Neurocomputing 138 (2014) 310–315

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Letters

A subset method for improving Linear Discriminant Analysis

Chao Yao [a,*], Zhaoyang Lu [a], Jing Li [a], Yamei Xu [a], Jungong Han [b]

[a] State Key Laboratory of Integrated Service Networks, Xidian University, Xian 710071, China
[b] Civolution Technology, High Tech Campus 9, Eindhoven, The Netherlands

ARTICLE INFO

Article history:
Received 15 April 2013
Received in revised form
14 November 2013
Accepted 1 February 2014
Communicated by Marco Loog
Available online 14 February 2014

Keywords:
Linear Discriminant Analysis
Dimension reduction
Subset
Graph cut

ABSTRACT

Linear Discriminant Analysis (LDA) is one of the most popular methods for dimension reduction. However, it suffers from class separation problem for C-class when the reduced dimensionality is less than $C-1$. To cope with this problem, we propose a subset improving method in this paper. In the method, the subspaces are found for each subset rather than that for the entire data set. To partition the entire data set into subsets, a cost matrix is first estimated from the training set with the pre-learned classifier, then the graph cut method is adopted to minimize the cost between each subset. We use LDA to find subspaces for each subset. Experimental results based on different applications demonstrate both the generality and effectiveness of the proposed method.

1. Introduction

Extracting good features is crucial for pattern recognition tasks, especially when facing the high-dimensional data. Discriminative dimensionality-reduction methods are developed to resolve this problem. These methods not only help to alleviate the problem of dimension curse [1] but also help to improve the efficiency and accuracy of the classification algorithms. One of the most popular discriminative dimensionality reduction methods is the Linear Discriminant Analysis (LDA), which is proposed by Fisher [2] for solving binary class problems. It is further extended to multi-class cases by Rao [3]. In general, LDA aims to find a subspace that minimizes the within-class scatter and maximizes the between-class scatter simultaneously.

LDA has been successfully applied to many applications such as handwritten character recognition, face recognition, image retrieval and so on [4–7]. However, for a multi-class problem, it is proved to be not optimal. The main reason is that LDA tends to merge the classes located closely if the final dimensionality d is less than $C-1$, where C is the number of classes. Such an example can be found in Fig. 1, where LDA fails in a two-dimensional data set with four classes. In the last decade, researchers tried to tackle this problem in different ways. Lotlikar and Kothari [8] proposed a combination of a fractional step and a weighting function, where the weights of the pairs that will be potentially merged are increased. Loog et al. [9] introduced a different weighting function approximating a pairwise Bayes function. The function assigns a large value to the closely located pairs according to the Bayes rules. Recently, Bian and Tao [10] presented a method that aims to maximize the minimum pairwise distance. In this way, the distance between the classes that might be merged is maximized. However, these methods lessen the separation problem of LDA to some extent, there is still room for improvement.

It is well known that it is notoriously difficult to develop a Bayes optimal criterion for general multi-class dimension reduction. But in some special cases, progresses have been made. Geisser [11] gave the functional expression of Bayes error for C homoscedastic classes with equal priors Gaussian distribution. Schervish [12] solved it for two-dimensional data from three classes. Hamsici and Martinez [13] proved the projections having the same order of projected means in 1D subspace actually form a convex set. Hence, the problem could be solved by conditional convex optimization algorithms. To find 2D or higher dimensional subspaces, they developed an algorithm that selects a set of 1D subspaces by orthogonal projection. However, it should be noted that it might be necessary to solve $C!/2$ convex problems for each projection. With the growing class number C, the computational cost increases sharply.

From Fig. 1, it is obvious that if the whole set is first partitioned into two subsets, then LDA is applied on each subset individually, the problem of LDA will be greatly relieved. Therefore, we consider that LDA could be more efficient handling a series of smaller C'-class problems instead of the C-class problem, where $C' < C$. In this paper, we proposed a new method, namely Subset improving LDA (S-LDA), for addressing the separation problem of LDA for multi-class from a new perspective. Different from the aforementioned methods, S-LDA finds subspaces for each subset instead of the whole set. To get the best partition of the whole set, we first make use of the relation between each individual class for the

* Corresponding author.
E-mail addresses: yaochao@mail.xidian.edu.cn (C. Yao), zhylu@xidian.edu.cn (Z. Lu), jinglixd@mail.xidian.edu.cn (J. Li), ymxu@mail.xidian.edu.cn (Y. Xu), jungonghan77@gmail.com (J. Han).

http://dx.doi.org/10.1016/j.neucom.2014.02.004
0925-2312

specific classifier, then adopt the graph cut algorithm to solve this optimization problem. Actually, the subset improving method could also be applied to other dimension reduction methods to improve their performances. Experiments on synthetic data, two data sets from UCI machine learning repository and MNIST, a real handwriting digital character data set, demonstrate the effectiveness of the proposed method.

The rest of the paper is organized as follows. Section 2 reviews the details of LDA, and Section 3 presents the subset improving LDA, then experiments are shown in Section 4. Finally, we conclude the proposed method in Section 5.

2. Preliminary

For C-class problem, LDA aims to seek a set of optimal vectors, denoted by $W=[w_1, w_2, \ldots, w_l]$, such that the Fisher criterion

$$J(\mathbf{W}) = \mathrm{tr}\left(\frac{\mathbf{W}^T S_b \mathbf{W}}{\mathbf{W}^T S_w \mathbf{W}}\right) \tag{1}$$

is maximized. Where the within scatter matrix S_w and between scatter matrix S_b is defined below:

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{n_i} p_i (x_{ij} - m_i)(x_{ij} - m_i)^T \tag{2}$$

$$S_b = \sum_{i=1}^{C} p_i (m_i - m)(m_i - m)^T \tag{3}$$

where p_i is the prior probability of the ith class, m_i is the centroid of class i, x_{ij} is the jth sample of class i and n_i is the number of training samples from class i, m is the centroid of the global centroid. The object of LDA is to learn a transformation $W \in \mathbf{R}^{d \times d'}$ to minimize the within-class variance and as well as maximize the between-class variance. The solution to this problem is obtained by an eigenvalue decomposition of $S_w^{-1} S_b$ and take the eigenvectors corresponding to the d' largest eigenvalues.

3. Subset improving Linear Discriminant Analysis

Fig. 1 illustrates the separation problem of LDA. Researchers addressed on this problem from the criterion viewpoint in the last decade. However, we consider this problem from a different perspective. In this section, the details of the subset improving LDA method will be discussed.

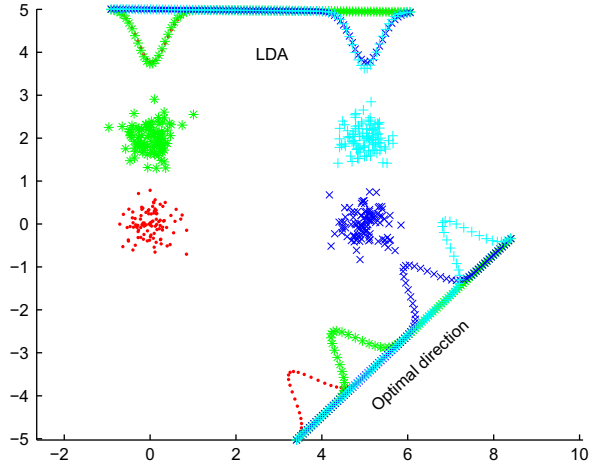

Fig. 1. An illustration in which LDA fails.

3.1. Partition method

The methods for dividing the samples into groups are known as clustering analysis, which have been well studied. However, it is not suitable to adopt those clustering methods in our scheme directly because we need the samples in the same class to be in the same group. Therefore, we propose a new partition method to meet the requirement.

We consider the partition of C-class problem in the following way. For the C-class problem $\{C_1, C_2, \ldots, C_c\}$, when it is divided into M subsets, it comes to $\{S_1, S_2, \ldots, S_M\}$, where $\{S_i\} = \{C_{S_{i1}}, C_{S_{i2}}, \ldots, C_{S_{in}}\}$. For a predefined cost between each pair of classes, the cost of a partition is

$$Cost = \sum_{i,j} cost(S_i, S_j) \tag{4}$$

where $cost(\cdot)$ is the cost between two subsets, and $cost(S_i, S_j) = \sum_{m \in S_i, n \in S_j} c_{mn}$, where c_{mn} is the predefined cost that separating two classes into different subset. Thus the best partition of the classes is equal to minimizing Eq. (4).

With the above definition, it is clear that the cost between each class would influent the partition result. Thus, the key problem of partition is how to define the cost between each pair of classes. We introduce a classifier-specific cost in the following way:

$$c_{ij} = \begin{cases} p_{i \to j} + p_{j \to i}, & i \neq j \\ 0, & i = j \end{cases} \tag{5}$$

where $p_{i \to j}$ is the probability of the ith class misclassified into the jth class with a pre-learned classifier, such as Nearest Mean, k-Nearest Neighbor and so on. In this way, the pair of classes easily misclassified will be in the same subset to make the final partition reliable.

To get better generalization performance, the cost should be estimated from a different data set from the training set of the predefined classifier. In our scheme, the training set is divided into two parts: one for training the classifier, the other for estimating the misclassification probabilities between each class. Cross-validation is used to make the probabilities more precise. Then the cost matrix is obtained:

$$E = \begin{bmatrix} 0 & c_{12} & \ldots & c_{1c} \\ c_{21} & 0 & \ldots & c_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ c_{c1} & c_{c2} & \ldots & 0 \end{bmatrix} \tag{6}$$

where the element c_{ij} is the cost that separating the ith class and jth class into different subsets. From Eq. (5), it is clear that $c_{ij} = c_{ji}$, which makes the cost matrix E symmetric.

3.2. Optimization method

To solve the optimization problem based on the above notifications, we employ graph cut methods which guarantee a globally optimal solution for a wide family of energy functions [14]. Let $G=(V, E)$ be an undirect graph with vertex set $V=\{v_1, v_2, \ldots, v_n\}$, the vertices v_i are the class indicator of class i and the weight w_{ij} between vertices v_i and v_j is the cost c_{ij} in our cost matrix E. Actually, in the cost matrix E, for $c_{ij} \geq 0$ and $c_{ij} = c_{ji}$, it meets the requirements of an undirect graph. That is the reason we can directly use graph cut methods to solve our problem.

Among the graph cut algorithms, Mini-Cut [15], Normalized Cut [16] and Ratio Cut [17] all aim to find a partition that minimizes the similarity between subsets according to weight matrix W, this is exactly what we need. However, Mini-Cut favors cutting small sets of isolated nodes in the graph, Ratio Cut performs slowly, we finally choose Normalized Cut to serve in our method. The partition method is summarized in the following Algorithm.