Image and Vision Computing 28 (2010) 872–880

Contents lists available at ScienceDirect

Image and Vision Computing

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

Distinguishing variance embedding

Wang Qinggang [a,*], Li Jianwei [b], Wang Xuchu [b]

[a] Chongqing University, Chongqing 400044, China
[b] Key Laboratory of Optoelectronic Technology and Systems of Ministry of Education, Chongqing University, Chongqing 400044, China

ARTICLE INFO

Article history:
Received 22 March 2008
Received in revised form 23 October 2009
Accepted 9 November 2009

Keywords:
Manifold learning
Dimensionality reduction
Maximum variance unfolding
Laplacian eigenmaps
Variance analysis

ABSTRACT

Nonlinear dimensionality reduction is a challenging problem encountered in a variety of high dimensional data analysis. Based on the different geometric intuitions of manifolds, maximum variance unfolding (MVU) and Laplacian eigenmaps are designed for detecting the different aspects of data set. In this paper, combining the ideas of MVU and Laplacian eigenmaps, we propose a new nonlinear dimensionality reduction method called distinguishing variance embedding (DVE), which unfolds the data manifold by maximizing the global variance subject to the proximity relation preservation constraint originated in Laplacian eigenmaps. We illustrate the algorithm on easily visualized examples of curves and surfaces, as well as on actual images of faces, handwritten digits, and rotating objects.

1. Introduction

Dimensionality reduction [4] is the transformation of high dimensional data into meaningful representation of reduced dimensionality. It is important in many domains, since it facilitates image compression [26], computer vision [9,14], pattern recognition [22], and computational neuroscience [19] by mitigating the curse of dimensionality and other undesired properties of high dimensional space. Principal component analysis (PCA) [7,8,12] and metric multidimensional scaling (MDS) [5] are two classical linear dimensionality reduction methods. They can generate faithful low dimensional representations when the high dimensional input patterns are mainly confined to a low dimensional linear subspace. However, they do not generally succeed in the case that the input patterns lie on or near a low dimensional manifold embedded in a high dimensional space [18]. A common example of a nonlinear manifold embedded in a high dimensional space is image vectors of a face observed under different poses and lighting conditions [21]. In such a case, the dimensionality is restricted by the degrees of freedom of the physical constraints under which the images were taken. Whereas, the data has a much greater dimensionality depending on the resolution of the image.

Recently, several manifold learning algorithms have been proposed for dealing with the high dimensional data that has been sampled from a low dimensional manifold, such as Isomap [21], locally linear embedding (LLE) [16,17], Laplacian eigenmaps [1,2], local tangent space analysis (LTSA) [28], and maximum variance unfolding (MVU) [20,23,25]. Based on the different geometric intuitions [4,18], these methods can reveal the low dimensional structure of the submanifold (and sometimes even the dimensionality itself) that cannot be detected by classical linear methods.

Laplacian eigenmaps is based on the graph Laplacian that can be viewed as a discrete approximation to the Laplace–Beltrami operator on continuous manifolds. Laplacian eigenmaps is a local algorithm for nonlinear dimensionality reduction, that the nearby points in the original space are mapped nearby and the far points are not explicitly considered. Hence the algorithm imposes a natural clusters of the data. However not all the data sets necessarily have meaningful clusters. Moreover, in the presence of noise around the manifold, the local properties of manifold do not necessarily follow the global structure of the manifold [3,15]. In these case the global algorithms such as MVU and Isomap might be more appropriate.

MVU is based fundamentally on the notion of isometry, a smooth invertible mapping that looks locally like a rotation plus translation. It attempts to "unfold" a data manifold by pulling the input patterns as far apart as possible subject to the constraints that distances and angles between neighboring points are strictly preserved, i.e. local isometry. MVU is a global algorithm for nonlinear dimensionality reduction, in which all the data pairs, nearby and far, are considered. The final optimization of MVU is reformulated as an instance of semidefinite programming (SDP). Large-scale application is a particular challenge to MVU due to the expense of solving SDP.

Combining the ideas of MVU and Laplacian eigenmaps, in this paper, we present a new algorithm for nonlinear dimensionality reduction, called distinguishing variance embedding (DVE). Similar

* Corresponding author. Tel.: +86 2365102516.
E-mail address: ygest@hotmail.com (W. Qinggang).

0262-8856/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2009.11.003

to MVU and Isomap, DVE attempts to detect the global structure of nonlinear manifolds. However, our algorithm is simple to implement, and its main optimization involves a eigenvalue problem. Compared with MVU and Isomap, the computational complexity is dramatically reduced. The algorithm can be viewed as a variance of MVU that relaxes the strict distance-preserving constraints. This relaxation makes DVE relatively insensitive to noise.

The organization of this paper is as follows. In Section 2, we briefly review MVU and Laplacian eignmaps. In Section 3, we first show how to derive the optimization of DVE from the geometric intuitions of MVU and Laplacian eignmaps; then formally present the algorithmic procedure and discuss the problem of parameters selection; the brief comparison between DVE and MVU is also provided in this section. In Section 4, we present experimental results on several data sets, including easily visualized examples of curves and surfaces, as well as images of faces, handwritten digits, and rotating objects. Finally, we provide some concluding remarks and suggestions for future work in Section 5.

2. Laplacian eignmaps and MVU

Assume that high dimensional data set has been sampled from a low dimensional manifold $\mathscr{M}$. Algorithms for manifold learning map high dimensional inputs $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^T$ to low dimensional outputs $Y = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)^T$ that provide faithful representations of high dimensional inputs, where $\mathbf{x}_i \in \mathbb{R}^d, \mathbf{y}_i \in \mathbb{R}^r$, and $r \ll d$.

Laplacian eignmaps and MVU both can be viewed as the graph-based methods, so they share some common properties: they both construct a weighted neighborhood graph G whose n nodes represent input patterns and edges indicate neighborhood relations; the low dimensional embeddings are derived from the bottom or top eigenvectors of the matrix computed from the neighborhood graph G. Our algorithms builds on the two methods, so we begin by reviewing them.

2.1. Laplacian eigenmaps

Laplacian eigenmaps is a local structure preserving algorithm which is based on a simple geometric intuition: nearby inputs in the high dimensional space should be mapped to nearby in the reduced space. It first constructs a neighborhood graph G. Typically, the weights of the edges in the graph are computed by using the Gaussian kernel function, leading to a sparse matrix W with entries

$$W_{ij} = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) \tag{1}$$

where σ^2 is a scale parameter. Let D denotes the diagonal matrix with elements $D_{ii} = \sum_j W_{ij}$, which provides a natural measure on the vertices of the graph. The outputs $\mathbf{y}_i$ can be chosen to minimize the cost function:

$$\begin{aligned} \text{minimize} \quad & \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{n} \|\mathbf{y}_i\|^2 D_{ii} = 1 \end{aligned} \tag{2}$$

the constraint removes an arbitrary scaling factor in the embedding.

In the cost function (2), large weights W_{ij} correspond to the small distances between the data pairs $\mathbf{x}_i$ and $\mathbf{x}_j$. Hence, the difference between their low dimensional representations $\mathbf{y}_i$ and $\mathbf{y}_j$ highly contributes to the cost function. As a consequence, nearby points in the high dimensional space are brought closer together in the low dimensional representation, so the neighborhood relations are correctly preserved by Laplacian eigenmaps.

2.2. Maximum variance unfolding

MVU is the nonlinear counterpart of PCA [18,24]. The algorithm attempts to find the low dimensional embeddings that have the maximum total variance, while preserving the distances between neighboring points. Like Laplacian eigenmaps, the first step of the algorithm is to construct a neighborhood graph G by connecting each input $\mathbf{x}_i$ with its k-nearest neighbors. Let $W_{ij} \in \{0, 1\}$ indicate whether there is an edge between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the graph G. The outputs $\mathbf{y}_i$ from MVU are those that solve the following optimization:

$$\begin{aligned} \text{maximize} \quad & \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \\ \text{subject to} \quad & \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2 W_{ij} \\ & \sum_i \mathbf{y}_i = 0 \end{aligned} \tag{3}$$

Here, the first constraint enforces that distances between nearby inputs match distances between nearby outputs. The second constraint removes a translational degree of freedom from the final solution.

The optimization as stated above is not convex, but it can be reformulated as a semidefinite programming (SDP) by defining a inner product matrix K of the low dimensional data representations, i.e. $K_{ij} = \mathbf{y}_i \cdot \mathbf{y}_j$. Let $\|\mathbf{x}_i - \mathbf{x}_j\|^2 = d_{ij}$, we can write the SDP problem as follows:

$$\begin{aligned} \text{maximize} \quad & \mathrm{Tr}(K) \\ \text{subject to} \quad & K_{ii} - 2K_{ij} + K_{jj} = d_{ij} \quad \text{for } \forall (i,j) \in G \\ & \sum_{ij} K_{ij} = 0 \\ & K \succcurlyeq 0 \end{aligned} \tag{4}$$

From the solution K of the SDP, the low dimensional data representation Y can be obtained by performing a singular value decomposition, i.e. $Y = K^{\frac{1}{2}}$.

3. Distinguishing variance embedding

3.1. The optimization

Here we first reformulate Eq. (3) of MVU by introducing the complementary graph G' of the neighborhood graph G. In graph G', an edge with weight $W'_{ij} = 1$ is added between nodes i and j if they are not connected in graph G, otherwise not. Noting that the graph sum $G + G'$ is a complete graph, so Eq. (3) can be rewritten as

$$\max \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} + \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W'_{ij} \tag{5}$$

For the fixed data set $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, the neighborhood relations are invariable with the certain nearest neighbor criteria. So the total sum of neighboring points $\sum_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|^2 W_{ij}$ is equal to a constant c. Due to local distance-preserving constraints $\|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2 W_{ij}$, we have $\sum_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} = c$. The equivalent characterization of the Eq. (3) is

$$\max \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W'_{ij} \tag{6}$$

For the convenience of comparison, we rewrite the objective function (2) of Laplacian eignmaps

$$\min \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{ij} \tag{7}$$

Interestingly, based on different geometric intuitions, completely different variance optimization strategies are adopted in MVU and Laplacian eignmaps. In MVU, to unfold the data set, the