Neurocomputing 174 (2016) 18–30

Contents lists available at ScienceDirect

Neurocomputing

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

Manifold learning in local tangent space via extreme learning machine

Qian Wang [a,b], Weiguo Wang [b], Rui Nian [a,*], Bo He [a], Yue Shen [a], Kaj-Mikael Björk [c], Amaury Lendasse [c,d]

[a] College of Information Science and Engineering, Ocean University of China, 266100 Qingdao, China
[b] School of Mathematical Sciences, Ocean University of China, 266100 Qingdao, China
[c] Arcada University of Applied Sciences, 00550 Helsinki, Finland
[d] Department of Mechanical and Industrial Engineering and the Iowa Informatics Initiative, The University of Iowa, Iowa City, IA 52242-1527, USA

ARTICLE INFO

Article history:
Received 30 September 2014
Received in revised form
17 March 2015
Accepted 23 March 2015
Available online 13 August 2015

Keywords:
Extreme learning machine
Manifold learning
Local tangent space alignment
High-dimensional space

ABSTRACT

In this paper, we propose a fast manifold learning strategy to estimate the underlying geometrical distribution and develop the relevant mathematical criterion on the basis of the extreme learning machine (ELM) in the high-dimensional space. The local tangent space alignment (LTSA) method has been used to perform the manifold production and the single hidden layer feedforward network (SLFN) is established via ELM to simulate the low-dimensional representation process. The scheme of the ELM ensemble then combines the individual SLFN for the model selection, where the manifold regularization mechanism has been brought into ELM to preserve the local geometrical structure of LTSA. Some developments have been done to evaluate the inherent representation embedding in the ELM learning. The simulation results have shown the excellent performance in the accuracy and efficiency of the developed approach.

1. Introduction

In the Era of Big Data, how to deal with the exponential explosive growth of high-dimensional data turns out to be one of the most challenging topics. The manifold learning has attracted a great deal of interests in recent years. The intuition that high-dimensional data are most likely to lie in or close to only a few intrinsic degrees of freedom than the ambient dimension would suggest, it is probable for us to estimate the geometrical properties of high-dimensional data from relatively simple topology manifolds in the lower dimensional space. Therefore, it is very important to discover and preserve the intrinsic geometry structure of the raw data on the low-dimensional subspaces [1].

As to the manifold learning, the classic linear dimension reduction techniques like principal component analysis (PCA) [2] and linear discriminant analysis (LDA) [3] fail to discover the underlying nonlinear manifold structure and preserve the local geometry. Recently, progress has been made in developing efficient algorithms to be able to learn the low-dimensional structure of nonlinear manifolds. These proposed methods include isometric feature mapping (ISOMAP) [4], locally linear embedding (LLE) [5], Laplacian eigenmap (LE) [6], Hessian LLE (HLLE) [7], and local tangent space alignment (LTSA) [8,9]. Since LTSA is from the geometrical intuition and straightforward to implement, it has received wide attention in the nonlinear manifold learning [1,8,9]. However, one of the bottlenecks comes from the computational speed, as LTSA needs to obtain every eigenvalue and eigenvector when aligning the local tangent coordinates in the local tangent space.

In the context of the machine learning, Artificial Neural Networks (ANNs) have been playing the dominant roles because of its benefits on generalization, flexibility, nonlinearity, fault tolerance, self-organization, adaptive learning, and computation in parallel. Recently, extreme learning machine (ELM) has attracted more and more attention in machine learning by providing the higher generalization performance at a much faster speed [10,11]. ELM was originally developed for the single hidden layer feedforward networks (SLFNs) instead of the classical gradient-based algorithms [10–12], then extended to the generalized SLFN that need not be the neuron alike [13,14], and can work for the conventional SVM and its variants. The essence of ELM is that when the input weights and the hidden layer biases are randomly assigned, the output weights can be computed by the generalized inverse of the hidden layer output matrix [10,11]. There are a great many ELM variations that have been proposed, including the random hidden layer feature mapping-based ELM [15], the Kernel-based ELM [15–17], the fully complex ELM [18], the incremental ELM (IELM) [12–14], the online sequential ELM (OS-ELM) [19–21], the pruning ELM (P-ELM, OP-ELM) [22,23], the circular-ELM (C-ELM) [24], and ELM ensembles [25–28], which have led to the

* Corresponding author.
E-mail address: nianrui_80@163.com (R. Nian).

http://dx.doi.org/10.1016/j.neucom.2015.03.116
0925-2312/

state-of-the-art results in many applications both for the regression [29,30] and for the pattern recognition problem [31–36].

In this paper, we come up with a fast and efficient manifold learning approach for the high-dimensional data via ELM. We first generate the underlying low-dimensional manifold coordinate by the local tangent space alignment and then establish a set of SLFNs in the ELM ensemble to simulate the entire process of manifold learning, and the relevant mathematical criterion of the model selection will also be developed for the performance improvements.

The rest of the paper is organized as follows: in Section 2, the basic of the ELM is outlined. In Section 3, the local tangent space alignment method will be briefly introduced. In Section 4, a fast and efficient scheme for the low-dimensional manifold learning via ELM is developed in detail. Section 5 is about the simulation and result analysis, and Section 6 comes to the conclusions.

2. Basic ELM

So far, ELM learning has been developed to work at a much faster learning speed with the higher generalization performance, both in the regression problem and in the pattern recognition. For the given N learning samples $\{x_i, y_i\}_{i=1}^N$, where $x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T$ and $y_i = [y_{i1}, y_{i2}, \ldots, y_{im}]^T$, the standard model of the ELM learning can be written as the following matrix format:

$$H\beta = Y \qquad H(x) = \begin{bmatrix} h_1 & h_2 & \ldots & h_L \end{bmatrix}$$
$$= \begin{bmatrix} h_1(x_1) & \ldots & h_L(x_1) \\ \vdots & h_i(x_j) & \vdots \\ h_1(x_N) & \ldots & h_L(x_N) \end{bmatrix}$$
$$= \begin{bmatrix} g(\omega_1 \times x_1 + b_1) & \ldots & g(\omega_L \times x_1 + b_L) \\ \vdots & g(\omega_i \times x_j + b_i) & \vdots \\ g(\omega_1 \times x_N + b_1) & \ldots & g(\omega_L \times x_N + b_L) \end{bmatrix}_{N \times L}$$
$$\beta = [\beta_1, \ \beta_2, \ldots, \beta_L]^T_{m \times L}, \quad Y = [y_1, \ y_2, \ldots, y_N]^T_{m \times N}. \tag{1}$$

where $\omega_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{in}]^T$, $i = 1, 2, \ldots, L$ is the weight vector connecting the ith hidden neuron and the input neurons, $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ is the weight vector connecting the ith hidden neuron and the output neurons, $\omega_i \times x_j$ denotes the inner product of ω_i and x_j, $j = 1, 2, \ldots, N$, and there are L hidden neurons with the activation function $g(x)$. All kinds of the activation functions can be chosen here, such as the Sigmoid function, the hard-limit function, the Gaussian function, and the multiquadric function.

If the activation function $g(x)$, ω and b are all set, the only learning parameter will be β. Different from the traditional learning algorithm, ELM tends to achieve the least training error and the least norm of output weight together. According to Bartlett's theory [37], when the feedforward neural networks get smaller training error, the norms of weights are smaller, and the generalization performance of the networks is better, $\beta = \arg\min(\|H\beta - Y\|^2, \ \|\beta\|)$. In order to solve the formation, both the standard optimization method and the minimal norm least square method need to be adopted. The original implementation of ELM will then be $\beta = H^+ Y$, where H^+ denotes the Moore–Penrose generalized inverse of matrix H [10]. The orthogonal projection method can be used here when $H^T H$ is nonsingular and $H^+ = (H^T H)^{-1} H^T$, or when HH^T is nonsingular and $H^+ = H^T (HH^T)^{-1}$. In addition, the resulting solution tends to be more stable with better generalization performance by adding a positive value to the diagonal of HH^T or $H^T H$ [38].

3. The local tangent space alignment algorithm

The local tangent space alignment (LTSA) algorithm can mathematically explore the underlying low-dimensional manifold in the high-dimensional space by intuition. Let $\mathcal{X} = \{X_1, \ X_2, \ \ldots, X_N\} \subset R^n$ be the original input with N samples embedded in an n-dimensional space which is actually drawn from an underlying m-dimensional manifold. Let $\mathcal{X}_i = \{X_{i1}, X_{i2}, \ldots, X_{ik}\}$ be denoted as the k neighbors extracted by the K-nearest Neighbor (K-NN) method for each data sample X_i, $m \le k < N$. The basic model of LTSA is to find a set of orthonormal bases $\mathcal{U}_i = \{U_{i1}, \ U_{i2}, \ldots, U_{im}\} \subset R^n$ by the following optimization:

$$\min_{\boldsymbol{U}_i^T \boldsymbol{U}_i = I_m} \sum_{j=1}^{k} \|X_{ij} - \overline{X}_i - \boldsymbol{U}_i \boldsymbol{U}_i^T (X_{ij} - \overline{X}_i)\|_2^2, \tag{2}$$

where $\overline{X}_i$ is the center of $\mathcal{X}_i$, $\overline{X}_i = 1/k \sum_{j=1}^k X_{ij}$, $\boldsymbol{U}_i = [U_{i1}, \ U_{i2}, \ldots, U_{im}] \in R^{n \times m}$, and I_m is the identity matrix of the order m.

Further, let $\tilde{X}_i = [X_{i1} - \overline{X}_i, \ X_{i2} - \overline{X}_i, \ldots, X_{ik} - \overline{X}_i]$ be denoted as a centered matrix. Then by some algebraic deductions the above optimization problem can be transformed into

$$\max_{\boldsymbol{U}_i^T \boldsymbol{U}_i = I_m} \mathrm{Tr}(\boldsymbol{U}_i^T \tilde{X}_i \tilde{X}_i^T \boldsymbol{U}_i), \tag{3}$$

where $\mathrm{Tr}(\cdot)$ is short for the trace of the square matrix, i.e., the sum of all the elements on the main diagonal (from upper left to lower right) in the matrix.

It is clear to see that $U_{i1}, \ U_{i2}, \ldots, U_{im}$ are the eigenvectors of $\tilde{X}_i \tilde{X}_i^T$ corresponding to the m largest eigenvalues. The local coordinate of each neighbor X_{ij} can then be formulated in the approximated tangent space as $a_{ij} = \boldsymbol{U}_i^T (X_{ij} - \overline{X}_i)$, $j = 1, 2, \ldots, k$ so as to obtain the local coordinate matrix $\boldsymbol{A}_i = [a_{i1}, a_{i2}, \ldots, a_{ik}]$, $i = 1, 2, \ldots, N$.

The set of the global coordinate for the manifold $\mathcal{T} = \{t_1, t_2, \ldots, t_N\}$ can be acquired by an affine transformation $G_i : a_{ij} \to t_{ij}, j = 1, 2, \ldots, k$. The total error of the alignment is denoted as $\varepsilon(\boldsymbol{T}) = \mathrm{Tr}(\boldsymbol{T} \boldsymbol{\Phi} \boldsymbol{T}^T)$, where $\boldsymbol{T} = [t_1, t_2, \ldots, t_N] \in R^{m \times N}$ is the global coordinate system of the low-dimensional representation, and $\boldsymbol{\Phi} = \sum_{i=1}^N P_i V_i V_i^T P_i^T$ is the alignment matrix, P_i is a $N \times k$ selection matrix with $\boldsymbol{T} P_i = [t_{i1}, \ t_{i2}, \ldots, t_{ik}]$, i.e., P_i can select k elements of $\boldsymbol{T}$, $V_i = (I_k - 1_k 1_k^T / k)(I_k - \boldsymbol{A}_i^+ \boldsymbol{A}_i)$, 1_k is a $k \times 1$ vector which has all ones and $\boldsymbol{A}_i^+$ is the Moore–Penrose generalized inverse: $\boldsymbol{A}_i^+ = \boldsymbol{A}_i^T (\boldsymbol{A}_i \boldsymbol{A}_i^T)^{-1}$.

In this way, LTSA can be finally written as the following optimization problem:

$$\min_{\boldsymbol{T}\boldsymbol{T}^T = I_m} \mathrm{Tr}(\boldsymbol{T} \boldsymbol{\Phi} \boldsymbol{T}^T). \tag{4}$$

where $\boldsymbol{T}\boldsymbol{T}^T = I_m$ is the constraints to uniquely determine $\boldsymbol{T}$. Since that the vector 1_N of all ones is an eigenvector of $\boldsymbol{\Phi}$ corresponding to a zero eigenvalue, therefore, the rows of the low-dimensional representation $\boldsymbol{T}$ are given by the eigenvectors of $\boldsymbol{\Phi}$ corresponding to the second to the $(m+1)$th smallest eigenvalues [1,8,9].

4. Fast manifold learning via ELM

4.1. General model

The basic idea here is that we feed the manifold production of LTSA about the visual perception into the ELM architecture and simulate the whole process in a faster way. It is found that, in the human visual system, the visual layers share components that provide common features encountered for all the visual tasks. Let the original input $X = [X_1, \ldots, X_q, \ldots, X_Q]$ in the training set consist of Q classes, with each class X_q composed of R samples, i.e., $X_q = \{x_{q1}, \ldots, x_{qr}, \ldots, x_{qR}\}$. After preprocessing, the normalized input can be written as $X' = [x'_1, x'_2, \ldots, x'_N] \in R^{n \times N}$ in an n-dimensional space, $N = R \times Q$. Let the normalized global coordinate of the manifold for the input in the training set be represented as $\boldsymbol{T}' = [t'_1, t'_2, \ldots, t'_N] \in R^{m \times N}$ by LTSA. Once a set of SLFNs is reasonably established via ELM leaning with the help of LTSA, the test set can