Neurocomputing 171 (2016) 664–672

Contents lists available at ScienceDirect

Neurocomputing

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

Locality sensitive batch feature extraction for high-dimensional data

Jie Ding [a], Changyun Wen [a], Guoqi Li [b,*], Chin Seng Chua [a]

[a] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore
[b] Center for Brain Inspired Computing Research, Department of Precision Instrument, Tsinghua University, Beijing, 100084 China

ARTICLE INFO

Article history:
Received 11 February 2015
Received in revised form
1 July 2015
Accepted 3 July 2015
Communicated by Zhaohong Deng
Available online 6 August 2015

Keywords:
Feature extraction
Dimensionality reduction
Classification
Local learning
Under sample problem

ABSTRACT

For feature extraction, the dimensionality of the feature space is usually much larger than the size of training set. This is known as under sample problem. At this time, local structure is more important than global structure. In this paper, locality sensitive batch feature extraction (LSBFE) is derived based on a new gradient optimization model by exploiting both local and global discriminant structure of data manifold. With the proposed LSBFE, a set of features can be extracted simultaneously. Recognition rate is improved compared with batch feature extraction (BFE), which only considers global information. It is shown that the proposed method achieves good performance for face databases, handwritten digit database, object database and DBWorld data set.

1. Introduction

Discriminative dimension reduction is a hot research topic in the field of pattern recognition since high-dimensional data in the feature space is not suitable for the subsequent work of classification due to dimension disaster. By extracting features in low dimension, it preserves well the discriminative information and then can separate different classes more accurately and efficiently. It has been widely applied in various areas, such as image and text retrieval [1,2], bioinformatics [3], biometrics [4], and signal processing [5].

For feature extraction, the dimensionality of the feature space is usually much larger than the size of training set. This is known as under sample problem (USP) [6,7]. A conventional classifier, for example, linear discriminant analysis (LDA) [8,9], often fails when faced by USP. One solution is to reduce the dimensionality of the feature space by using the principal components analysis (PCA) [10,11] or the multilinear subspace analysis (MSA) [12]. Unfortunately, some discriminant information is discarded by PCA and MSA. Some other algorithms such as discriminant common vectors [13], regularized LDA [14], general tensor discriminant analysis (GTDA) [15] based on differential scatter discriminant criterion (DSDC) [8] and Fukunaga–Koontz transform (FKT) [16] just consider global discriminant information without exploiting the underlying structure of data manifold.

Recently there has been a lot of interest in geometrically motivated approaches to data analysis in high dimensional spaces, including Laplacian Eigenmaps [17,18], locally linear embedding (LLE) [19], locality preserving projections (LPP) [20], local discriminant embedding (LDE) [21] and LDE-based algorithms [22]. These methods have been shown to be effective in discovering the geometrical structure of the underlying manifold. From this point of view, to reduce USP, we propose a new algorithm called locality sensitive batch feature extraction (LSBFE) by exploiting the geometry of data manifold, which can extract a set of features simultaneously. Firstly, we construct a nearest neighbor region for every sample to model the local geometrical structure of the underlying manifold and introduce local differential scatter discriminant criterion (LDSDC). Subsequently based on LDSDC, a constraint optimization problem with a new objective function is formulated. Then we transfer the optimization problem into an unconstraint optimization problem. However, solving such an unconstraint optimization problem is still a challenging task. To overcome this, an idea of by applying the gradient method on two unknown matrices respectively and iteratively is initiated. This enables us to propose an algorithm that ensures the objective function converges and an optimal projection matrix is obtained.

Laplacian Eigenmaps and LLE can be considered as the proper baselines which incorporate local structure of data manifold. They are closely related and the optimization problem as stated by LLE can be reformulated as trying to find the eigenvectors of graph Laplacian $\mathcal{L}^2$ [18]. The nonlinear property in these algorithms is more general compared with linear algorithms, yielding impressive results. Based on the characteristics of locality-preserving, Laplacian Eigenmaps is relatively robust to outliers and noise and exhibits stability with respect to the embedding. The optimization of LLE avoids local minimum problem and LLE can also be

* Corresponding author.
E-mail addresses: jding002@e.ntu.edu.sg (J. Ding), ecywen@ntu.edu.sg (C. Wen), liguoqi@mail.tsinghua.edu.cn (G. Li), ecschua@ntu.edu.sg (C.S. Chua).

http://dx.doi.org/10.1016/j.neucom.2015.07.076
0925-2312/

extended in classification [23]. However, experiments show that LLE is an effective method for visualization but may not seem to be very useful in classification [24]. For small numbers of features, LLE outperforms PCA. As the number of dimensions increases, it starts overfitting and cannot extract further information while PCA continues to improve the performance since less and less information is discarded [23,24]. Later by taking label information into account, supervised LLE (SLLE) is proposed [25,26]. It is worthwhile to highlight the differences of our proposed algorithm LSBFE from Laplacian Eigenmaps and LLE here. Firstly, by exploiting local information, Laplacian Eigenmaps and LLE are to construct local graph while LSBFE is to reconstruct scatter matrices and reduce under sample problem. Secondly, Laplacian Eigenmaps or LLE seeks to preserve the intrinsic geometry of the data and local structure after dimension reduction while LSBFE preserves the discriminant information by attempting to maximize between-class scatter and minimize within-class scatter for classification. Finally, LSBFE is a linear method for dimension reduction while Laplacian Eigenmaps and LLE are nonlinear algorithms.

The main advantage of our algorithm is that LSBFE can extract a set of features simultaneously and capture both local and global structures of the data sets, resulting in clear improvement of recognition rate over the results of batch feature extraction (BFE), which is based on differential scatter discriminant criterion (DSDC) and only takes account of global information. Thus LSBFE is more efficient and convenient than BFE for feature extraction. The main contributions of this paper are summarized as follows:

1. By constructing a nearest neighbor region for every training sample, class scatter matrices are redefined. Through this process, LDSDC is proposed and USP is reduced.
2. The problem is solved by exploring a new methodology from the point of view of gradient optimization. This enables us to propose a new algorithm in a different way from conventional approaches such as LDA and PCA based on eigenvalue decomposition of matrices.
3. An alternating optimization procedure is designed. It is illustrated that the procedure enables the objective function to converge, resulting in that the feature space spanned by a corresponding projection matrix is optimal.

The rest paper is organized as follows. In Section 2 we briefly introduce the model of LDSDC. The optimization procedure of the proposed algorithm is given in Section 3. The algorithm of LSBFE and related analyses are presented in Section 4. Experimental results on different data sets are reported and analyzed in Section 5. Paper is concluded in Section 6.

2. Problem formulation

LDA, as one of the prototypical method, has been widely applied to feature extraction and dimension reduction due to its effectiveness and simplicity. The aim of LDA is to find a projection matrix, which can separate different classes well in a low-dimensional subspace. The subspace is spanned by a set of vectors w_i, $1 \le i \le m$, which form the projection matrix W. Therefore, an optimal problem is formulated as determining a projection matrix so that the ratio between the trace of between-class matrix S_b and the trace of within-class matrix S_w [22] is maximized, namely

$$W^* = \underset{W}{\operatorname{argmax}} \frac{\operatorname{tr}(W^T S_b W)}{\operatorname{tr}(W^T S_w W)} = [w_1, w_2, \ldots, w_m] \tag{1}$$

For real problems, there generally exist similar examples in the same class, which can be regarded as subclasses or local structures. However, LDA only exploits the global discriminant information and fails to discover the underlying structure of data manifold. In order to capture local geometrical structures, we construct a nearest neighbor region for every training sample and redefine scatter matrices. In detail, given that a training set includes c classes. Let $x_{ij} \in R^d$ denote a training sample, where i is the class number with $1 \le i \le c$, j is the sample number in the ith class with $1 \le j \le n_i$ and R^d is a d-dimensional feature space. There are totally $n = \sum_{i=1}^{c} n_i$ samples in this training set. For a training sample x_{ij}, consider its k nearest neighbors in the same class j, denoted as $x_{ij}^1, x_{ij}^2, \ldots, x_{ij}^k$. Let $\overline{x}_{ij} = 1/(p_0 + p_1 + \cdots + p_k)(p_0 x_{ij} + p_1 x_{ij}^1 + p_2 x_{ij}^2 + \cdots + p_k x_{ij}^k)$ where $p_0, p_1, \ldots, p_k$ are weights of different neighbors. The weight value $p_l (0 \le l \le k)$ is defined as

$$p_l = \begin{cases} \exp(-\frac{\| x^k - x_{ij}^k \|}{t}), & 1 \le l \le k; \\ 1, & l = 0. \end{cases}$$

where $\| \cdot \|$ denotes the distance between training samples measured by Euclidean distance for low-dimensional data while Hamming distance for high-dimensional data[27,28]. Since Euclidean distance is particularly sensitive to noise, it may fail for high-dimensional data if lots of attributes are useless.

Then within-class matrix $\tilde{S_w}$ and the between-class matrix S_b are defined as

$$\tilde{S_w} = \frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{n_i} (\overline{x}_{ij} - \overline{m}_i)(\overline{x}_{ij} - \overline{m}_i)^T$$

$$S_b = \frac{1}{n} \sum_{i=1}^{c} n_i (\overline{m}_i - \overline{m})(\overline{m}_i - \overline{m})^T \tag{2}$$

where $\overline{m}_i = (1/n_i) \sum_{j=1}^{n_i} x_{ij}$ is the mean vector of the ith class, and $\overline{m} = (1/n) \sum_{i=1}^{c} \sum_{j=1}^{n_i} x_{ij}$ is the total mean vector for all classes.

Based on LDA and (2), we propose a local differential scatter discriminant criterion (LDSDC), which is defined as

$$W^* = \underset{W^T W = I}{\operatorname{argmax}} \left(\operatorname{tr}(W^T S_b W) - \mu \operatorname{tr}(W^T \tilde{S_w} W) \right) \tag{3}$$

where μ is a tuning parameter, W is the projection matrix, constrained by $W^T W = I$.

In this paper, LSBFE is proposed to find the optimal projection matrix W based on LDSDC. On the other hand, BFE is based on DSDC. Note that when any local information is not considered, no nearest neighbor regions will be constructed for training samples. For this special case, $\overline{x}_{ij} = x_{ij}$ in (2). Then LDSDC is actually DSDC [8] and the proposed algorithm LSBFE becomes BFE in this case.

3. Optimization procedure of LSBFE

In this section, we first formulate our problem and then transform it into two sub-models. Finally we propose a new iterative algorithm to obtain an optimal projection matrix.

3.1. Model transformation

Definition 1. Let A and B be two matrices of the same dimension $m \times n$. The Hadamard product $A \circ B$ is defined as $(A \circ B)_{ij} = A_{ij} B_{ij}$. More explicitly,

$$A \circ B = \begin{pmatrix} A_{11}B_{11} & \ldots & A_{1n}B_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1}B_{m1} & \cdots & A_{mn}B_{mn} \end{pmatrix}$$

Now assume that we are going to extract m vectors in problem (3). Then (3) is actually equivalent to the following optimization