_{ij}]$ is defined by

$$s_b = \sum_{g=1}^G (\bar{x}_g - \bar{x})(\bar{x}_g - \bar{x})^T \cdot n_g \quad (2)$$

where $\bar{x}_g^{(i)} = \frac{1}{n_g} \sum_{k=1}^{n_g} x_{gk}^{(i)}$, $\bar{x}^{(i)} = \frac{1}{N} \sum_{g=1}^G \sum_{k=1}^{n_g} x_{gk}^{(i)}$. Its element

b_{ij} can be calculated by:

$$b_{ij} = \sum_{g=1}^G n_g (\bar{x}_g^{(i)} - \bar{x}^{(i)})(\bar{x}_g^{(j)} - \bar{x}^{(j)}).$$

The within-class scatter matrix $s_w = [w_{ij}]$ is defined by

$$s_w = \sum_{g=1}^G \sum_{k=1}^{n_g} (x_{gk} - \bar{x}_g)(x_{gk} - \bar{x}_g)^T \quad (3)$$

Its element w_{ij} can be calculated by:

$$w_{ij} = \sum_{g=1}^G \sum_{k=1}^{n_g} (x_{gk}^{(i)} - \bar{x}_g^{(i)})(x_{gk}^{(j)} - \bar{x}_g^{(j)}).$$

The scatter matrices have the relationship $S_t = S_b + S_w$.

The objective of Fisher discriminate analysis is to find a linear projection w over x to maximize the ratio of between-class s_b against within-class scatter s_w . The optimal projection w is chosen as the matrix with orthonormal columns by the following criterion.

$$w_{opt} = \arg \max_w \frac{w^T s_b w}{w^T s_w w} \quad (4)$$

It is easy to prove that if s_w is a nonsingular matrix, the ratio $\frac{w^T s_b w}{w^T s_w w}$ is maximized when the column vectors w are the eigenvectors of $s_w^{-1} s_b$.

This traditional Fisher discriminate analysis can not directly used in streaming environment, due to the characteristics of stream data. Since new data are continuously arriving, it is impossible to store all the data in the stream. The computation resources such as storage and CPU time can not afford the direct use of FDA. The traditional FDA method needs to maintain s_w and s_b in main memory. Along with the continuously arriving data, new classes of data could be emerged at any time. There mean values \bar{X}_g and \bar{X} of s_w and s_b have to be adjusted accordingly. This will cost large amount of computation time. In addition, since the column vectors of projection matrix w are the eigenvectors of $s_w^{-1} s_b$, it requires computations of matrix inversion and matrix multiplication. Therefore obtaining w is computationally expensive and not suitable for real time processing. Moreover, traditional FDA will be no longer effective when s_w is a singular matrix.

III. IMPROVED FISHER DISCRIMINATE ANALYSIS

In an effort to overcome above drawbacks of traditional FDA, we proposed improved Fisher discriminate analysis based on a new criterion.

Lemma 1.

Let $X_i = (x_{i1}, x_{i2}, \dots, x_{in_i})$, $\bar{X} = (\bar{x}^{(1)}, \bar{x}^{(2)}, \dots, \bar{x}^{(n)})^T$, $e^T = (1, 1, \dots, 1)_{n \times 1}$, denote $D_i = X_i - \bar{X}e^T$, $D = (D_1, D_2, \dots, D_G)_{n \times N}$, then $s_i = D \cdot D^T$.

Proof:

$$t_{ij} = \sum_{g=1}^G \sum_{k=1}^{n_g} (x_{gk}^{(i)} - \bar{x}^{(i)})(x_{gk}^{(j)} - \bar{x}^{(j)}) = \sum_{r=1}^N (x_r^{(i)} - \bar{x}^{(i)})(x_r^{(j)} - \bar{x}^{(j)})$$

$$= \sum_{g=1}^G \begin{bmatrix} x_{g1}^{(1)} - \bar{X}^{(1)} & x_{g2}^{(1)} - \bar{X}^{(1)} & \dots & x_{gn_g}^{(1)} - \bar{X}^{(1)} \\ x_{g1}^{(2)} - \bar{X}^{(2)} & x_{g2}^{(2)} - \bar{X}^{(2)} & \dots & x_{gn_g}^{(2)} - \bar{X}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_{g1}^{(n)} - \bar{X}^{(n)} & x_{g2}^{(n)} - \bar{X}^{(n)} & \dots & x_{gn_g}^{(n)} - \bar{X}^{(n)} \end{bmatrix}$$

$$\begin{bmatrix} x_{g1}^{(1)} - \bar{X}^{(1)} & x_{g2}^{(1)} - \bar{X}^{(1)} & \dots & x_{gn_g}^{(1)} - \bar{X}^{(1)} \\ x_{g1}^{(2)} - \bar{X}^{(2)} & x_{g2}^{(2)} - \bar{X}^{(2)} & \dots & x_{gn_g}^{(2)} - \bar{X}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_{g1}^{(n)} - \bar{X}^{(n)} & x_{g2}^{(n)} - \bar{X}^{(n)} & \dots & x_{gn_g}^{(n)} - \bar{X}^{(n)} \end{bmatrix}^T$$

$$= D \cdot D^T$$

Q.E.D.

Lemma 2.

Let $E = \text{diag}(E_{n_1}, E_{n_2}, \dots, E_{n_g})$ be an $N \times N$ matrix, E_{n_g} is an $n_g \times n_g$ matrix where each element is equal to $\frac{1}{n_g}$, then $s_b = D \cdot E \cdot D^T$.

Proof:

$$s_b = \sum_{g=1}^G (\bar{X}_g - \bar{X})(\bar{X}_g - \bar{X})^T \cdot n_g$$

$$= \sum_{g=1}^G n_g \left[\frac{1}{n_g} \sum_{k=1}^{n_g} (x_{gk} - \bar{X}) \right] \left[\frac{1}{n_g} \sum_{k=1}^{n_g} (x_{gk} - \bar{X}) \right]^T$$

$$= \sum_{g=1}^G \frac{1}{n_g} \left[\sum_{k=1}^{n_g} (x_{gk} - \bar{X}) \right] \left[\sum_{k=1}^{n_g} (x_{gk} - \bar{X}) \right]^T$$

$$= \sum_{g=1}^G \frac{1}{n_g} (\bar{x}_{g1} - \bar{x}, \bar{x}_{g2} - \bar{x}, \dots, \bar{x}_{gn_g} - \bar{x}) \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} (1 \ 1 \ \dots \ 1) \begin{pmatrix} \bar{x}_{g1} - \bar{x} \\ \bar{x}_{g2} - \bar{x} \\ \vdots \\ \bar{x}_{gn_g} - \bar{x} \end{pmatrix}$$

$$= \sum_{g=1}^G \begin{bmatrix} x_{g1}^{(1)} - \bar{X}^{(1)} & x_{g2}^{(1)} - \bar{X}^{(1)} & \dots & x_{gn_g}^{(1)} - \bar{X}^{(1)} \\ x_{g1}^{(2)} - \bar{X}^{(2)} & x_{g2}^{(2)} - \bar{X}^{(2)} & \dots & x_{gn_g}^{(2)} - \bar{X}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_{g1}^{(n)} - \bar{X}^{(n)} & x_{g2}^{(n)} - \bar{X}^{(n)} & \dots & x_{gn_g}^{(n)} - \bar{X}^{(n)} \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{n_g} & \frac{1}{n_g} & \dots & \frac{1}{n_g} \\ \frac{1}{n_g} & \frac{1}{n_g} & \dots & \frac{1}{n_g} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{n_g} & \frac{1}{n_g} & \dots & \frac{1}{n_g} \end{bmatrix} \begin{bmatrix} x_{g1}^{(1)} - \bar{X}^{(1)} & x_{g2}^{(1)} - \bar{X}^{(1)} & \dots & x_{gn_g}^{(1)} - \bar{X}^{(1)} \\ x_{g1}^{(2)} - \bar{X}^{(2)} & x_{g2}^{(2)} - \bar{X}^{(2)} & \dots & x_{gn_g}^{(2)} - \bar{X}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_{g1}^{(n)} - \bar{X}^{(n)} & x_{g2}^{(n)} - \bar{X}^{(n)} & \dots & x_{gn_g}^{(n)} - \bar{X}^{(n)} \end{bmatrix}$$

$$= \begin{bmatrix} D_1 & D_2 & \dots & D_G \end{bmatrix} \begin{bmatrix} E_1 & & & \\ & E_2 & & \\ & & \ddots & \\ & & & E_G \end{bmatrix} \begin{bmatrix} D_1^T \\ D_2^T \\ \vdots \\ D_G^T \end{bmatrix}$$

$$= D \cdot E \cdot D^T$$

Q.E.D.

In traditional FDA, If s_w is nonsingular matrix, the column vectors of projection matrix w are the eigenvectors of $s_w^{-1}s_b$. But in some cases, we are confronted the difficulty that the within-class scatter matrix s_w is singular. This stems from the fact that the number of learning samples is much smaller than the number of dimensions in some cases. To overcome this drawback, we need to define new modified criterion for Fisher discriminate analysis. This method retrieve an optimal vector w_{opt} by maximizing the difference between $w^T s_w w$ and $w^T s_b w$ instate of their ratio.

$$w_{opt} = \arg \max_{\|w\|=1} (w^T s_b w - w^T s_w w)$$

$$\text{and } w^T s_b w - w^T s_w w > 0. \quad (5)$$

It can be easily be seen that the optimal vector w_{opt} obtained by criterion (5) is equal to that by (4) in the traditional FDA. They all find a projection w_{opt} that maximizes between-class scatter s_b and minimize within-class s_w . But our method need not to compute the inverse of s_w , so it can run faster than traditional FDA.

Lemma 3.

The difference between $w^T s_b w$ and $w^T s_w w$ is maximized, when the column vectors of projection matrix w are the eigenvectors of $(s_b - s_w)$.

Proof: Consider the following Lagrange function:

$$F(w, \lambda) = (w^T s_b w - w^T s_w w) - \lambda (\|w\| - 1)$$

$$= w^T (s_b - s_w) w - \lambda (w^T w - 1)$$

Let $\frac{\partial F(w, \lambda)}{\partial w} = 0$, then $(s_b - s_w) \times w = \lambda \times w$. We

get $(w^T s_b w - w^T s_w w) = \lambda \times w^T w = \lambda$. Therefore the optimal direction is the eigenvector corresponding to the largest eigenvalue of matrix $(s_b - s_w)$. **Q.E.D.**

According to Lemma 3, we just compute the eigenvectors of matrix $(s_b - s_w)$ when using criterion (5) to replace (4) to avoid matrix inversion and the problem of singular s_w matrix.

Noticing that

$$s_b = D \cdot E \cdot D^T, s_w = s_t - s_b = D \cdot D^T - D \cdot E \cdot D^T,$$

we have

$$s_b - s_w = D \cdot E \cdot D^T - D \cdot D^T + D \cdot E \cdot D^T$$

$$= 2D \cdot E \cdot D^T - D \cdot D^T = D(2E - I)D^T$$

Therefore in our algorithm, only matrices D and E are saved. Since $E = \text{diag}(E_{n_1}, E_{n_2}, \dots, E_{n_g})$ and all

the elements of E_{n_g} are equal to $\frac{1}{n_g}$, we only retain

matrix D and $n_g, g=1,2,\dots,G$. Compare with the traditional FDA, much less memory space is required in our method.

The modified Fisher discriminate analysis is more suitable for stream data environment, because it is much faster and suitable for online processing, it can overcome the singular problem, and it requires much less memory space.

IV. CLASSIFY THOUGH SLIDING WINDOW

Traditional classification methods work in a batch mode, it constructs models over the entire datasets. But for stream data classifying, where data are coming continuously and is potentially infinite, it is impossible to build a model on the entire data stream due to the time and space limitation. Besides, it is incorrect to assume that the classes of the data is stationary, but in fact the number of the total classes is changing over time. Along with the new data coming continuously, some classes may disappear while some new classes emerge. So it is insufficient to build a static model to classify the entire data due to the presence of concept drifts. For example, the economic condition may change abruptly, and the weather condition may vary rapidly. A new class which has not been observed in the entire history of the stream may suddenly emerge because of the changes in the underlying process which generates the stream. In such a case, if the model learned from the evolving data stream which generated by a series of concepts, the inaccurate results could be obtained. Therefore, it is not desirable to treat the data stream as a long sequence of static data

since we are interested in the most recent classification feature of the data stream. In such a case it may be advantageous to use a smaller and more recent portion of the stream for classification process.

Our approach is to repeatedly apply modified Fisher discriminate analysis to a sliding window of N examples. Sliding window is a noticeable technique in mining data stream. This technique involves using recent data from the data stream rather than considering the entire data stream. When new data is continuously streaming in and the older data is expired concurrently, new examples are inserted into the beginning of the sliding window, and corresponding old examples are removed from the end. Then a new model using modified Fisher discriminate analysis is built in the current sliding window. The model learned can reflect current concept. The sliding window size relies on the rate of concept drift. We do not construct a model when every new datum arrives, but at the rate of l , which is the size of a basic window. This means, the algorithm builds a new sliding window for every l new coming data. The process of classifying stream data over sliding window is depicted in Fig. 1.

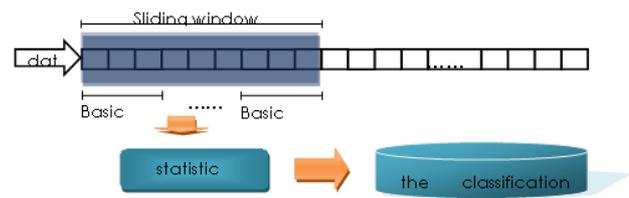


Figure 1. classify on a sliding window

In our algorithm, we only store matrix D . Assuming each sliding window consists of m basic windows, the algorithm only store information of the newest $N=m \cdot l$ data. Therefore D is an $n \cdot N$ matrix.

The outline of our algorithm is as follows:

Procedure DFDA

Input: s : a data stream

Output: M : an update classification model

1. Initialization phase.
 2. **While** not end of the stream **do**
 - 2.1 Once a new labeled data streaming in, store its elements and class information, increase the size of the class by one which the new data belongs to.
 - 2.2 **If** the number of new data reaches the size of basic window l **then**
 - 2.2.1 expire the data in the oldest basic window, count the number of elements in each class for current sliding window by adding the new ones and subtracting the old ones.
 - 2.2.2 update the new mean value for all data in the sliding window .
 - 2.2.3 update the matrix D and $n_g, g=1,2,\dots,G$.
 - 2.2.4 rebuild a model M using modified Fisher discriminate analysis.
 - 2.3 **endif**
 3. **endwhile**
 4. **Return** M .
-

The initialization phase is implemented as an offline process at the beginning of the data stream processing. In this phase, the initial D and E matrices are constructed and initial model M is created.

Once the updated model M is created, the unlabelled data within the same sliding window can be classified, and the changing tendency of each class can be predicted.

V. EXPERIMENTAL RESULT

We conducted a series of experiments to compare the efficiency and accuracy of our algorithm with that of incremental PCA and traditional Fisher discriminate analysis for classifying stream data. We test our algorithm on both synthetic and real datasets. All the experiments were conducted on a PC with Windows XP professional operating system.

A. Testing data

In the experiments, the real datasets used are selected from the UCI Machine Learning Repository at <http://www.ics.uci.edu/ml/learn/MLRepository.html>. We choose five standard datasets, each of which has its features 100% of continuous values and no missing value. Assuming the data comes in a time series manner in the test.

We also use synthetic datasets in the experiments. Those data were generated randomly in the range from 100k to 1000k, the number of classes is from 10 to 40, and the dimensionality is in the range of 10 to 50. The data elements of the same dimension in each class follow a Gaussian distribution.

B. Accuracy study

We carried out our experiment on five UCI datasets to test the recognition rate. Fig. 2 shows the classification results on Iris dataset under different sliding window size. 2(a) is the original distribution of Iris data by the first two dimensions. 2(b) and 2(c) denote the classification results by our algorithm DFDA when the sliding window size are 90 and 150 respectively. 2(d) demonstrates IPCA algorithm on Iris dataset when the sliding window size are 90. As we can see from 2(b) and 2(d), our algorithm DFDA can achieve higher classification accuracy than IPCA.

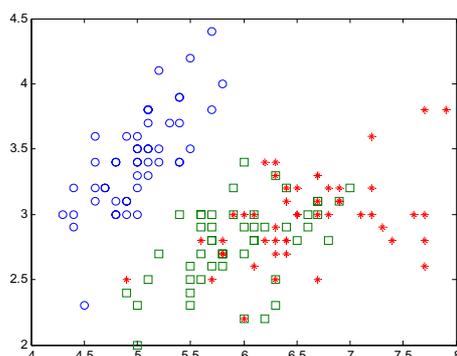


Figure 2(a)

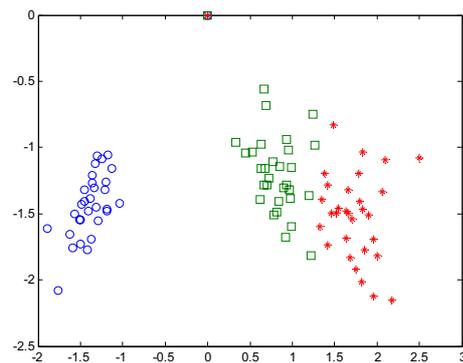


Figure 2(b)

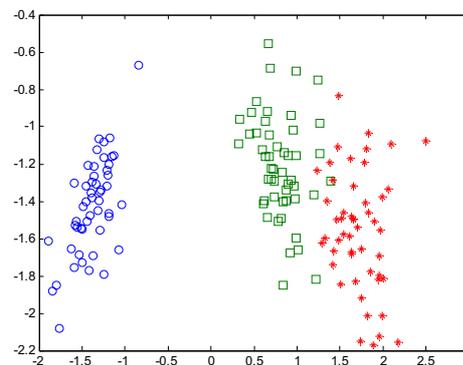


Figure 2(c)

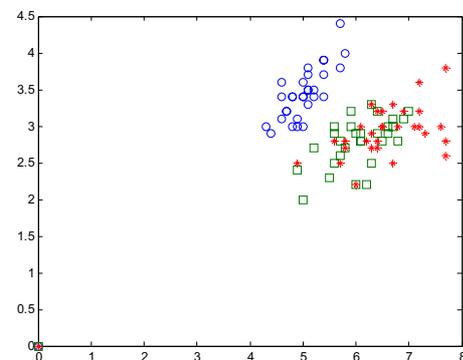


Figure 2(d)

Table 1 shows the comparison of classification accuracy of DFDA, FDA, and IPCA on five UCI datasets (Cancer, Iris, Liver, Glass, Wine). We use a leave-one-out cross-validation policy to evaluate the classification accuracy.

TABLE I
RECOGNITION RATES OF DFDA, FDA, AND IPCA ON 5 UCI DATASETS(%)

Name of datasets	Cancer	Iris	Liver	Glass	Wine
Dimensionality of datasets	699×9	150×4	345×6	214×10	178×13
Recognition rates(%)					
FDA	97.5	98.1	62.8	67.1	97.8
DFDA	97.5	98.0	62.7	67.1	97.8
IPCA	96.4	96.3	57.0	51.8	88.6

As shown in table 1, the recognition rate of our algorithm DFDA is exactly the same as the standard FDA, but is higher than that of IPCA.

C. Efficiency and Scalability test

In streaming environment, the processing rate is important for online analysis. In order to test the efficiency and scalability of our algorithm, we conduct experiments on synthetic stream data and compare its performance with that of traditional FDA.

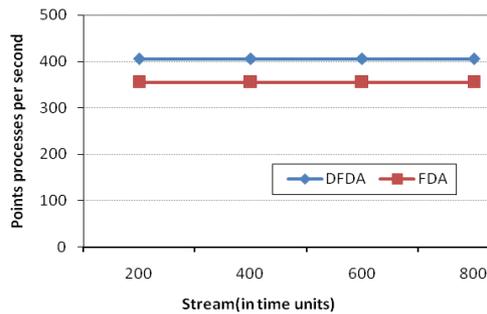


Figure 3. Processing time

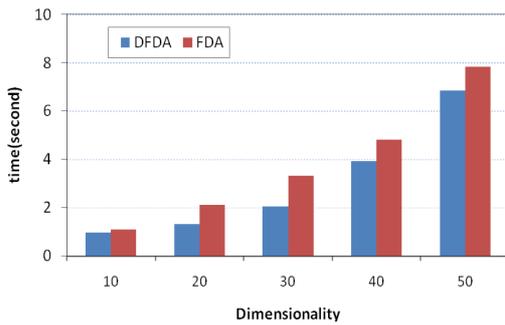


Figure 4. Scalability with dimensionality

Fig. 3 shows comparison of the processing speed of our DFDA with that of traditional FDA. It can easily be seen from the figure that our algorithm DFDA can process about 400 data per second at different time, while traditional FDA processes only about 350 data. This is because in our algorithm, the optimal projection w is computed by maximizing the difference of between-class scatter matrix and within-class scatter matrix. But in FDA, obtaining projection w requires operations of matrix inversion and matrix multiplication which consume large amount of computation time. Therefore computation expense of traditional FDA is much higher than that of DFDA which is more suitable in streaming environment.

Fig. 4 compares the scalability of DFDA with that of traditional FDA. We test the algorithms on data sets with different dimensionalities to compare their computation times. The dimensionality varies from 10 to 50. From the figure we can see that although for both algorithms, the computational time grows as dimensionality increases, our algorithm has a better linear scalability than traditional FDA.

VI. CONCLUSION

A modified Fisher discriminate analysis method for classifying stream data is presented. To satisfy the real-time demand in classifying stream data, this method defines a new criterion for Fisher discriminate analysis. Since the new criterion requires less computation and memory space, it is much faster and more suitable for online processing in stream data environment. It can overcome the problem of singular within-class scatter matrix in traditional FDA. Our algorithm speeds up the mining process while maintaining the high classification accuracy and capturing the up-to-date trends in the stream. Experiments on real and synthetic data sets show that our algorithm can improve the classification accuracy and speed for stream data classification.

ACKNOWLEDGMENT

This research was supported in part by the Chinese National Natural Science Foundation under grant No. 60673060, Natural Science Foundation of Jiangsu Province under contract BK2008075, and the Graduated Education Research Foundation of Jiangsu Province.

REFERENCES

- [1] B. Babcock, S. Babu, M. Datar, R. Motawani, and J. Widom. Models and issues in data stream systems. In PODS, 2002.
- [2] S. Babu and J. Widom. Continuous queries over data streams. SIGMOD Record, 30:109–120, 2001.
- [3] M. Greenwald and S. Khanna. Space-efficient online computation of quantile summaries. In SIGMOD, pages 58–66, Santa Barbara, CA, May 2001.
- [4] Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang. Multi-dimensional regression analysis of time-series data streams. In VLDB, Hongkong, China, 2002.
- [5] Guha S, Mishra N, Motwani R, Callaghan LO. Clustering data stream. In: Proc. of the 41st Annual Symp. on Foundations of Computer Science. Redondo Beach: IEEE Computer Society, 2000. 359-366.
- [6] Aggarwal CC, Han J, Wang J, Yu PS. A framework for projected clustering of high dimensional data streams. In: Nascimento MA, -zsu MT, Kossmann D, Miller RJ, Blakeley JA, Schiefer KB, eds. Proc. of the VLDB. Toronto: Morgan Kaufmann Publishers, 2004. 852-863
- [7] Chalaghan LO, Mishra N, Meyerson A, Guha S. Streaming data algorithms for high-quality clustering. In: Proc. of the 18th Int'l Conf. on Data Engineering. San Jose, 2002. 685-694.
- [8] Domingos P, Hulten G, Spencer L. Mining time-changing data streams. In: Provost F, Srikant R, eds. Proc. of the 7th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. San Francisco: ACM Press, 2001. 97~106.
- [9] Domingos P, Hulten G. Mining high-speed data streams. In: Ramakrishnan R, Stolfo S, Pregibon D, eds. Proc. of the 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Boston: ACM Press, 2000. 71~80.
- [10] H. Wang, W. Fan, P. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In Int'l Conf. on Knowledge Discovery and Data Mining (SIGKDD), 2003.

- [11] C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "On Demand Classification of Data Streams", Proc. 2004 Int. Conf. on Knowledge Discovery and Data Mining (KDD'04), Seattle, WA, Aug. 2004.
- [12] Shaoning Pang, Seiichi Ozawa and Nik Kasabov, Incremental Linear Discriminant Analysis for Classification of Data Streams, IEEE Trans. on System, Man, and Cybernetics-Part B, vol. 35, no. 5, pp. 905-914, 2005
- [13] W. Nick Street and YongSeog Kim. A streaming ensemble algorithm (SEA) for large-scale classification. In SIGKDD, 2001
- [14] J. Weng, Y. Zhang, and W.-S. Hwang, "Candid covariance-free incremental principal component analysis," IEEE Trans. Pattern Anal. Machine Intell., vol. 25, no. 8, pp. 1034-1040, Aug. 2003.
- [15] P. Hall, D. Marshall, and R. Martin, "Merging and splitting eigenspace models," IEEE Trans. Pattern Anal. Machine Intell., vol. 22, no. 9, pp. 1042-1049, Sep. 2000.
- [16] P. Hall and R. Martin, Incremental eigenanalysis for classification," British Machine Vision Conference, Vol. 1, pp. 286-295, 1998.
- [17] Aleix M Martinez, Avinash C Kak. PCA versus LDA [J]. IEEE Trans. On Pattern Analysis and Machine Intelligence, 2001-02, 23(2):228-233.
- [18] R.A. Fisher, "The Statistical Utilization of Multiple Measurements," Annals of Eugenics, vol.8, pp. 376-386, 1938.
- [19] Belhumeur P N, Hespanha J P, Kriegman D J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Analysis and Machine Intelligence, 1997, 19(7): 711-720.

L. Chen was born in Baoying, Jiangsu, P.R.China, in September 10, 1951. He received B. Sc degree in mathematics from Yangzhou Teachers' College, P.R. China in 1976.

He is currently professor of computer science, and the dean of Information Technology College, Yangzhou University, Jiangsu Province, P.R. China. He has published more than 120 papers in journals including IEEE Transactions on Parallel and

Distributed System, Journal of Supercomputing, The Computer Journal. In addition, he has published over 100 papers in refereed conferences. He has also co-authored/co-edited 5 books (including proceedings) and contributed several book chapters. His research interest includes data mining, bioinformatics and parallel processing.

Prof. Chen is a member of IEEE and senior member of the Chinese Computer Society. His recent research has been supported by the Chinese National Natural Science Foundation, Chinese National Foundation for Science and Technology Development and Natural Science Foundation of Jiangsu Province, China. Prof. Chen has organized several national conferences and workshops and has also served as a program committee member for several major international conferences. He was awarded the Government Special Allowance by the State Council, the title of "National Excellent Teacher" by the Ministry of Education, and the Award of Progress in Science and Technology by the Government of Jiangsu Province.

L. J. Zou was born in Luhe, Jiangsu Province, P.R.China, in March 3, 1984. She received B. Sc degree and M. Sc degree in computer science from Yangzhou University, P.R. China in 2005 and 2008 respectively.

Her research interest includes data mining, bioinformatics and parallel processing.

L. Tu was born in Huaiyin, Jiangsu Province, P.R.China, in August 12, 1980. She received B. Sc degree and M. Sc degree in computer science from Yangzhou University, P.R. China in 2003 and 2006 respectively. She is currently a Ph.D candidate in the Institute of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, P.R.China.

Her research interest includes data mining, bioinformatics and parallel processing. She has published more than 10 papers in journals and conferences.