ARTICLE IN PRESS

Neurocomputing (xxxx) xxxx–xxxx

Contents lists available at ScienceDirect

Neurocomputing

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

Brief papers

Semi-supervised manifold alignment with few correspondences

Jing Wang[*,1], Xin Zhang, Xueqing Li, Jixiang Du

The School of Computer Science and Technology, Huaqiao University, Xiamen City 361021, PR China

ARTICLE INFO

Communicated by Deng Cai

Keywords:
Manifold alignment
Geodesic distance
Correspondence information
Semi-supervised

ABSTRACT

In recent years, manifold alignment methods have aroused a great of interest in the machine learning community which construct a common latent space shared by multiple input data sets. In a semi-supervised problem, it is assumed that some predetermined correspondences are available to us. The effectiveness of the semi-supervised manifold alignment methods may be very limited with very limited prior information. In this paper, we propose a novel semi-supervised manifold alignment algorithm with few given pairwise correspondences. Our approach characterizes the manifold structure of each sample point using the geodesic distances between the sample point and given correspondences. Then we build the connections between the points sampled different manifolds using the characterized manifold structure. The points of multiple data sets are finally projected to a common space simultaneously preserving the local geometry of each manifold and the captured connections between manifolds. We demonstrate the effectiveness of our method in a series of carefully designed experiments.

1. Introduction

In many real-world applications, observations represented as high-dimensional data can be modeled as sample points lying on low-dimensional nonlinear manifolds. There have been advances in developing effective and efficient algorithms for learning meaningful low-dimensional manifold from the high-dimensional data. These algorithms include isometric mapping (Isomap) [20], locally linear embedding (LLE) [17], Laplacian eigenmaps (LE) [2], and local tangent space alignment (LTSA) [30], etc. Due to simple geometric intuitions, straightforward implementation, and global optimization, these algorithms have been successfully applied in many research fields such as data mining, machine learning, image analysis, and computer vision.

Although these manifold learning algorithms have been successfully applied for real-world data analysis, they are designed to discover the low-dimensional features of the sample points lying on a single manifold. In many real-world applications like matching words and pictures [15,18], cross-lingual information retrieval [4–6,21], image interpretation [1,28], it is required to discover the latent features of two or more disparate input data sets. Since these input data sets lie on different manifolds, the general manifold learning algorithms are not applicable.

More recently, manifold alignment algorithms are proposed to construct a common latent space shared by multiple input data sets [9,23]. The framework of the most manifold alignment algorithms consists of the following steps:

(1) Discover manifold structure of each data set. The discovered manifold structure can be local or global geometry. Most of the existing manifold alignment algorithms are designed to preserve local geometries of the input manifolds [9,22,23]. In some real-world applications such as cross-lingual information retrieval, the global manifold geometry need to be respected [26]. The global geometry preserving algorithms can be more effective in these applications.
(2) Build connections between the input data sets. To align different input manifolds, it is required to build the connections between the points sampled from the input manifolds. Many existing approaches build the connections using the manifold structures [7,16,24] or the given correspondence information [8,9,19,28].
(3) Map all input data sets into a common low-dimensional embedding space. Manifold alignment approaches try to map all the points to the same space preserving the discovered manifold structure and connections. In general, the manifold alignment approaches can be done at instance-level or feature-level. Instance-level alignment [22] computes the embedding coordinates of the data points. Feature-level alignment [23,26] computes the mapping functions that map the input data sets to the latent space.

* Corresponding author.
E-mail address: wroaring@hqu.edu.cn (J. Wang).

[1] The work of this author was supported in part by NSFC (61370006, 61673186), NSF of Fujian Province 2014J01237, and Promotion Program for Young and Middle-aged Teacher in Science and Technology Research of Huaqiao University ZQN-PY116.

http://dx.doi.org/10.1016/j.neucom.2016.12.010
Received 27 June 2016; Received in revised form 26 November 2016; Accepted 4 December 2016
0925-2312/ © 2016 Elsevier B.V. All rights reserved.

Please cite this article as: Wang, J., Neurocomputing (2016), http://dx.doi.org/10.1016/j.neucom.2016.12.010

Since the mapping functions can be easily generalized to new test points, feature-level alignment outperforms instance-level alignment on handling the new test points.

A key issue that determines the effectiveness of the manifold alignment approaches is how to accurately build connections between the input data sets. Since the points from different data sets are represented by different features, it is difficult to compare them directly. There are some efforts on solving this problem. One line is to build the connections using their local on-manifold structures such as the distance matrix of [24] and parameterized curve representation [16]. Using the manifold structures rather than the original features to represent the data points makes it possible to compare the points from different data sets directly. However, the true match may be difficult to identify since the manifolds may have multiple similar manifold structures. The effectiveness of these approaches may be very limited in this situation. The other line is to build the connections using the given correspondence information such as pairwise correspondences [9], relative comparison information [27] or label information [25]. These approaches can have good performance when sufficient prior information is available. Consider the fact that the data sets from real-world often have very limited prior information, the applications of these approaches are also greatly limited.

The purpose of this paper is to address the key issue. Assume that very limited correspondence information such as few pairwise correspondences is given. For each point x_i, we use the geodesic distances along the manifold between x_i and the correspondences to characterize x_i's manifold structure. Using global geometry rather than local geometry to characterize the manifold structure can avoid the false positive matchings. To build the connections between the points from different data sets, the similarities between the characterized manifold structure are computed. The data points from each data set are then mapped to a common space at instance-level or feature-level by solving a constrained embedding function, simultaneously preserving the manifold topology and discovered connections.

The rest of this paper is organized as follows. In Section 2, we propose a novel semi-supervised manifold alignment algorithm when only few pairwise correspondences are given. We also give a theory analysis of the proposed algorithm in Section 3. After that, the related work will be introduced in Section 4. We will give numerical experiments in Section 5 to show the effectiveness of the proposed manifold alignment algorithm. Some conclusion remarks are given in Section 6.

2. The Main algorithm

Given two data sets $X=[x_1,\ldots,x_n]$ and $Y=[y_1,\ldots,y_m]$ with $x_i\in R^{p_x}$, $y_j\in R^{p_y}$, sampled from two d-dimensional manifolds $\mathcal{X}$ and $\mathcal{Y}$. For the semi-supervised manifold alignment problem, it is assumed that the pairwise correspondences of some of the sample points are given. Without loss of generality, suppose the first l points in X and Y are the pairwise correspondences, i.e., $x_i\leftrightarrow y_i,\ \ i=1,\ldots,l$. The purpose of our proposed algorithm is to learn the embedding coordinates $s=[s_1,\ldots,s_n]\in R^{d\times n}$ of X and $t=[t_1,\ldots,t_m]\in R^{d\times m}$ of Y, or the linear projection function $f\in R^{p_x\times d}$, $g\in R^{p_y\times d}$ such that the embedding results f^TX and g^TY preserve local geometries of the input manifolds and the connections between manifolds.

2.1. The optimization model

The graph Laplacian is widely used in dimensionality reduction due to its nice properties. The graph Laplacian based methods such as Laplacian eigenmaps [2] construct a weighted graph to capture the local geometry in the manifold. For the points $x_1,\ldots,x_n$ sampled from the manifold $\mathcal{X}$, Laplacian eigenmaps construct an $n\times n$ weight matrix $W^{\mathcal{X}}$ to summary the similarity between points, i.e., $W_{i,j}^{\mathcal{X}}=e^{-\frac{\|x_i-x_j\|^2}{t}}$ with t being the heat kernel parameter or $W_{i,j}^{\mathcal{X}}=1$ for simplicity when x_i and x_j are neighbors, otherwise $W_{i,j}^{\mathcal{X}}=0$. Then Laplacian eigenmaps construct $s=[s_1,\ldots,s_n]$ to minimize the function

$$\frac{1}{2}\sum_{i,j}W_{i,j}^{\mathcal{X}}\parallel s_i-s_j\parallel^2=\mathrm{tr}(sL^{\mathcal{X}}s^T),$$

where $\mathrm{tr}(\cdot)$ denotes the trace of the matrix, $L^{\mathcal{X}}=D^{\mathcal{X}}-W^{\mathcal{X}}$ is the graph Laplacian matrix, and $D^{\mathcal{X}}$ is a diagonal matrix with $D_{i,i}=\sum_j W_{i,j}^{\mathcal{X}}$. Similarly, we can construct the graph Laplacian matrix $L^{\mathcal{Y}}$ for the data sampled from the manifold $\mathcal{Y}$. In our model, we use the graph Laplacian to capture the local geometries of the input manifolds.

In order to preserve the local geometries of each manifold and the connections between x_i and y_j, we want to minimize the cost function

$$C(s,t)=\frac{1}{2}\sum_{i,j}W_{i,j}^{\mathcal{X}}\parallel s_i-s_j\parallel^2+\frac{1}{2}\sum_{i,j}W_{i,j}^{\mathcal{Y}}\parallel t_i-t_j\parallel^2+\lambda\sum_{i,j}W_{i,j}\parallel s_i-t_j\parallel^2,\tag{1}$$

where $W_{i,j}$ represent the similarity between x_i and y_j. We will propose a novel approach to determine $W_{i,j}$ in the next subsection. The first two terms guarantee that the learned local geometries can be preserved in the embedding space, while the third term penalizes the difference between the embedding coordinates of the data sets X and Y. λ is a tradeoff parameter. Notice that $\sum_{i,j}W_{i,j}\parallel s_i-t_j\parallel^2$ can be rewritten as

$$\begin{aligned}\sum_{i,j}W_{i,j}(s_i^Ts_i-2s_i^Tt_j+t_j^Tt_j)&=\sum_i\left(\sum_j W_{i,j}\right)s_i^Ts_i\\&\quad-2\sum_{i,j}W_{i,j}s_i^Tt_j\\&\quad+\sum_j\left(\sum_i W_{i,j}\right)t_j^Tt_j\\&=\mathrm{tr}(sD^rs^T)\\&\quad-2\,\mathrm{tr}(sWt^T)+\mathrm{tr}(tD^ct^T),\end{aligned}$$

where D^r, D^c are the diagonal matrices with $D_{i,i}^r=\sum_j W_{i,j}$, $D_{j,j}^c=\sum_i W_{i,j}$. The cost function (1) can be rewritten as

$$\begin{aligned}C(s,t)&=\mathrm{tr}(sL^{\mathcal{X}}s^T)+\mathrm{tr}(tL^{\mathcal{Y}}t^T)+\lambda(\mathrm{tr}(sD^rs^T)-2\,\mathrm{tr}(sWt^T)+\mathrm{tr}(tD^ct^T))\\&=\mathrm{tr}\left([s\ \ t]\begin{pmatrix}L^{\mathcal{X}}+\lambda D^r & -\lambda W\\ -\lambda W^T & L^{\mathcal{Y}}+\lambda D^c\end{pmatrix}\begin{pmatrix}s^T\\ t^T\end{pmatrix}\right).\end{aligned}\tag{2}$$

Denote

$$L=\begin{pmatrix}L^{\mathcal{X}}+\lambda D^r & -\lambda W\\ -\lambda W^T & L^{\mathcal{Y}}+\lambda D^c\end{pmatrix},\quad D=\begin{pmatrix}D^{\mathcal{X}} & 0\\ 0 & D^{\mathcal{Y}}\end{pmatrix},\ y=[s\ \ t].\tag{3}$$

It is easy to follow that $D_{ii}=\sum_j L_{ij}$. And the cost function (2) can be rewritten as

$$C(y)=\mathrm{tr}(yLy^T).\tag{4}$$

To remove an arbitrary scaling factor in the embedding, the constraint $yDy^T=I$ is generally imposed on the cost function. Then the embedding coordinates of X and Y can be obtained by computing d eigenvectors $y_1^T,\ldots,y_d^T$ corresponding to the d smallest eigenvectors of the generalized eigenvalue decomposition

$$Ly^T=\gamma Dy^T.\tag{5}$$

The embedding results s and t can be obtained from the top n and the next m columns of $y=[y_1^T,\ldots,y_d^T]^T$.

In the cost function (2), it directly computes the embedding results s and t rather than the mapping functions f and g. And it is difficult to handle new points. To learn the linear mapping function f, g, the constraints $s=f^TX$ and $t=g^TY$ can be imposed on (2), i.e.

Download English Version:

https://daneshyari.com/en/article/4947822

Download Persian Version:

https://daneshyari.com/article/4947822

Daneshyari.com