Neurocomputing 73 (2010) 968–974

Contents lists available at ScienceDirect

Neurocomputing

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

Outlier-resisting graph embedding

Yanwei Pang [a], Yuan Yuan [b,*]

[a] School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China
[b] School of Engineering and Applied Science, Aston University, Birmingham B4 7ET, UK

ARTICLE INFO

Article history:
Received 14 March 2009
Received in revised form
28 July 2009
Accepted 18 August 2009
Communicated by X. Li
Available online 9 October 2009

Keywords:
Graph embedding
Locality preserving projection
L1-norm
Outlier
Dimensionality reduction

ABSTRACT

Graph embedding is a general framework for subspace learning. However, because of the well-known outlier-sensitiveness disadvantage of the L2-norm, conventional graph embedding is not robust to outliers which occur in many practical applications. In this paper, an improved graph embedding algorithm (termed LPP-L1) is proposed by replacing L2-norm with L1-norm. In addition to its robustness property, LPP-L1 avoids *small sample* size problem. Experimental results on both synthetic and real-world data demonstrate these advantages.

1. Introduction

Locality, known as neighborhood preserving property, is of fundamental importance in both nonlinear and linear dimensionality reduction methods. The underlying idea behind the locality/neighborhood preserving algorithms is that close points in high dimensional space remain close and similarly co-located w.r.t. to one another in the low dimensional space. Such an idea of *think globally and fit locally* has been applied in many computer vision applications [21,13,5].

Among much previous efforts, representative nonlinear dimensionality reduction algorithms, which have locality preserving property, are *locally linear embedding* (LLE) [20], Laplacian Eigenmap [1], Isomap [25], etc. While, some linear versions of Laplacian Eigenmap are *locality preserving projection* (LPP) [7], Laplacianfaces [8], *marginal Fisher analysis* (MFA) [26], *maximum margin projection* (MMP) [9]. Linear versions of LLE include *neighborhood preserving projection* (NPP) [19], *neighborhood preserving embedding* (NPE) [6], *nonlinear data projection on non-Euclidean manifolds* [14], etc. Compared to the *think globally and fit locally* algorithms, traditional *principal component analysis* (PCA) [28] and variants of PCA [16,22,17,18] and multidimensional scaling (MDS) are *fit globally* subspace algorithms. PCA and MDS can successfully discover low dimensional manifold on the premise of Gaussian data. LLE, Laplacian Eigenmap, and Isomap can work for non-Gaussian and nonlinear data structure and their linear versions are usually superior to PCA and MDS. All of these locality preserving algorithms are easy to implement and are not prone to local minima. Recent research showed that the above algorithms are instances of a general dimensionality reduction framework: *Graph Embedding* [26].

In this paper, we focus on improving LPP, one of the most important locality preserving based graph embedding. The idea of the enhanced LPP can be easily generalized to other graph embedding algorithms. LPP has been successfully used in image retrieval, face recognition, and scientific visualization. However, LPP as well as Laplacian Eigenmap and LLE are sensitive to outliers which corrupt the training data [2]. LPP as well as Laplacian Eigenmap can be treated as a graph embedding algorithm [26]. The objective of LPP is to minimize the weighted sum of squared distances between any two points. Though the edge weights have positive contribution to resisting outliers, the employment of L2-norm distance weakens its robustness to outliers. Because of the L2-norm distance, large distance dominates the sum. So the resulting subspace is biased.

To improve the robustness of LPP against outliers, we propose to employ L1-norm based distance to measure the dissimilarity between pairs of points. It has been shown that with the presence of outliers the L1-norm performs better than L2-norm [4]. We named the robustified LPP LPP-L1. The optimization process of LPP-L1 is similar to that of PCA-L1 [10]. But LPP-L1 has many advantages over PCA-L1. PCA-L1 is a L1-norm based PCA which removes the absolute value operator by introducing a polarity function and updates the basis vector by weighted sum of residuals of the training data. The merits of LPP-L1 are as follows:

(1) LPP-L1 is more robust than LPP against outliers;

[*] Corresponding author.
E-mail addresses: pyw@tju.edu.cn (Y. Pang), yuany1@aston.ac.uk (Y. Yuan).

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.08.020

(2) for high-dimensional data, LPP always encounters small sample size problem which makes the generalized eigendecomposition problem of LPP unstable. LPP-L1 needs not to deal with eigenvalue computation and so can avoids the small sample size problem; and
(3) compared to PCA-L1, for nonlinear manifold, LPP-L1 can remain the nonlinear structure in low-dimensional subspace.

In this paper, vectors are denoted as lower case bold roman letters such as $\mathbf{u}$ and $\mathbf{u}_i$, and all vectors are assumed to be column vectors. The subscript i of $\mathbf{u}_i$ indexes the number of the vector. But the subscript i is omitted when the index number is not explicitly specified. In that case, $\mathbf{u}$ is used rather than $\mathbf{u}_i$. Uppercase bold roman letters, such as $\mathbf{U}$, denote matrices.

The rest of this paper is organized as follows: Section 2 gives an introduction to LPP and PCA-L1. We present the proposed method, LPP-L1, in Section 3. Section 4 gives experimental results. Section 5 concludes the paper.

2. LPP and PCA-L1

The proposed LPP-L1 is a robust version of LPP and inspired by PCA-L1. Both LPP and PCA-L1 are described briefly in this section.

2.1. LPP

LPP is a famous linear subspace learning algorithm [7]. In essence, LPP is a linear approximation of the nonlinear Laplacian Eigenmap [1]. Laplacian Eigenmap can only map known training data while LPP by using linear transformation matrix can easily map any new data points.

Given a training matrix $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_N]\in\mathbb{R}^{n\times N}$ with each training point $\mathbf{x}_i\in\mathbb{R}^{n\times 1}$. As other subspace learning algorithms, LPP uses the obtained transformation matrix $\mathbf{U}=[\mathbf{u}_1,\ldots,\mathbf{u}_r]\in\mathbb{R}^{n\times r}$ with the basis vector $\mathbf{u}_i\in\mathbb{R}^{n\times 1}$ to map the high-dimensional points $\mathbf{x}\in\mathbb{R}^{n\times 1}$ to low-dimensional point $\mathbf{y}\in\mathbb{R}^{r\times 1}$:

$$\mathbf{y}=\mathbf{U}^T\mathbf{x}. \tag{1}$$

LPP differs from other methods in its criterion for computing the optimal basis vector $\mathbf{u}\in\mathbb{R}^{n\times 1}$:

$$\mathbf{u}=\arg\min_{\mathbf{u}} f(\mathbf{u})=\arg\min_{\mathbf{u}}\sum_{i,j}^{N}(\mathbf{u}^T\mathbf{x}_i-\mathbf{u}^T\mathbf{x}_j)^2 s_{ij}, \tag{2}$$

$$\text{s.t.}\mathbf{u}^T\mathbf{X}\mathbf{D}(\mathbf{u}^T\mathbf{X})^T=1,$$

where s_{ij} measures the similarity of $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathbf{D}$ is a diagonal matrix with its element $D_{ii}=\sum_j s_{ij}$. A frequently used similarity is heat kernel [1,7]:

$$s_{ij}=\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|_2^2}{t}\right), \tag{3}$$

where t is a pre-defined parameter. In (3) the similarity s_{ij} monotonously increases with the decrease of the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. Hence, s_{ij} incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart [7]. The net effect of minimizing the objective function is locality preserving, i.e. if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close then $\mathbf{u}^T\mathbf{x}_i$ and $\mathbf{u}^T\mathbf{x}_j$ are close as well [7]. As stated in [26], for larger similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$, the distance between $\mathbf{u}^T\mathbf{x}_i$ and $\mathbf{u}^T\mathbf{x}_j$ should be smaller to minimize the objective function. Likewise, smaller similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ should lead to larger distances between $\mathbf{u}^T\mathbf{x}_i$ and $\mathbf{u}^T\mathbf{x}_j$ for minimization [26].

It is worth noting that the value of s_{ij} is nonzero only if $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighbors. Therefore LPP adopts the "thinking globally but fit locally" strategy [21] similar to LLE.

Let s_{ij} constitutes a weight matrix $\mathbf{S}$. The Laplacian matrix $\mathbf{L}$ is then formed by subtracting $\mathbf{S}$ from $\mathbf{D}$:

$$\mathbf{L}=\mathbf{D}-\mathbf{S}. \tag{4}$$

The optimization problem of (2) can be reduced to a generalized eigendecomposition problem [7]:

$$\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{u}=\lambda\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{u}. \tag{5}$$

When $\mathbf{X}\mathbf{D}\mathbf{X}^T$ in (5) is singular, original LPP is unstable. That it is caused by the so called small sample size problem.

2.2. PCA-L1

LPP-L1 is a robustified LPP. But the optimization process is similar to that of PCA-L1. The objective of PCA-L1 is to maximize the L1-norm variance $f(\mathbf{u})$ in feature space:

$$f(\mathbf{u})=\|\mathbf{u}^T\mathbf{X}\|_1=\sum_{i=1}^{N}|\mathbf{u}^T\mathbf{x}_i|, \tag{6}$$

subject to

$$\|\mathbf{u}\|_2=1, \tag{7}$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ denote L1 norm and L2 norm, respectively.

One of the novelties of PCA-L1 is to convert sum of the absolute values in (6) into an ordinary sum. This is done by introducing a polarity function p_i:

$$p_i=\begin{cases}1 & \text{if } \mathbf{u}^T\mathbf{x}_i\geq 0\\ -1 & \text{if } \mathbf{u}^T\mathbf{x}_i<0\end{cases}. \tag{8}$$

Armed with this polarity function, Eq. (6) can be written as

$$f(\mathbf{u})=\sum_{i=1}^{N}|\mathbf{u}^T\mathbf{x}_i|=\sum_{i=1}^{N}p_i\mathbf{u}^T\mathbf{x}_i. \tag{9}$$

The maximization process is performed in an iterative manner:

$$\mathbf{u}(t)=\frac{\sum_{i=1}^{N}p_i(t-1)}{\|\sum_{i=1}^{N}p_i(t-1)\mathbf{x}_i\|_2}\mathbf{x}_i. \tag{10}$$

where the polarity function at time $t=1$ is chosen so that

$$p_i(t-1)=\begin{cases}1 & \mathbf{u}(t-1)^T\mathbf{x}_i\geq 0\\ -1 & \mathbf{u}(t-1)^T\mathbf{x}_i<0\end{cases}. \tag{11}$$

The solution of PCA-L1 is guaranteed to reach a local maximum. Refer to [10] for the proof. It has been shown that PCA-L1 converges faster than R1-PCA [4]. As can be seen from (10), the computation of PCA-L1 is as simple as weighted sum of the training samples and the weight is either 1 or -1.

3. L1-norm based LPP

3.1. Objective Function of LPP-L1

Despite its advantages, LPP is sensitive to outliers in some degree. Each item $(\mathbf{u}^T\mathbf{x}_i-\mathbf{u}^T\mathbf{x}_j)^2$ of the sum in (2) is the squared distance in embedded subspace. If there are outliers, the square will make the optimization process to place much emphasis on this squared distance between two outliers or between one regular point and an outlying point to some extent. Though the similarity s_{ij}, which monotonously decreases with the increase of the squared distance, is helpful for suppressing the negative effect of outliers, the quadratic distance weaken the robustness. As a result, the solution is biased to fit outliers.