



Incremental learning from chunk data for IDR/QR[☆]



Gui-Fu Lu^{*}, Jian Zou, Yong Wang

School of Computer and Information, Anhui Polytechnic University, Wuhu, Anhui 241000, China

ARTICLE INFO

Article history:

Received 25 September 2013

Received in revised form 15 September 2014

Accepted 5 January 2015

Available online 19 February 2015

Keywords:

Feature extraction

Dimensionality reduction

Linear discriminant analysis

Small sample size problem

Incremental learning

ABSTRACT

IDR/QR, which is an incremental dimension reduction algorithm based on linear discriminant analysis (LDA) and QR decomposition, has been successfully employed for feature extraction and incremental learning. IDR/QR can update the discriminant vectors with light computation when new training samples are inserted into the training data set. However, IDR/QR has two limitations: 1) IDR/QR can only process new samples one instance after another even if a chunk of training samples is available at a time; and 2) the approximate trick is used in IDR/QR. Then there exists a gap in performance between incremental and batch IDR/QR solutions. To address the problems of IDR/QR, in this paper, we propose a new chunk IDR method which is capable of processing multiple data instances at a time and can accurately update the discriminant vectors when new data items are added dynamically. Experiments on some real databases demonstrate the effectiveness of the proposed algorithm over the original one.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Linear discriminant analysis (LDA) [1] is a well-known supervised feature extraction algorithm for pattern recognition. LDA aims to find a set of projection vectors to maximize the between-class scatter matrix and minimize the within-class scatter matrix at the same time. In many applications, however, LDA often suffers from the so-called small sample size (SSS) problem or undersampled problems [2], where the number of samples is smaller than the dimensionality of the feature space. To address this problem, many efficient LDA-based methods, e.g. principal component analysis (PCA) + LDA [3], null space based LDA (NLDA) [4–6], LDA/GSVD [7], complete CLDA (CLDA) [8], and dual-space LDA (DSLDA) [9], have been proposed to deal with the high-dimensional data and learn the optimal discriminant vectors.

PCA + LDA, also called Fisherfaces, may be the most popular method to address the SSS problem. To make the within-class scatter nonsingular, it first uses PCA to reduce the dimensionality of samples to $n - c$, where n is the number of the training samples and c is the number of the classes, before the application of LDA. Since the smallest $c - 1$ projection components are thrown away in the PCA step, some useful discriminatory information will be lost. The NLDA method supposes that the most discriminant information is contained in the null space of within-class scatter. In order to obtain the discriminant information in the null space of within-class scatter, NLDA firstly transforms the data into the null space of within-class scatter and then maximizes the between-class scatter. A

drawback of NLDA is that it solves the discriminant vectors only in the null space of the within-class scatter. Then the discriminant information in the range space of the within-class scatter will be lost.

LDA/GSVD, which is a LDA method based on generalized singular value decomposition, uses the Moore–Penrose generalized inverse of the total-scatter to replace the inverse of the within-class scatter defined in the LDA objective function. However, as pointed in [10], LDA/GSVD obtains the same projection subspace as NLDA when the rank of the total-class scatter is equal to the sum of the ranks of within-class and between-class scatter. Therefore, LDA/GSVD will suffer the same drawback of NLDA. The basic ideas of CLDA and DSLDA, respectively proposed by Yang et al. and Wang et al., are the same. They can obtain their discriminant vectors in the null space of within-class scatter and its complementary space, respectively.

There exists a common aspect of above algorithms, that is, they use singular value decomposition (SVD) or GSVD to compute the discriminant vectors. Compared with QR decomposition, however, SVD/GSVD is computationally expensive [11]. In [12], Ye et al. proposed a new LDA-based algorithm, called LDA/QR, which uses QR decomposition rather than SVD or GSVD to reduce the dimensionality of the training samples. LDA/QR can address the SSS problem, while achieving efficiency and scalability simultaneously.

The aforementioned LDA-based algorithms are all batch algorithms. That is, we must obtain all the training samples in advance. Then, these algorithms have to discard the discriminant vectors acquired in the past and repeat the training processing from the beginning when some new data items are inserted dynamically. To use these algorithms in on-line scenarios, it is crucial to update the projection matrix with light computation instead of full retraining when new data items are added

[☆] This paper has been recommended for acceptance by Seong-Whan Lee.

^{*} Corresponding author.

E-mail address: luguifu_jsj@163.com (G.-F. Lu).

dynamically. Incremental learning¹ (or online learning) can provide a way to address the above problem.

Recently, incremental learning has received a great attention in many practical applications, and some incremental LDA-based algorithms have been proposed. To learn the discriminant vectors of LDA incrementally, Pang et al. [13] proposed an incremental LDA (ILDA) method to update within-class scatter and between-class scatter. Then Pang's method can be used for classification of data streams. However, Pang's method is computationally expensive since no incremental method is introduced for the subsequent generalized eigenanalysis of the scatter matrices. By using fast SVD updating technique [14], Zhao et al. [15] proposed a new incremental LDA method, namely GSVD-ILDA, to incrementally update the discriminant vectors of LDA/GSVD when new data items are added. However, an approximate trick is used in GSVD-ILDA and there exists a performance gap between GSVD-ILDA and LDA/GSVD solutions. By borrowing the idea of incremental principal component analysis (IPCA) [16], Kim et al. [17,18] proposed another incremental LDA to update the discriminant vectors when new samples are inserted into the training data set. Similar to GSVD-ILDA, Kim's method is also an approximate technique. In [19], Uray et al. proposed an incremental LDA algorithm, called ILDAaPCA, which is a combination of two linear subspace algorithms, i.e., an incremental PCA and LDA. ILDAaPCA first performs IPCA and then updates the projection matrix of LDA on the updated subspace obtained by IPCA. Finally, ILDAaPCA projects the projection matrix of LDA onto another subspace of desired size while preserving the full discriminative information. Thus, ILDAaPCA can combine reconstructive and discriminative information of the training sample.

In [10], Zheng et al. proposed a modified dual-space LDA (MDSLDA) and its corresponding incremental extension. MDSLDA first projects all the training data onto the subspace spanned by the class means before the application of DSLDA. They also proposed an incremental MDSLDA which can accurately update the discriminant vectors of MDSLDA. Based on CLDA, Lu et al. [20,21] proposed two different implementations of incremental CLDA. Based on maximum margin criterion (MMC) [22,23], Yan et al. proposed an incremental implementation of MMC (IMMC) [24,25] which borrows the idea of the candid covariance-free incremental PCA (CCIPCA) [26].

To overcome the SSS problem and develop incremental LDA algorithm, in [27,28], Ye et al. proposed another incremental dimension reduction method via QR decomposition (IDR/QR). IDR/QR use QR decomposition of the class means matrix rather than SVD to reduce the dimension of the training samples. The projection vectors are contained in the range space of the class means matrix. By using efficient QR-updating techniques, IDR/QR can update the discriminant vectors with very low computation. Note that IDR/QR can also be performed in batch mode and the batch IDR/QR is the same as LDA/QR except that LDA/QR computes its discriminant vectors in the range space of between-class scatter where the global centroid is subtracted. Although some experiments on real database demonstrate the effectiveness of IDR/QR, it still has two limitations: 1) IDR/QR can only process new samples one instance after another even if a chunk of training samples is available at a time; 2) the approximate trick is used in IDR/QR. Then there exists a gap in performance between incremental and batch IDR/QR solutions.

To solve the limitations of IDR/QR, in this paper, we propose a new chunk IDR which is capable of processing multiple data instances at a time and can accurately update the discriminant

vectors when new data items are added dynamically. Firstly, we propose a new implementation of batch IDR/QR, which is equivalent to the original implementation of batch IDR/QR theoretically but whose computational cost is lower than the original ones. Then, based on our proposed new implementation of batch IDR/QR, we propose a new chunk IDR. Experiments on some real-world databases demonstrate the effectiveness of the proposed algorithm over the original one. Note that in [29], Pen has already proposed a chunk IDR/QR. However, Pen's method is also an approximate trick and the computational complexity of Pen's method is more expensive than Ye's incremental IDR/QR for single data processing.

The rest of this paper is organized as follows. In Section 2, we review briefly the LDA method and the IDR/QR method. In Section 3, we propose the new implementation of batch IDR/QR. The new chunk IDR is presented in Section 4. Section 5 is devoted to the experiments. Finally, we conclude the paper in Section 6.

2. Related works

2.1. Outline of LDA

Let $X = \{x_i^j, j = 1, 2, \dots, n_i; i = 1, 2, \dots, c\}$ be a d -dimensional sample set with n data points, where c is the number of the classes, n_i is the number of the samples of the i th class, x_i^j is the j th sample of the i th class and $\sum_{i=1}^c n_i = n$. In LDA, three matrices, called within-class, between-class and total scatter matrices, are defined as follows:

$$S_w = \sum_{i=1}^c \sum_{j=1}^{n_i} (x_i^j - m_i)(x_i^j - m_i)^T = H_w H_w^T \quad (1)$$

$$S_b = \sum_{i=1}^c n_i (m_i - m)(m_i - m)^T = H_b H_b^T \quad (2)$$

$$S_t = \sum_{i=1}^c \sum_{j=1}^{n_i} (x_i^j - m)(x_i^j - m)^T = H_t H_t^T = S_b + S_w \quad (3)$$

where the centroid m_i of the i th class is defined as $m_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_i^j$, the global centroid m is defined as $m = \frac{1}{n} \sum_{i=1}^c \sum_{j=1}^{n_i} x_i^j$ and H_b , H_w and H_t are defined, respectively, as

$$\begin{aligned} H_w &= [x_1^1 - m_1, \dots, x_1^{n_1} - m_1, \dots, x_c^1 - m_c, \dots, x_c^{n_c} - m_c] \in \mathbb{R}^{d \times n} \\ H_b &= [\sqrt{n_1}(m_1 - m), \dots, \sqrt{n_c}(m_c - m)] \in \mathbb{R}^{d \times c} \\ H_t &= [x_1^1 - m, \dots, x_1^{n_1} - m, \dots, x_c^1 - m, \dots, x_c^{n_c} - m] \in \mathbb{R}^{d \times n} \end{aligned} \quad (4)$$

The classical LDA algorithm aims to find an optimal projection matrix G that maximizes the so-called Fisher's criterion

$$G = \arg \max_G \text{trace} \left((G^T S_w G)^{-1} (G^T S_b G) \right) \quad (5)$$

Table 1

Computational complexity of Algorithms 1–2.

	Step 1	Step 2	Step 3	Step 4
Algorithm 1	$O(dn)$	$4dc^2 - 4c^3 / 3$	$2dnc$	$2dc^2$
Algorithm 2	$O(dn)$	$2dnc$	$2dc^2$	$2nc^2$
	Step 5	Step 6	Step 7	Step 8
Algorithm 1	$2nc^2$	$2c^3$	$O(c^3)$	$2dc^2$
Algorithm 2	$2c^3$	$O(dc)$	$O(c^3)$	$2dc^2$
Total computational cost				
Algorithm 1	$8dc^2 + 2c^3 / 3 + 2dnc + 2nc^2 + O(c^3) + O(dn)$			
Algorithm 2	$4dc^2 + 2c^3 + O(dc) + 2dnc + 2nc^2 + O(c^3) + O(dn)$			

¹ Incremental learning methods are designed to address real-world situations when the samples are frequently presented. There are two kinds of manner, i.e., sequential manner, where only one new sample is presented at a time, and chunk manner, where a subset of samples (at least two samples) is presented at a time.

Download English Version:

<https://daneshyari.com/en/article/526864>

Download Persian Version:

<https://daneshyari.com/article/526864>

[Daneshyari.com](https://daneshyari.com)