Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Information Sciences 176 (2006) 2042–2065
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

A scalable supervised algorithm for dimensionality reduction on streaming data ☆

Jun Yan [a], Benyu Zhang [b,*], Shuicheng Yan [a], Ning Liu [c], Qiang Yang [d], Qiansheng Cheng [a], Hua Li [e], Zheng Chen [b], Wei-Ying Ma [b]

[a] *LMAM, Department of Information Science, School of Mathematical Science, Peking University, Beijing 100871, PR China*
[b] *Microsoft Research Asia, 49, Zhichun Road, Beijing 100080, PR China*
[c] *Department of Mathematics, Tsinghua University, Beijing 100084, PR China*
[d] *Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong*
[e] *Department of Mathematics, School of Mathematical Science, Peking University, Beijing 100871, PR China*

Abstract

Algorithms on streaming data have attracted increasing attention in the past decade. Among them, dimensionality reduction algorithms are greatly interesting due to the desirability of real tasks. Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two of the most widely used dimensionality reduction approaches. However, PCA is not optimal for general classification problems because it is unsupervised and ignores valuable label information for classification. On the other hand, the performance of LDA is degraded when encountering limited available low-dimensional spaces and singularity problem. Recently, Maximum Margin Criterion (MMC) was

☆ This work conducted at Microsoft Research Asia, Beijing, PR China.

* Corresponding author.
E-mail address: byzhang@microsoft.com (B. Zhang).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2005.11.005

proposed to overcome the shortcomings of PCA and LDA. Nevertheless, the original MMC algorithm could not satisfy the streaming data model to handle large-scale high-dimensional data set. Thus an effective, efficient and scalable approach is needed. In this paper, we propose a supervised incremental dimensionality reduction algorithm and its extension to infer adaptive low-dimensional spaces by optimizing the maximum margin criterion. Experimental results on a synthetic dataset and real datasets demonstrate the superior performance of our proposed algorithm on streaming data.

Keywords: Dimensionality reduction; Streaming data; Principal component analysis (PCA); Linear discriminant analysis (LDA); Maximum margin criterion (MMC)

1. Introduction

There has been an increasing interest on mining streaming data due to the growing number of real applications [2,10,18,19,24]. Streaming data differ from the conventional stored-relation data model in the following ways: (1) the data are not available all at once since they arrive as continuous streams; (2) the order in which data elements arrive cannot be controlled; (3) data streams are potentially without ending; (4) an element from a data stream cannot be retrieved repeatedly; in other words, each sample data could be processed only once; (5) the size of the data can be very large sometimes. These characteristics make the processing of streaming data a challenging issue.

Dimensionality reduction [8] has been one of the main techniques used to facilitate the processing of streaming data. The dimension reduction algorithms are generally classified into feature selection, feature extraction and random projection. The characters of streaming data require the dimension reduction techniques to be as efficient as possible. Thus the common used dimension reduction algorithms used for data streams are random projection [10] and feature selection [25]. However, few works have addressed the feature extraction algorithms for streaming data dimension reduction though they are very effective. This is due to the reason that the computation complexity of most feature extraction techniques is so high that they cannot meet the requirements of streaming data. In this paper, we study the dimensionality reduction on streaming data by different linear feature extraction methods such as Principal Component Analysis (PCA) [16], Linear Discriminant Analysis (LDA) [3], and Maximum Margin Criterion (MMC) [14]. The PCA is unsupervised and is not optimal for general classification problems. The LDA is limited by the singularity problem and class number problem. The recent proposed MMC solved all their shortcomings. Based on the result of analysis, we propose a supervised dimensionality reduction algorithm called Incremental Maximum Margin Criterion (IMMC) to generate the low-dimensional space from streaming data samples. We also use incremental

inter-class scatter criterion to make the algorithm more efficient and robust without the need to reconstruct the criterion matrix when a new sample arrives. Our experiments show that the proposed algorithms can achieve very good performance on reducing the dimension of streaming data.

The rest of the paper is organized as follows. In Section 2, we describe the problem and provide some background information for dimensionality reduction on streaming data. In Section 3, we propose the algorithm called Incremental Maximum Margin Criterion (IMMC) and its extension based on Incremental Inter-class Scatter criterion (IIS). In Section 4, we introduce related work on incremental dimensionality reduction. The experimental results are presented in Section 5. We conclude the paper and discuss the future work in Section 6.

2. Problem definitions and background

For better comprehension, the mathematical definition of our problem is given firstly. Following that, some necessary background such as streaming data and dimensionality reduction concept are showed in this section.

2.1. Problem definitions

Let us consider a labeled data stream $\{u(i)\}$ where each $u(i)$ is a multi-dimensional vector, i denotes the order of sample data as they arrive, and the class label of $u(i)$ is l_i. Formally, the incremental dimensionality reduction problem on streaming data is described as follows: suppose $u(i) \in R^d$, at the time step i, find a transformation matrix $W(i) \in R^{d\times p}$ to project $u(i)$ to a low-dimensional space such that the sample data $y(i) = W^{\mathrm{T}}(i)u(i)$ are well separated by class in the p-dimensional space. Moreover, the sample data before time i, i.e., $u(1), u(2), \ldots, u(i-1)$, could not be used to find $W(i)$. Without loss of generality, the whole problem is defined in Table 1.

Table 1
Problem statement

Input: original high-dimensional data stream $\{u(i) \in R^d\}$, the class label of the ith data is l_i
for $i = 1, 2, \ldots$
$W(i) = f(W(i-1), u(i))$, where f is a function of $W(i-1)$ and $u(i)$
(Note that different algorithms are determined by different functions f)
$y(i) = W^{\mathrm{T}}(i)u(i)$
break when converge
end for
Output: an optimal projection matrix, $W \in R^{d\times p}$

Our two proposed algorithms in this paper are to give two f functions by optimizing two different criterions. Note that the value of f should be updated by its value of last step and the new arrived sample, thus what the problem needs is a one pass algorithm. Then when the iteration procedure converges, we can use the learned $W \in R^{d\times p}$ to project the future data without label and predict their class label in the p-dimensional space.

2.2. Streaming data

Streaming data is a sequence of digitally encoded signals used to represent information in transmission. For streaming data, the input data that are to be operated are not available all at once, but rather arrive as continuous data sequences. For the problem stated in Section 2.1, streaming data differ from the conventional stored relation model in several ways: (1) the sample data in the stream arrive continuously with a high rate, which needs an incremental algorithm with low time complexity; (2) we could not control the order in which data elements arrive; (3) data streams are potentially without ending; (4) an element from a data stream cannot be retrieved easily, in other words, one pass algorithms are preferred for streaming data; and (5) the scale of the sample data is often very large, especially for the online web documents.

The basic mathematical ideas to process streaming data are sampling and random projections. Many different sampling methods [11,15,20] have been proposed: domain sampling, universe sampling, reservoir sampling, etc. There are two main difficulties with sampling for streaming data. First, sampling is not a powerful primitive for many problems since too many samples are needed for performing sophisticated analysis and a lower bound is given in [22]. Second, as stream unfolds, if the samples maintained by the algorithm get deleted, one may be forced to resample from the past, which is in general, expensive or impossible in practice and in any case, not allowed in streaming data problems. Random projections rely on dimensionality reduction, using projection along random vectors. The random vectors are generated by space-efficient computation of random variables. These projections are called the sketches. There are many variations of random projections which are of simpler ilk. For example, counting sketches [6], random subset sums [7] and bloom filters [5].

Streaming data processing is related to the following areas: (1) PAC learning [17]; (2) online algorithms: data stream algorithms have an online component where input is revealed in steps; (3) property testing: this area focused typically on sub-linear time algorithms for testing objects and separating them based on whether they are far from having a desired property, or not; and (4) Markov methods: some data streams may be considered as intermixed states of multiple Markov chains.

2.3. Dimensionality reduction

Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are two most common dimensionality reduction algorithms. Another supervised dimensionality reduction approach named Maximum Margin Criterion (MMC) was proposed recently. Fig. 1 shows the performance of different dimensionality reduction approaches on a synthetic dataset. In this experiment we generated a three-dimensional data set with two classes. As an example, each class consists of 100 samples following a normal distribution with mean $(0,0,0)$ and $(5,5,5)$, respectively. Fig. 1(a) shows a scatter plot of the data set. (b)–(f) are low-dimensional projections of original data by different dimensionality reduction approaches. From (b) and (e) we can see that PCA mixes these two classes of data in both one-dimensional space and two. Although LDA could separate them in (d), the dimension of low-dimensional space found by LDA could not beyond one. (c) and (f) show that MMC could separate these two class in any lower dimensional space.

From this experiment we can see that PCA is not optimal for classification tasks and mixes the two classes; the dimension of space found by LDA could not beyond one since this is a two class problem (the theoretical reason could be found in Section 2.3.2). It is clear that MMC algorithm outperforms PCA for classification and is not limited by the number of classes.

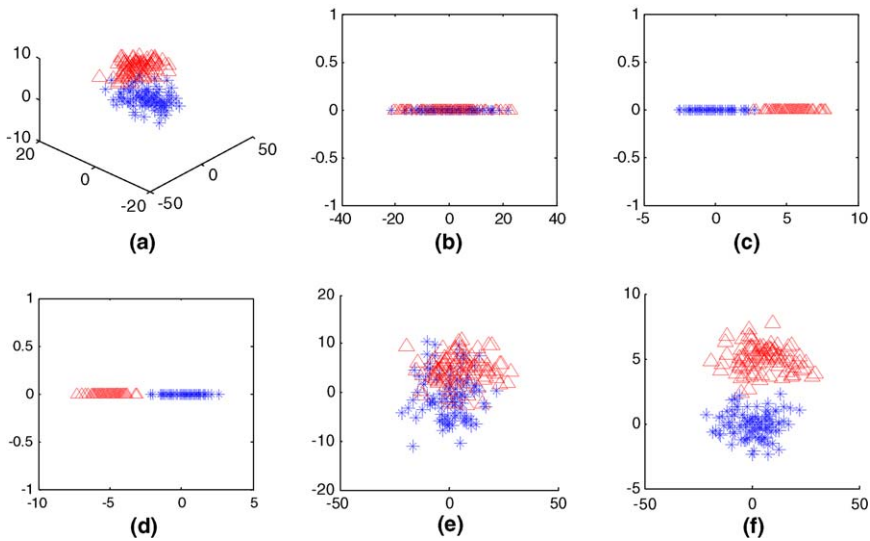

Fig. 1. Different dimensionality reduction algorithms on synthetic data. (a) Original data set; (b) projected to one dimension by PCA; (c) projected to one dimension by MMC; (d) projected to one dimension by LDA; (e) and (f) projected to two dimension by PCA and MMC.

2.3.1. Principal component analysis

The goal of the PCA is to find a low-dimensional space whose basis vectors correspond to the directions with maximal variances. PCA projects the original data into a low-dimensional space; that is, PCA minimizes the reconstruction error in the sense of least square errors. PCA finds the most representative features, but it ignores the class label information. Thus, PCA is not ideal for general classification tasks. The computation cost of PCA mainly lies in the SVD processing of covariance matrix C with computational complexity $O(m^3)$, where m is the smaller number of the data dimension and the number of samples. It is difficult or even impossible to conduct PCA on large-scale data sets with any higher dimensional representations.

2.3.2. Linear discriminant analysis

Linear Discriminant Analysis (LDA) is used to pursue a low-dimensional space that best discriminates the samples from different classes. Suppose $W \in R^{d\times p}$ is the linear projection matrix; LDA aims to maximize the so-called Fisher criterion

$$J(W) = \frac{\mathrm{Trace}(W^{\mathrm{T}} S_b W)}{\mathrm{Trace}(W^{\mathrm{T}} S_w W)},$$

where

$$S_b = \sum_{i=1}^{c} p_i (m_i - m)(m_i - m)^{\mathrm{T}}, \quad S_w = \sum_{i=1}^{c} p_i E(u_i - m_{l_i})(u_i - m_{l_i})^{\mathrm{T}}$$

are called inter-class scatter matrix and intra-class scatter matrix, respectively, where m is the mean of all samples, m_i is the mean of the samples belonging to class i and p_i is the prior probability for a sample belonging to class i. The projection matrix W can be obtained by solving the following generalized eigenvector decomposition problem:

$$S_b w = \lambda S_w w.$$

LDA explicitly utilizes the label information of the samples and is a supervised algorithm. As many works have stated, LDA can outperform PCA in classification problems. On the other hand, there are at most $c - 1$ nonzero eigenvalues, so the upper bound of p is $c - 1$; and S_w must not be singular, which limit the application of LDA. Moreover, it is still difficult for LDA to hand large scale database with high-dimensional representation. As in the Reuters Corpus Volume 1, the number of dimensions in inter-class scatter matrix and intra-class scatter matrix is about 300,000 and it is impossible to conduct generalized eigenvector decomposition on current computer with moderate configuration.

2.3.3. Maximum margin criterion

The Maximum Margin Criterion (MMC) is a feature extraction criterion proposed recently. This new criterion is general in the sense that, when combined with a suitable constraint, it can actually give rise to linear discriminant analysis. Using the same representation with LDA, the purpose of MMC is to maximize the criterion $J(W) = W^{\mathrm{T}}(S_b - S_w)W$. Although both MMC and LDA are supervised dimensionality reduction approaches, the computation of MMC is easier than that of LDA without the inverse of the matrix. The projection matrix W can be obtained by solving the following eigenvector decomposition problem:

$$(S_b - S_w)w = \lambda w.$$

As what has been discussed above, PCA is not optimal for general classification problems since it is an unsupervised algorithm that ignores the valuable label information for classification. On the other hand, the applications of LDA are degraded by the limited available dimensional space and the singularity problem. MMC overcomes these shortcomings. However, MMC has no incremental algorithm thus it could not hand the streaming data or large scale data. In the next section, an incremental MMC and its extension are proposed to transact streaming data and large scale data.

3. Incremental algorithms on streaming data

In this section, we propose a novel incremental supervised dimensionality reduction algorithm and its extension on streaming data. Firstly, we propose the incremental algorithm IMMC to reduce the dimensionality of original high-dimensional data stream by adaptively optimizing the maximum margin criterion. After that we show an extension of MMC, namely weighted MMC. For streaming data, the data arrive with a high rate, the simplicity of the criterion for supervised dimensionality reduction is very important. Thus, we give an incremental algorithm on a special case of the weighted MMC which is more stable and efficient than the original IMMC. This incremental algorithm on the special case is proposed in Section 3.2. Discussion of these two algorithms, which helps us to select appropriate algorithm for different tasks is also given in Section 3.2.

3.1. Incremental maximum margin criterion

In this section, we consider the scenario to maximize the maximum margin criterion that aims to make the class centers as far as possible while the data points in the same class as close as possible. Denote the projection matrix from original space to the low-dimensional space as $W \in R^{d\times p}$. In this work, we propose to incrementally maximize the criterion $J(W) = W^{\mathrm{T}}(S_b - S_w)W$, where S_b

and S_w are the inter-class scatter and intra-class scatter matrix respectively. C is the covariance matrix. Ref. [14] tells us that W is the first k leading eigenvectors of the matrix $S_b - S_w$ and the column vectors of W are orthogonal to each other.

3.1.1. Algorithm derivation

Let us analyze the criterion matrix $S_b - S_w$ and transform it into another convenient form. The transformation and the derivation of our incremental algorithm are based on the two lemmas as below.

Lemma 1. *If* $\lim_{n\to\infty} a_n = a$ *then* $\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} a_i = a$.

Lemma 2. $S_b + S_w = C$.

The detailed proof of lemma could be found in Appendix A. From Lemma 2, the maximum margin criterion can be transformed as $J(W) = W^{\mathrm{T}}(2S_b - C)W$. In the following, the *variable* (i) denotes the value of *variable* at step i. From Lemma 1 we know that

$$S_b = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} S_b(i). \tag{1}$$

On the other hand,

$$\begin{aligned} C &= E\{(u(n) - m)(u(n) - m)^{\mathrm{T}}\} \\ &\doteq \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} (u(n) - m(n))(u(n) - m(n))^{\mathrm{T}}. \end{aligned} \tag{2}$$

Assume that θ is a positive real number and $I \in R^{d\times d}$ is an identity matrix, if λ is an eigenvalue of matrix A and x is its corresponding eigenvector, then $(A + \theta I)x = Ax + \theta Ix = (\lambda + \theta)x$, i.e., A should have the same eigenvectors as matrix $A + \theta I$. The convergence of our algorithm [3] needs the MMC criterion matrix to be nonnegative determined, thus a parameter θ is designed to meet this requirement. Therefore, $2S_b - C$ should have the same eigenvectors as $2S_b - C + \theta I$.

Notice that θI can be represented as

$$\theta I = E\{\theta I\} = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \theta I. \tag{3}$$

From (1)–(3) we can have

$$\begin{aligned} 2S_b - C + \theta I &= \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} (2S_b(i) - (u(i) - m(i))(u(i) - m(i))^{\mathrm{T}} + \theta I) \\ &= \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} A(i) = A, \end{aligned} \tag{4}$$

where

$$A(i) = 2S_b(i) - (u(i) - m(i))(u(i) - m(i))^{\mathrm{T}} + \theta I,$$
$$A = 2S_b - C + \theta I. \tag{5}$$

Notice that $E\{A(n)\} = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} A(i)$.

The general eigenvector form is $Ax = \lambda x$, where x is the eigenvector of matrix A corresponding to eigenvalue λ. By replacing matrix A with the maximum margin criterion matrix at step n, we obtain an approximate iterative eigenvector computation formulation with $v = \lambda x$:

$$\begin{aligned} v(n) &= \frac{1}{n}\sum_{i=1}^{n}\Big(2S_b(i) - (u(i) - m(i))(u(i) - m(i))^{\mathrm{T}} + \theta I\Big)x(i) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left(2\sum_{j=1}^{c} p_j(i)\Phi_j(i)\Phi_j(i)^{\mathrm{T}} - (u(i) - m(i))(u(i) - m(i))^{\mathrm{T}} + \theta I\right)x(i), \end{aligned} \tag{6}$$

where $\Phi_j(i) = m_j(i) - m(i)$. Let $x(i) = v(i-1)/\|v(i-1)\|$, we have the following incremental formulation:

$$\begin{aligned} v(n) = {} & \frac{n-1}{n}v(n-1) \\ & + \frac{1}{n}\left(2\sum_{j=1}^{c} p_j(n)\Phi_j(n)\Phi_j(n)^{\mathrm{T}} - (u(n) - m(n))(u(n) - m(n))^{\mathrm{T}} + \theta I\right) \\ & \times v(n-1)/\|v(n-1)\| = \frac{n-1}{n}v(n-1) \\ & + \frac{1}{n}\left(2\sum_{j=1}^{c} p_j(n)\alpha_j(n)\Phi_j(n) - \beta(n)(u(n) - m(n)) + \theta v(n-1)\right) \\ & \Big/\|v(n-1)\|, \end{aligned} \tag{7}$$

where $\alpha_j(n) = \Phi_j(n)^{\mathrm{T}}v(n-1)$ and $\beta(n) = (u(n) - m(n))^{\mathrm{T}}v(n-1)$. $v(n)$ is the nth step estimation of v. For initialization, we set $v(0)$ as the first sample.

Notice that different eigenvectors are orthogonal to each other. Thus it helps to generate "observations" only in a complementary space for the computation of the higher order eigenvectors. To compute the $(j+1)$th eigenvector, we first subtract its projection on the estimated jth eigenvector from the data,

$$u_{l_n}^{j+1}(n) = u_{l_n}^{j}(n) - (u_{l_n}^{j}(n)^{\mathrm{T}}v^{j}(n))v^{j}(n)/\|v^{j}(n)\|^2, \tag{8}$$

where $u_{l_n}^{1}(n) = u_{l_n}(n)$. The same method is used to update $m_i^j(n)$ and $m^j(n)$, $i = 1, 2, \ldots, c$. Since $m_i^j(n)$ and $m^j(n)$ are linear combinations of $x_{l_i}^j(i)$, where

$i = 1,2,\ldots,n$, $j = 1,2,\ldots,k$, and $l_i \in \{1,2,\ldots,C\}$, Φ_i are linear combination of m_i and m, for convenience, we can only update Φ at each iteration step by

$$\Phi_{l_n}^{j+1}(n) = \Phi_{l_n}^{j}(n) - (\Phi_{l_n}^{j}(n)^{\mathrm{T}} v^j(n)) v^j(n) / \|v^j(n)\|^2. \tag{9}$$

In this way, the time-consuming orthonormalization is avoided and the orthogonal is always enforced when the convergence is reached, although not exactly so at early stages. Through the projection procedure using (8) and (9) at each step, we can get the eigenvectors of maximum margin criterion matrix one by one. It is much more efficient in comparison to the time-consuming orthonormalization process.

3.1.2. Algorithm summary

Suppose that $u(n)$ is the input sample at step n, which belongs to class l_n, $l_n \in \{1,2,\ldots,c\}$, $N_i(n)$ is the total sample number of class i. $m_i(n)$ is the mean of class i. $m(n)$ is the mean of all samples. Set $\Phi_j(i) = m_j(i) - m(i)$. The algorithm is listed in Table 2 and we can update our data at each step as that in Table 3.

The time complexity of IMMC to train N input samples is $O(Ncdp)$, where c is the number of classes, d is the dimension of the original data space, and p is the target dimension, which is linear with each factor. Furthermore, when handling each input sample, IMMC only need to keep the learned eigen-space and several first-order statistics of the past samples, such as the mean and sample numbers for different classes. Hence, IMMC is able to handle large scale and continuous data.

Table 2
IMMC algorithm

for $n = 1,2,\ldots,$ do the following steps,
  Update $N_i(n)$, $m_i(n)$, $\Phi_i(n)$, $m(n)$ following the steps in Table 3;
  $\Phi_i^1(n) = \Phi_i(n), i = 1,2,\ldots,c$
  for $j = 1,2,\ldots,\min(p,n)$
    if $j = n$ then
      $v^j(n) = u^j(n)$
    else

$$\{\alpha_i^j(n) = \Phi_i^j(n)^{\mathrm{T}} v^j(n-1)$$
$$\beta^j(n) = u^j(n)^{\mathrm{T}} v^j(n-1)$$
$$v^j(n) = \frac{n-1}{n} v^j(n-1) + \frac{1}{n}\left(2\sum_{i=1}^{c} p_i(n)\alpha_i^j(n)\Phi_i^j(n) - \beta^j(n)u^j(n) + \theta(j)v^j(n-1)\right) \Big/ \|v(n-1)\|$$
$$\Phi_i^{j+1}(n) = \Phi_i^j(n) - \Phi_i^j(n)^{\mathrm{T}} v^j(n) v^j(n) / \|v^j(n)\| \|v^j(n)\|$$
$$u_i^{j+1}(n) = u_i^j(n) - u_i^j(n)^{\mathrm{T}} v^j(n) v^j(n) / \|v^j(n)\| \|v^j(n)\|$$
$$\text{where } p_j(n) = N_i(n)/n\}$$

end for

Table 3
Update process

$m(n) = \frac{1}{n}((n-1)m(n-1) + u(n))$
for $i = 1, 2, \ldots, c$
  if $i = l_n$

$$\{N_i(n) = N_i(n-1) + 1$$
$$m_i(n) = \frac{1}{N_i(n)}(N_i(n-1)m_i(n-1) + u(n))\}$$

  else

$$\{N_i(n) = N_i(n-1)$$
$$m_i(n) = m_i(n-1)\}$$

  $\Phi_i(n) = m_i(n) - m(n)$
  $u(n) = u(n) - m(n)$
end for

Notice that IMMC algorithm relies on a parameter θ and it sometimes could not be estimated beforehand accurately. In these cases, it is assigned as experiential value. These difficulties motivate us to propose a weighted maximum margin criterion $A = S_b - \varepsilon S_w$, where ε is a real parameter. Some advanced experiments show that the classical MMC ($\varepsilon = 1$) is not usually optimal for classification tasks. In other words, a proper ε could improve the performance of MMC and it could make sure that the criterion matrix is nonnegative determined. Then we could make the criterion matrix nonnegative determined by giving a proper ε instead of parameter θ.

3.2. Incremental inter-class scatter criterion

Since the data arrive in a high rate in streaming data model, the simplicity of the criterion for supervised dimensionality reduction algorithm is very important. The inter-class scatter criterion ($\varepsilon = 0$) aims to make the class centers as far as possible. The speed and scale of data in the streaming data model is very high, thus this criterion satisfies the simplicity requirement. The inter-class scatter criterion is as follows: $J(W) = W^{\mathrm{T}} S_b W$, where S_b is the inter-class scatter matrix same as in LDA. In the above formulation, we exercised freedom to multiply W with some nonzero constant. Thus, we additionally require that W consists of unit vectors, i.e., $W = [w_1, w_2, \ldots, w_p]$ and $w_k^{\mathrm{T}} w_k = 1$. Then, the optimization problem is transformed to the following constrained optimization problem:

$$\max \sum_{k=1}^{p} w_k^{\mathrm{T}} S_b w_k, \quad \text{subject to } w_k^{\mathrm{T}} w_k = 1, \ k = 1, 2, \ldots, p.$$

We maximize the above optimization problem by introducing a Lagrangian function below,

$$L(w_k, \lambda_k) = \sum_{k=1}^{p} w_k^{\mathrm{T}} S_b w_k - \lambda_k (w_k^{\mathrm{T}} w_k - 1),$$

where λ_k are the Lagrange multipliers. The condition for the saddle point is that the derivatives of L must vanish, which leads to $S_b w_k = \lambda_k w_k$, i.e., the columns of W are eigenvectors of S_b. Therefore, $J(W)$ is maximized when W is composed of the first d leading eigenvectors of S_b.

In the following subsections, we will present the details on how to incrementally derive the leading eigenvectors of S_b, i.e., the Incremental Inter-class Scatter criterion (IIS); then the algorithm summary are also presented.

3.2.1. Algorithm derivation

As in Section 3.1, assume that a sample sequence is presented as $\{u_{l_n}(n)\}$, where $n = 1, 2, \ldots$ The purpose of IIS is to maximize the inter-class scatter criterion $J_s(W) = W^{\mathrm{T}} S_b W$.

The inter-class scatter matrix of step n after learning from the first n samples can be written as

$$S_b(n) = \sum_{j=1}^{c} p_j(n)(m_j(n) - m(n))(m_j(n) - m(n))^{\mathrm{T}}. \tag{10}$$

From the fact that $\lim_{n\to\infty} S_b(n) = S_b$ and Lemma 1, we obtain

$$S_b = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} S_b(i). \tag{11}$$

The general eigenvector form is $Au = \lambda u$, where u is the eigenvector corresponding to the eigenvalue λ. By replacing the matrix A with the inter-class scatter matrix at step n, we can obtain an approximate iterative eigenvector computation formulation with $v = \lambda u$:

$$\begin{aligned} v(n) &= \frac{1}{n} \sum_{i=1}^{n} S_b(i) u(i) \\ &= \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} p_j(i)(m_j(i) - m(i))(m_j(i) - m(i))^{\mathrm{T}} u(i) \\ &= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{c} p_j(i) \Phi_j(i) \Phi_j(i)^{\mathrm{T}} \right) u(i), \end{aligned} \tag{12}$$

where $\Phi_j(i) = m_j(i) - m(i)$, $v(n)$ is the nth step estimation of v and $u(n)$ is the nth step estimation of u. The convergence proof below tells us that once we obtain

the estimate of v, eigenvector u can be directly computed as $u = v/\|v\|$. Let $u(i) = v(i-1)/\|v(i-1)\|$, we have the following iterative formulation:

$$v(n) = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{c} p_j(i)\Phi_j(i)\Phi_j(i)^{\mathrm{T}}\right)v(i-1)\Big/\|v(i-1)\|, \tag{13}$$

i.e.,

$$\begin{aligned} v(n) &= \frac{n-1}{n}v(n-1) + \frac{1}{n}\left(\sum_{j=1}^{c} p_j(n)\Phi_j(n)\Phi_j(n)^{\mathrm{T}}\right)v(n-1)\Big/\|v(n-1)\| \\ &= \frac{n-1}{n}v(n-1) + \frac{1}{n}\sum_{j=1}^{c} p_j(n)\alpha_j(n)\Phi_j(n), \end{aligned} \tag{14}$$

where $\alpha_j(n) = \Phi_j(n)^{\mathrm{T}}v(n-1)/\|v(n-1)\|$, $j = 1, 2, \ldots, c$.

For initialization, we set $v(0)$as the first sample. The computation of other eigenvectors is the same as that of IMMC in Section 3.1

3.2.2. Algorithm summary

Suppose that $N_i(n)$ is the total sample number of class i at step n. The solution of step n is $v^j(n)$, $j = 1, 2, \ldots, p$. We update each element as its counterpart in Section 3.1 and the algorithm is listed in Table 4.

The time complexity of IIS to train a input samples is O(cdp), where c is the number of classes, d is the dimension of the original data space, and p is the target dimension, which is linear with each factor. Furthermore, when handling each input sample, IIS only need to keep the learned eigen-space and several

Table 4
IIS algorithm

for $n = 1, 2, \ldots,$ do the following steps,
  Update $N_i(n), m_i(n), \Phi_i(n), m(n)$
  $\Phi_i^1(n) = \Phi_i(n) i = 1, 2, \ldots, c$
  for $j = 1, 2, \ldots, \min\{p, n\}$
    if $j = n$ then
      $v_j(n) = u_i^j(n)$
    else
      $\alpha_i^j(n) = \Phi_i^j(n)^{\mathrm{T}} v^j(n-1)$
      $v^j(n) = \frac{n-1}{n}v^j(n-1) + \frac{1}{n}\sum_{i=1}^{c}\alpha_i^j(n)p_i(n)\Phi_i^j(n)$
      $\Phi_i^{j+1}(n) = \Phi_i^j(n) - \Phi_i^j(n)^{\mathrm{T}} v^j(n) v^j(n)/\|v^j(n)\|\|v^j(n)\|$
      $u_i^{j+1}(n) = u_i^j(n) - u_i^j(n)^{\mathrm{T}} v^j(n) v^j(n)/\|v^j(n)\|\|v^j(n)\|$
end for

first-order statistics of the past samples, such as the mean and the counts. Hence, IIS is suitable for streaming data model. Moreover, IIS is also robust since IIS focuses on the mean of each class and all samples.

Both IIS and IMMC are general in the sense that they maximize the distance among different class centers and minimize the distance among sample data of the same class simultaneously. The only difference between the two algorithms is the choice of weight ε of weighted MMC $A = S_b - \varepsilon S_w$. If $\varepsilon = 0$, there has no parameter choice problem to identify the criterion matrix to be nonnegative determined, thus the incremental algorithm should be more stable. Moreover, the algorithm will be more efficient due to the simpler expression of criterion matrix. On the other hand, if $\varepsilon = 1$, the criterion is more reasonable for supervised dimensionality reduction and should be more effective for classification tasks since it considers not only the inter-class information, but also the intra-class information. Due to this reason, the choice of algorithms depends on the real task.

4. Related works

Incremental learning for dimensionality reduction on streaming data has attracted much attention in the last few decades. Incremental PCA is a well studied incremental leaning algorithm. Many types of IPCA algorithms have been proposed and the main difference is the incremental representation of the covariance matrix. Almost all of them base on the Singular Value Decomposition. The latest version of IPCA with convergence proof which does not base on Singular Value Decomposition is called Candid Covariance-free Incremental Principal Component Analysis (CCIPCA) which does not need to reconstruct the covariance matrix at each iteration step. Furthermore, an Incremental Linear Discriminant Analysis (ILDA) algorithm has also been proposed recently.

4.1. Incremental PCA by singular value decomposition

The algorithms for incremental principal component analysis [1] have been proposed since 1970s. Most of them are based on the Singular Value Decomposition (SVD). The main difference among the several SVD-based IPCA algorithms is how to express the covariance matrix incrementally. The incremental algorithms by SVD are not applicable to streaming data model since the computational complexity is very high.

4.2. Candid covariance-free IPCA

CCIPCA [23] is a recently proposed algorithm for unsupervised incremental dimensionality reduction. It incrementally computes the principal components

of a sequence of samples without estimating the covariance matrix (thus covariance-free). The new method is motivated by the concept of statistical efficiency (the estimation has the smallest variance given the observed data). To do this, it keeps the number of observations and computes the mean of observations incrementally, which is an efficient estimate for some well known distributions (e.g., Gaussian), although the highest possible efficiency is not guaranteed in the case because of unknown sample distribution. It converges very fast for high-dimensional data. The convergence of this algorithm is proved in [12,28].

4.3. Incremental linear discriminant analysis

The ILDA algorithm [9] proposed recently is an effective supervised dimensionality reduction algorithm. It is based on the theory of neural network which is a Gradient Descent algorithm. Since LDA has some shortcomings which limit its applications, we proposed the above two novel supervised dimensionality reduction algorithms to avoid the singularity problem and the limitation of the available dimension of the space.

5. Experimental results

In order to test the performance of the incremental algorithms on streaming data, we modify some commonly used datasets as streaming data. We conducted three sets of experiments. In the first set of experiments, we used synthetic data that follows the normal distribution to illustrate the low-dimensional space learned by IIS, IMMC, LDA and PCA. In the second set of experiments, we applied several dimensionality reduction methods on the Reuters Corpus Volume 1 (RCV1) dataset, whose dimension is about 300,000, and then compare the classification performance and the time cost. This experiment is used to demonstrate the classification performance of IIS on very large scale dataset. In the third set of experiments, we applied IIS and IMMC on some suitable subsets of UCI datasets to demonstrate the convergence performance of IIS and IMMC. Moreover, experiments on these UCI datasets are also used to compare the classification error rate on the low-dimensional space found by different dimensionality reduction algorithm.

5.1. Synthetic dataset

We generated the synthetic data by normal distribution. Fig. 2 shows a scatter plot of the data set. The stars are two-dimensional data points belonging to class 1, and the triangles are two-dimensional data points belonging to class 2. The one-dimensional space direction found by IMMC is the same as that of

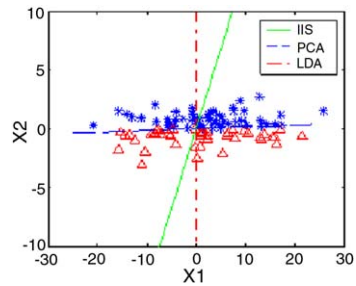

Fig. 2. Low-dimensional spaces learned by IIS, LDA and PCA, respectively.

LDA and not plotted in Fig. 2 to avoid overlapping. We can see from Fig. 2 that the two classes of data points are mixed together in the one-dimensional space found by PCA; meanwhile, data points of different classes are separated well in the one-dimensional space found by LDA and MMC. Moreover, the one-dimensional IIS space is close to its counterpart found by LDA and IMMC. In a word, IIS can outperform PCA and yield comparable performance to LDA for classification. Moreover, IMMC can find a very similar low-dimensional space with LDA.

5.2. Real world datasets

The Reuters Corpus Volume 1 (RCV1) [13] data set which contains over 800,000 documents with 300,000 dimensions and some suitable subsets of UCI [4] are used in the real world data experiments to evaluate the effectiveness and efficiency of IIS in this subsection.

5.2.1. RCV1

To compare the effectiveness and efficiency of IIS with that of other dimensionality reduction algorithms, we conducted classification experiments on the Reuters Corpus Volume 1 (RCV1) [13] data set which contains over 800,000 documents (300,000-dimension).

5.2.1.1. Experiments setup. We choose the data samples with the highest four topic codes (CCAT, ECAT, GCAT, and MCAT) in the "Topic Codes" hierarchy, which contains 789,670 documents. Then we split them into five equal-sized subsets, and each time four of them are used as the training set and the remaining ones are for test set. The experimental results reported in this paper are the average of the five runs. The experiment consists of the following steps:

- Applying the dimension reduction algorithm on a specific size of training data to learn a low-dimensional presentation.

- Transforming all the training data to the low-dimensional presentation.
- Training SVM by SMO [21].
- Transforming all the test data to the low-dimensional space.
- Evaluate the classification performance, using F1 value, on the transformed test data.And the dimension reduction algorithms applied are:
- The proposed IIS generating a 3-d space. We applied IIS on the first 10, 100, 1,000, 10,000, and 100,000 training data to study the convergence speed.
- Information Gain (IG). This is a state-of-the-art text classification method [26]. In this paper, we applied IG on all training data to generate 3-d and 500-d spaces, denoted by IG3 and IG500, respectively. Note IG500 will yields almost the best classification performance, since SVM is insensitive to the number of feature [27].
- IPCA following the CCIPCA algorithm [23].

5.2.1.2. Effectiveness of IIS. The RCV1 is considered as a data streaming. IIS learns from it sample by sample. The classification performance comparison is summarized in Fig. 3. From it, we can conclude that the eigen-space learned by IIS on 100 input samples is significantly better than the ones learned by IPCA and IG3; and after learning 100,000 input samples (<20%), IIS can generate a comparable eigen-space to the one generated by IG500 in terms of classification performance. Hence, IIS is an effective dimensionality reduction algorithm for classification tasks. On the other hand, we can see that IIS generated a near optimal eigen-space after just learning 10,000 samples, so the convergence speed of IIS is very fast.

5.2.1.3. Efficiency of IIS. The times of each algorithm spent in dimension reduction and classification training are reported in Table 5. We can see that the dimension reduction time of IIS is almost linear related to the number of

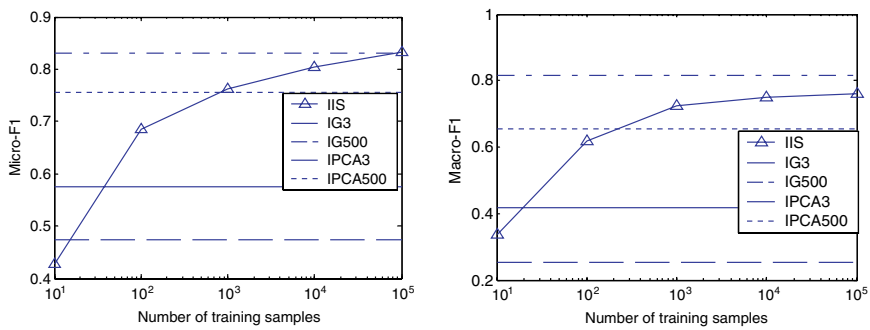

Fig. 3. Micro-F1 and macro-F1 after reducing dimension by several dimensionality reduction algorithms.

Table 5
Time cost (s) for each dimension reduction algorithm

	Dimension reduction	Classification
IIS 3@10	1.85	1.298
IIS 3@100	17.1	11.061
IIS 3@1000	177	6474
IIS 3@10000	2288	7560
IIS 3@100000	26,884	3343
IG 3@all	136	52,605
IG 500@all	137	312,887
IPCA 3@all	28,960	25,374
IPCA 500@all	3,763,296	9327

input samples. Although the IG is faster than IIS, its classification time is much longer than that of IIS.

5.2.2. UCI data

UCI machine learning dataset [4] is a repository of databases, domain theories and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. We use this data set mainly for demonstrating the convergence performance of our two algorithms. The UCI data set include over 80 subsets, the subsets chosen by us must satisfy: (1) numerical valued; (2) without missing value; and (3) labeled, this is due to the reason that our algorithm is designed for supervised dimensionality reduction. We choose six available datasets for experiments to demonstrate the convergence performance of our proposed algorithm.

(1) *IRIS Data Set*
Since the amount of this data set is not very large (50 data in each of three classes), we use it to demonstrate that our incremental algorithm could converge very fast within a little amount of training data. We reordered the dataset to mix different classes. In other words, at each iteration step, we choose a sample from some class of the three randomly with equal probability. For this IRIS data, the eigenvalues of $A = 2S_b - C$ are -0.2133, -0.0571, -0.0222 and 3.6396. We choose $\theta = 0.3$ to make sure the criterion matrix is nonnegative determined.

(2) *Balance Scale Data*
Balance Scale Data was generated to model psychological experimental results. This data set is very special since the covariance matrix of it is an identity matrix. Note that PCA could not work well under this condition. For this Balance Scale data set, the eigenvalues of $A = 2S_b - C + \theta I$ are -2.0, -2.0, -1.9774, and 0.7067. We choose $\theta = 2.0$ to make sure the criterion matrix is nonnegative determined.

(3) *Wine Recognition Data*
These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. This dataset has been used to test the comparison of classifiers in high-dimensional settings and many other applications.

(4) *Waveform Database*
The original MMC criterion matrix on this data set is negative determined. We have an very interesting observation by experimental results that if the maximum margin criterion matrix is negative determined, IMMC converge very slowly or even could not converge through the limited number of samples available in this dataset no matter how we to choose the parameter. We will explain this phenomenon in our future work.

(5) *The Monk's Problem*
The Monk's Problem was the basis of a first international comparison of learning algorithms. There are several Monk's problems. The domains for all Monk's problems are the same. One of the Monk's problems has noise added. For each problem, the domain has been partitioned into a train and test set. We choose the first one and combine the training data and the testing data to plot the convergence curve.

(6) *Pima Indians Diabetes Database*
The original owners of this dataset are National Institute of Diabetes and Digestive and Kidney Diseases. Pima Indians Diabetes Database includes 768 samples, two classes and eight attributes for each sample.
Since $\|v - v'\| = 2(1 - v \cdot v')$, and $v = v'$ iff $v \cdot v' = 1$, the correlation between two unit eigenvectors is represented by their inner product, and the larger the inner product is, the more similar the two eigenvectors are. We should like to plot the inner product curve between incremental approaches and their corresponding batch approaches to represent the convergence curve. Fig. 4 shows the convergence curve of different dimensionality reduction algorithms on all the six UCI datasets introduced above. Proper parameters are given to (a), (b), (c) and (d) when performing IMMC. We choose a popular Incremental PCA algorithm, CCIPCA as our baseline in these experiments. From Fig. 4 we can see that IIS converges very fast and is very stable. On the other hand, IMMC could converge very fast if a proper parameter is given. However, (e) and (f) tell us that if the criterion matrix is negative determined, we could not choose the proper parameter, and then IMMC could not converge as expected.

Following the convergence experiments, we conducted some more experiments on these UCI data sets to test the classification error rate on the learned low-dimensional space by nearest neighbor classifier. As showed in (e) and (f) of Fig. 4, the IMMC could not converge stably on these two sets, we could not

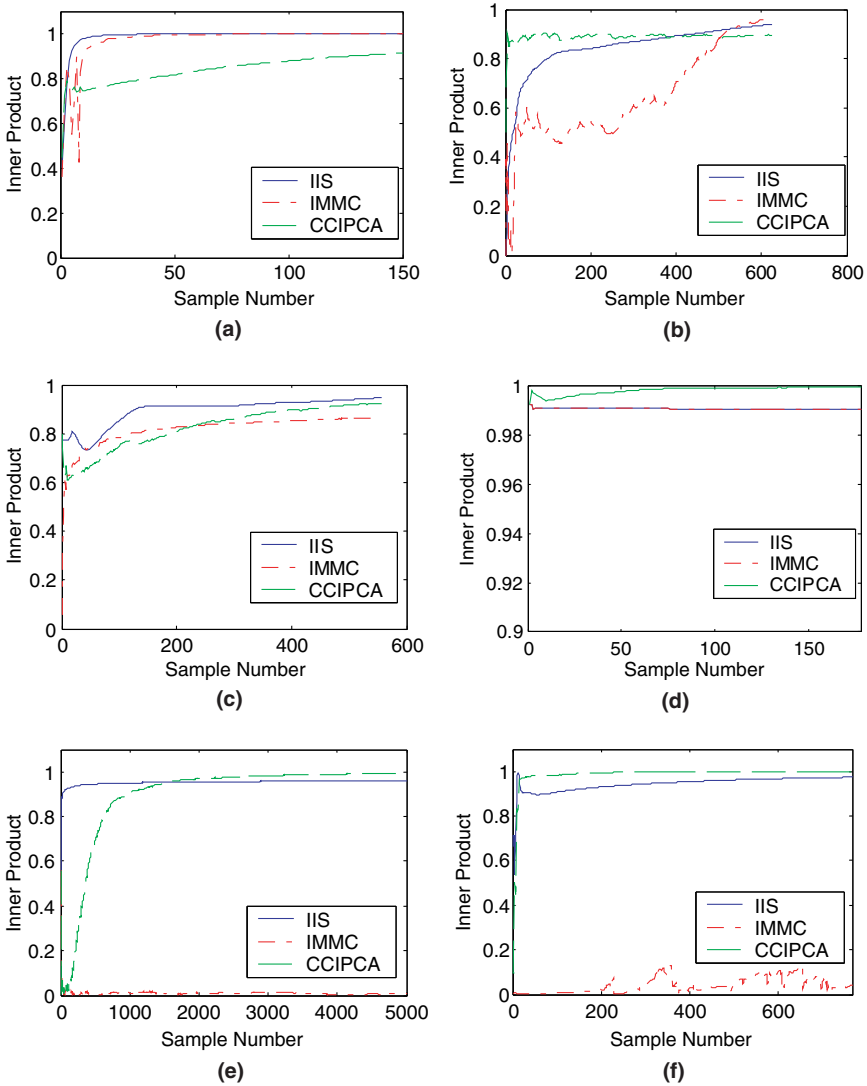

Fig. 4. Performance of different dimensionality reduction algorithms: (a) IRIS; (b) Balance Scale; (c) The Monk's Problem; (d) Wine Recognition Data; (e) Waveform Database; (f) Pima Indians Diabetes Database. Note that IIS is stable on all the six data sets. CCIPCA also perform well. IMMC in (e) and (f) could not converge to the object vector as the maximum eigenvalue of these two data sets are negative.

show the error rate pictures on these two datasets. For each UCI dataset that do not provide training-testing split, we used repeated holdout methods by repeatedly separating them into two folds randomly. IMMC, PCA and LDA

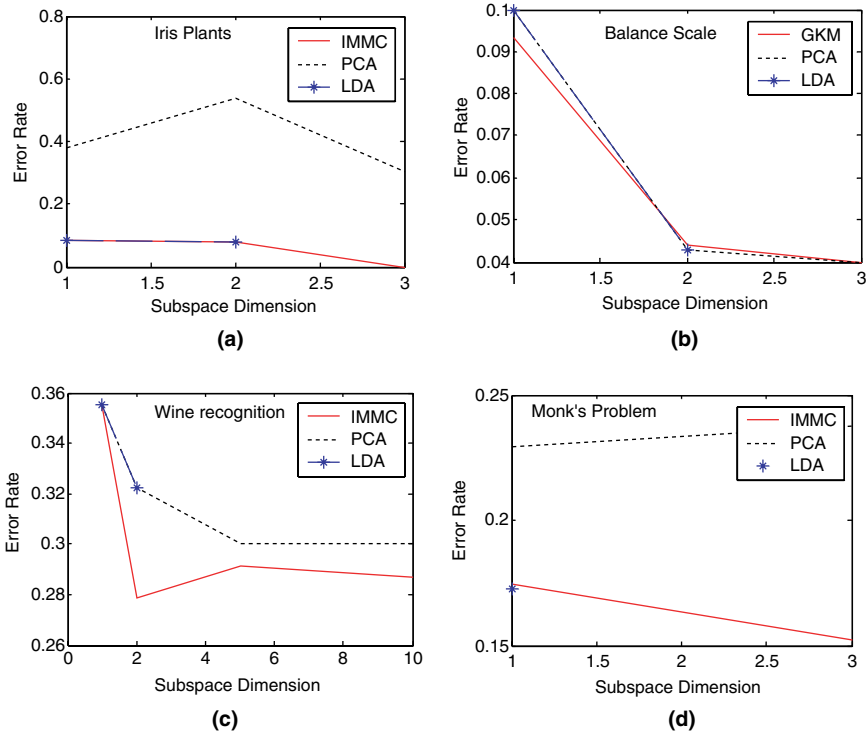

Fig. 5. Performance comparison of different dimensionality reduction algorithms. (Note that IMMC can outperform PCA and is comparable to LDA. PCA and LDA are conducted by batch approaches using Singular Value Decomposition.)

are then applied to the training data to find the lower dimensional space. The Nearest Neighbor classifier (NN) is used to classify these testing data. Fig. 5 shows the error rates of IMMC, PCA and LDA on these UCI subsets. It can be seen that PCA is not optimal for classification tasks and LDA are limited by the number of classes. On the other hand IMMC could outperform them in terms of classification error rate.

6. Conclusions and future work

To meet the challenging issue of computing dominating eigenvectors from incrementally arriving data stream, in this paper, we proposed an incremental supervised dimensionality reduction algorithm called Incremental Maximum Margin Criterion (IMMC) and its efficient and stable extension, Incremental Inter-class Scatter (IIS). In contrast to the most popular dimensionality reduc-

tion approaches on streaming data, IMMC and IIS are supervised algorithms, and therefore, more suitable for classification tasks. The proposed algorithms are fast in terms of the convergence rate and low in the computational complexity. Experimental results on synthetic dataset and real text dataset demonstrated that they can outperform IPCA on classification tasks. The convergence property of IIS is better than IMMC due to its simple criterion. On the other hand, the classification performance of IMMC should be better than IIS in theory. Both of them are applicable under a same general framework. In the future, we wish to continue to explore the dimensionality reduction algorithm for the data streams such as streaming multimedia. Moreover, we plan to design some automatic algorithm to learn the parameter of IMMC.

Acknowledgements

We gratefully appreciate the anonymous reviewers' comments and the suggestions of Huan Liu to improve the presentation of the paper. We also thank the support of Fengshan Bai from Department of Mathematical Science, Tsinghua University, PR China.

Appendix A

Lemma 2. $S_b + S_w = C$.

Proof. By simple algebra formulation, we can rewrite the matrix S_w as follows: (For convenience, we do not consider the constant $1/n$ in our proof.)

$$S_w = \sum_{j=1}^{C} N_j E[(u_j - m_j)(u_j - m_j)^{\mathrm{T}}] = \sum_{j=1}^{C} \left(\sum_{i=1}^{N_j} u_j(i) u_j(i)^{\mathrm{T}} - N_j m_j m_j^{\mathrm{T}} \right)$$
$$= \sum_{j=1}^{C} U_j L_j U_j^{\mathrm{T}},$$

where $U_j L_j U_j^{\mathrm{T}}$ is the data covariance matrix of the jth class, $U_j = [u_j(1), u_j(2), \ldots, u_j(N_j)]$, $L_j = I - \frac{1}{N_i} e_i e_i^{\mathrm{T}}$ where I is the identity matrix and $e_j = (1, 1, \ldots, 1)^{\mathrm{T}}$ is N_j dimension vector. For further simplification, set $U = (u(1), u(2), \ldots, u(n))$, for $u_{l_i}(i)$ and $u_{l_j}(j)$ if

$$W_{ij} = \begin{cases} 1 & l_i = l_j, \\ 0 & l_i \neq l_j. \end{cases}$$

Denote $L = I - W$, then we have $S_w = ULU^{\mathrm{T}}$.

Similarly, we can compute the matrix S_b as follows:

$$S_b = \sum_{j=1}^{C} N_j(m_j - m)(m_j - m)^{\mathrm{T}} = UWU^{\mathrm{T}} - nmm^{\mathrm{T}} = U\left(W - \frac{1}{n}ee^{\mathrm{T}}\right)U^{\mathrm{T}}$$

$$= -ULU^{\mathrm{T}} + U\left(I - \frac{1}{n}ee^{\mathrm{T}}\right)U^{\mathrm{T}} = -S_w + C,$$

where $e = (1, 1, \ldots, 1)^{\mathrm{T}}$ is a n dimension vector and

$$C = U\left(I - \frac{1}{n}ee^{\mathrm{T}}\right)U^{\mathrm{T}}$$

is the data covariance matrix which is independent to the labels. □

References

[1] M. Artae, M. Jogan, A. Leonardis, Incremental PCA for on-line visual learning and recognition, in: Proceeding of the 16th International Conference on Pattern Recognition, Quebec City, QC, Canada, 2002.

[2] B. Babcock, Models and issues in data stream systems, in: Proceeding of the ACM PODS, 2002.

[3] S. Balakrishnama, A. Ganapathiraju, Linear Discriminant Analysis—A brief Tutorial, Institute for Signal and Information Processing, Mississippi, 1998.

[4] C.L. Blake, C.J. Merz, UCI Repository of machine learning databases Irvine, University of California, Department of Information and Computer Science, California, 1998.

[5] A. Broder, M. Mitzenmacher, Network applications of bloom filters: a survey, in: Proceeding of the Allerton Conference, 2002.

[6] M. Charikar, K. Chen, M. Farach-Colton, Finding frequent items in data streams, in: Proceeding of the ICALP, 2002.

[7] A. Gilbert et al., Surfing wavelets on streams: one pass summaries for approximate aggregate queries, VLDB Journal (2001) 79–88.

[8] A. Globerson, G. Chechik, N. Tishby, Sufficient dimensionality reduction with irrelevance statistics, in: Proceeding of the 19th Conference on Uncertainty in Artificial Intelligence, Acapulco, Mexico, 2003.

[9] K. Hiraoka et al., Convergence analysis of online linear discriminant analysis, International Joint Conference on Neural Networks IJCNN'00 3 (2000) 3387.

[10] P. Indyk, Stable distributions, pseudorandom generators, embeddings and data stream computation, IEEE FOCS (2000) 189–197.

[11] S. Khanna, M. Greenwald, Space-efficient online computation of quantile summaries, in: Proceeding of the ACM SIGMOD, 2001.

[12] H.J. Kushner, D.S. Clark, Stochastic Approximation Methods for Constrained and Unconstrained System, Springer-Verlag, New York, 1978.

[13] D. Lewis et al., RCV1: a new benchmark collection for text categorization research, Journal of Machine Learning Research 5 (2004) 361–397.

[14] H. Li, T. Jiang, K. Zhang, Efficient and robust feature extraction by maximum margin criterion, in: Proceeding of the Advances in Neural Information Processing Systems 16, MIT Press, Vancouver, Canada, 2004.

[15] G. Manku, R. Motwani, Approximate frequency counts over data streams, in: Proceeding of the VLDB, 2002.

[16] A.M. Martinez, A.C. Kak, PCA versus LDA, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2) (2001) 228–233.
[17] N. Mishra, D. Oblinger, L. Pitt, Sublinear time approximate clustering, in: Proceeding of the ACM-SIAM SODA, 2001.
[18] S. Muthukrishnan, Data Stream Algorithms and Applications, Rutgers/AT&T Shannon Labs, 2004.
[19] L. OCallaghan et al., Streaming-data algorithms for high-quality clustering, in: Proceeding of the International Conference on Data Engineering (ICDE), 2002.
[20] P. Gibbons, Distinct sampling for highly-accurate answer to distinct values queries and event reports in VLDB, 2001.
[21] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 185–208.
[22] R. Motwani, S. Chaudhuri, V. Narasayya, Random sampling for histogram construction: how much is enough? SIGMOD (1998) 436–447.
[23] J. Weng, Y. Zhang, W.S. Hwang, Candid covariance-free incremental principal component analysis, IEEE Transactions of the Pattern Analysis and Machine Intelligence 25 (8) (2003) 1034–1040.
[24] H. Wong, W. Fan, P. Yu, J. Han, Mining concept-drifting data streams using ensemble classifiers, in: Proceeding of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, 2003.
[25] J. Yan, B.Y. Zhang, S.C. Yan, Z. Chen, W.G. Fan, Q. Yang, W.Y. Ma, Q.S. Cheng, IMMC: Incremental maximum marginal criterion, in: Proceeding of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 2004.
[26] Y. Yang, J.O. Pedersen, A comparative study on feature selection in text categorization, in: Proceeding of the 14th International Conference on Machine Learning, 1997.
[27] J. Zhang et al., Modified logistic regression: an approximation to SVM and its applications in large-scale text categorization, in: Proceeding of the 20th International Conference on Machine Learning, Washington, DC, 2003.
[28] Y. Zhang, J. Weng, Convergence analysis of complementary candid incremental principal component Analysis, Technical Report, MSU-CSE-01-23, Computer Science and Engineering Department of Michigan State University, August, 2001.