Study on improved LLE algorithm based on a sample set of well-distributed and weights matrix

Song Fei*, Cui Zhe

Chengdu Computer Institute of Chinese Academy of Science

Received 1 March 2014, www.tsi.lv

Abstract

There are large amounts of data has accumulated along with technology of computer, information and network developed. How can we using these data and mining out the valuable information are hot topics in information processing field. There are some distress and difficulties caused by the high-dimensional data on data modelling and data analysis. In this paper, a local linear embedding algorithm based on the improved uniform sample set and the weight value matrix is proposed. The test shows that the improved dimensionality reduction algorithm accuracy is significantly higher than the original LLE algorithm.

Keywords: LLE, Well-distributed, Weights Matrix

1 Introduction

Big data cause a lot of inconvenience on applications because of the diversity and complexity of the data. These rich data resources bring convenience to people but have also brought a lot of problems at the same time. Such as Information overload, Data is difficult to choose and useful information submerged in the massive amounts of data, etc. Faced with these difficult to deal with data, we cannot mining out the effective information implied in big data and cannot speculate on future trends if we have no effective means for analysis and processing[1].

Data dimensionality reduction techniques [1-2] can be used to solve the above-mentioned problems. It can explore the internal structure and association of the original data, and eliminated redundant data, improved efficiency of computation. It can also improve the understand ability of the data to improve the accuracy of data [2].

2 Dimensionality reduction principle

Depending on the type of the data to be processed, existing dimensionality reduction algorithm [3-4] is divided into two categories: linear dimension reduction technique and nonlinear dimensionality reduction technique.

Define: Let $\mathbf{X}=(x_1,x_2,...,x_D)^{\mathrm{T}}$ is a vector in high-dimensional space, by the following formula:

$$\mathbf{F}(\mathbf{X})=\begin{pmatrix} F_1(\mathbf{X}) \\ F_2(\mathbf{X}) \\ ... \\ F_d(\mathbf{X}) \end{pmatrix}=\begin{pmatrix} F_1(x_1,x_2,...,x_D) \\ F_2(x_1,x_2,...,x_D) \\ ... \\ F_d(x_1,x_2,...,x_D) \end{pmatrix}.$$

We can get a vector $\mathbf{Y}=(y_1,y_2,...,y_D)^{\mathrm{T}}$ in low-dimensional space. If each component F_i of **F** is a linear function, then **F** is a linear dimensionality reduction. Otherwise, **F** is nonlinear dimensionality reduction.

3 Locally linear embedding algorithm LLE analyses

Locally linear embedding algorithm that proposed by Roweis and Saul [5] in 2000. It is a method for nonlinear dimensionality reduction. Its core idea is using local linear approaching global nonlinear, keep the geometry structure of local sample points unchanged and using local neighbourhood data that overlapped with each other to provide global information. So as to maintain the overall geometric properties of whole sample points [3].

Assuming that the sample set consists of N D-dimensional vector X_i and the entire sample points in a $d(d \le D)$ -dimensional manifold. When we have a sufficient number of sample points, we can think approximately that each sample point and its adjacent sample points in a local linear manifold. In this case, each sample point can be represented with a linear combination of its adjacent sample points that possessed weighting coefficient [4]. These weighting coefficients reflect the local geometry information of a small area. We can use this information to seek out a low-dimensional embedding space that keeps the geometrical characteristics of the original high-dimensional space. We can get the overall information of the original high-

* ***Corresponding author*** e-mail: asfei@aliyun.com

dimensional space by the overlapping local neighbourhood and get a global coordinate system [5]. The greatest advantage of this method is that the data does not produce a large offset. That because of the curvature of the manifold is very small so it can be approximated as flat.

LLE using matrix $X_{D,N}$ as it input and matrix $Y_{d,N}$ as it output. Here $X_{D,N}$ and $Y_{d,N}$ are composed by N d-dimensional vector ($d<<D$) and the K^{th} column of matrix Y corresponding to K-th column of matrix X.

The algorithm is divided into three steps.

1) Looking for a point and its adjacent point, constitute a piece of local adjacent area. For a sample point X_i (i=1,2,…,N) that in the high-dimensional space, we can calculate the distance between X_i and the other N-1 sample point. According to their distances, we can find a close neighbour points of X_i. We usually measure the distance between two points using the Euclidean distance, that is $d_{ij} = \|X_i - X_j\|$. There are two ways for choosing nearest neighbour points: Choosing the K points that have minimum distance from X_i as the adjacent points of X_i. All the points that X_i as the centre and all the points within the sphere of radius ε are the adjacent points of X_i. We generally use the first method to determine the adjacent points. Each point in the space have the same number of neighbour points due to .the number of K is defined. Thus, calculate getting more simple and convenient.
2) Calculate the weight of X_i and its nearest neighbour points. Weight values describe the degree of approximation between the two points. When we defined X_i and found its K adjacent points, we need to calculate the weights between this point and each of its adjacent point. Assume that X_j is a close neighbours point of the of X_i, then the weight between them is $W_{ij} = e^{\frac{|x_i - x_j|}{2\sigma^2}}$, wherein σ is a parameter. The calculated weight W_{ij} that between X_i and each of its adjacent points, and let the error minimum when X_i is reconstructed by this K points, then: $\min \varepsilon(W) = \sum_{i=1}^{N} \left\| X_i - \sum_{j=1}^{N} w_{ij} X_j \right\|$.

   In order to ensure translational invariance, make $\sum_{j=1}^{N} w_{ij} = 1$. If we want to make $\varepsilon(W)$ minimum, then: $\varepsilon(W_i) = \left\| X_i - \sum_{j-1}^{N} w_{ij} X_j \right\|^2 = \sum w_{ij} w_{ik} c_{jk} = W_i C W_i'^T$, where $W_i' = (w_{i1}', w_{i2}', w_{i3}' \cdots w_{iK}')$ are K components that value are not 0 of W_i. Now the loss function can be rewritten as: $\min \varepsilon(Y) = \sum_{i=1}^{N} \sum_{j=1}^{N} M_{ij} y_i^T y_j$.

   Construct a local covariance matrix C: $c_{jk} = (X_i - X_j)^T (X_i - X_k)$.

   Set $W_i' l = 1$, $l = \underbrace{(1,1,1 \cdots 1)}_{K}^T$, by the Lagrange multiplier method to obtain: $\min L = W_i' C W_i'^T + \lambda(W_i' l - 1)$,

   $\frac{\partial L}{\partial W_i'} = 2W_i' C + \lambda l^T = 0 \quad ===> W_i' = \frac{l^T C^{-1}}{l^T C^{-1} l}$.

   When $K>D$, C is a singular matrix. Therefore, we need a renormalization operation for it. We sum regular numbers on the diagonal of the singular matrix: $C_{jk} \leftarrow C_{jk} + \delta_{jk} (\frac{\Delta^2}{K}) Tr(C)$, where $Tr(C)$ is the trace of C, $\Delta^2 << 1$.

   $C_{jk} \leftarrow C_{jk} + \delta_{jk} (\frac{\Delta^2}{K}) Tr(C)$, where $Tr(C)$ is the trace of C, $\Delta^2 << 1$.
3) Calculated points Y_i in low-dimensional space. The final step of LLE is a calculating of the value of low-dimensional embedding space according to samples X_i in the high-dimensional space and the weight values W_{ij}. W_{ij} is the weight value between X_i and its adjacent point X_j. To make the low-dimensional space as much as possible be consistent with the partial linear structure in the high-dimensional space, a local information W_{ij} should be fixed. We should minimize a loss function $\phi(Y) = \sum_{i=1}^{N} \left\| Y_i - \sum_{i=1}^{N} w_{ij} Y_j \right\|^2$, taking into account requirements for $\phi(Y)$ without deformation, translation, rotation and scaling transformation. Thus, $\sum_{i=1}^{N} Y_i = 0$ and $\frac{1}{N} \sum_{i=1}^{N} Y_i Y_j^T = 1$. So, $\phi(Y) = \sum_{i=1}^{N} \left\| Y_i - \sum_{i=1}^{N} w_{ij} Y_j \right\|^2$, $= tr((Y - WY)^T (Y - WY)) = tr(Y^T M Y)$, wherein M is an $N * N$ stacked matrix: $M = (1-W)^T (1-W)$.

At this moment, the minimization solution of a loss function is eigenvectors matrix that consisted by several minimum eigenvalues of matrix M. We take the non-feature vector that corresponding m eigenvalues of matrix M according to the order of small to large. Because the smallest eigenvalue is infinitely close to 0, therefore we discarded this eigenvector so as to satisfy the condition

$\sum_{i=1}^{N} Y_i = 0$. The rest of 2 to m +1 feature vectors will compose a matrix. This matrix is the samples in the low-dimensional space. LLE algorithm processes are shown in Figure 1.

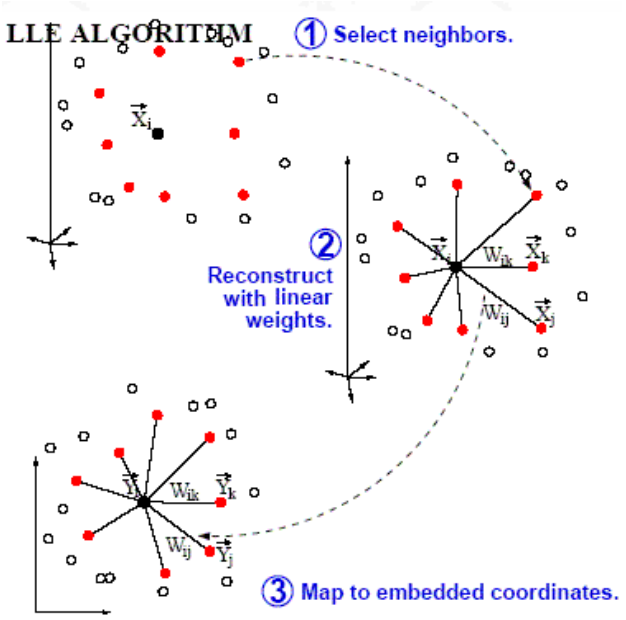

FIGURE 1 LLE algorithm processes

4 The improved LLE algorithm - DLLE

The data that correspond to LLE algorithm is static. That is every time it will enter the entire sample set and then mapped sample set to a low-dimensional embedding space and get the corresponding samples from the embedding space finally. When a new sample point (the new data) is added, the new sample point and the original sample must be merged into a new sample set. The new sample set will be re-entered into LLE algorithm and be running.

K value of LLE algorithm indicates the quantity of selected adjacent sample points. LLE is very sensitive on *K*. The larger the value *K*, the greater the difference of the geometric characteristics of high-dimensional space manifold. This will be lead to the range of data mining conclusions greater and lead to reduced accuracy. Eventually. But, if the value of *K* is not big enough (i.e, the number of adjacent points is not enough), then this may cause the continuous manifold in high-dimensional space split into disconnected submanifold. Obtained conclusions by data mining can be completely unrelated with expected ones. References [6-8] have analysed how to select the range of *K* value. For the large number of data, the ranges of *K* value are between 5 and 20 generally.

In the really big data mining process, we found that the different component values have different effects on mining conclusions. This means that the weights values of each component importance are not the same and the different values will have a huge impact on the actual results. We propose an improved algorithm to solve the considered problems.

In order to reduce the sensitivity of the LLE algorithm on the value of K, we design a new method for definition of distance, where the average of the distance between a central point and its *K* neighbouring points is represented.

This new approach allows to obtain the distance between the samples in the sample-point-intensive areas (relatively increased) and the distance between the samples in the sample point's sparse area (relatively narrow). Thus, the distribution of the sample set leads to homogenizing and reducing of the impact on the K value for LLE calculation result [6].

For solution of the problem of weights values between components, we present an idea of embedded importance weights into LLE algorithm.

Indicating the original sample set, the paper contains N samples, setting an importance weight vectors. Each component is a positive number and the sum of all the components is equal to 1. Calculating a new eigenvector using LLE algorithm of original sample set, we can get the weight matrix and the sample, which in embedding space [7].

Making the neighbours of all the sample points in the original sample set non-changed, we can get the weight matrix of the samples space. Finally, by step 3 (section 3), we obtain the sample set of low-dimensional space, realizing data dimensionality reduction.

5 Experiments and conclusions

Randomly select a UCI database as experimental data to compare the effect of the improved DLLE algorithm and traditional LLE algorithm. This database contains a training set and a test set. 2000 samples from the training set to do the training object and take 1000 samples from the test set to do the test were taken. The principle of nearest neighbour to look for the K-nearest neighbours of the training set for each test sample was used. The range of dimension d in embedding space is from 2 to 10 and the range of K is from 2 to 10. Two experiments use the same training samples and the same test samples. The result of the experiment is as follows tables.

Based on the data in Table 1 and Table 2 can be seen that the error rate of improved LLE algorithm lower than the error rate of original LLE algorithm and the accuracy of dimensionality reduction higher than the original LLE algorithm obviously.

TABLE 1 The average error rate of the original LLE algorithm

	2	3	4	5	6	7
10	0.192	0.124	0.118	0.092	0.082	0.076
11	0.212	0.136	0.112	0.084	0.072	0.066
12	0.256	0.182	0.158	0.108	0.098	0.092
13	0.328	0.174	0.144	0.120	0.116	0.110
14	0.282	0.168	0.138	0.104	0.108	0.098
20	0.386	0.364	0.332	0.324	0.302	0.284

TABLE 2 The average error rate of the improved LLE algorithm

	2	3	4	5	6	7
10	0.142	0.118	0.084	0.080	0.078	0.062
11	0.164	0.130	0.096	0.088	0.074	0.068
12	0.208	0.166	0.104	0.094	0.082	0.076
13	0.220	0.158	0.118	0.106	0.098	0.104
14	0.234	0.144	0.126	0.104	0.102	0.086
15	0.246	0.182	0.154	0.126	0.116	0.104
16	0.262	0.262	0.186	0.148	0.124	0.108
17	0.266	0.294	0.242	0.186	0.158	0.132
18	0.274	0.308	0.278	0.230	0.212	0.188
19	0.280	0.314	0.306	0.264	0.232	0.206
20	0.298	0.344	0.312	0.272	0.238	0.220

References

[1] Yan S, Bouaziz S, Lee D, Barlow J 2012 *Neurocomputing* **76**(1) 114-24
[2] Huang W, Yin H 2012 *Image and Vision Computing* **30**(4) 355-66
[3] Cho J H, Kurup P U 2011 *Sensors and Actuators* B: *Chemical* **160**(1) 542-8
[4] Gu H, Deng S P, Wang X, Shi Jin-Qin, Jin Jian-Qiu, Tian Shi-Yi 2012 *Sensors and Actuators* B: *Chemical* **163**(1) 281-9
[5] Roweis S T, Saul L K 2000 *Science* **290**(5500) 2323-6
[6] Xing E P, Ng A Y, Jordan M I, Russell S 2003 Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems* **15** 505-12 MA: MIT Press
[7] Tsai F S 2012 *Expert Systems with Applications* **39**(2) 1747-52
[8] Raymer M L, Punch W F, Goodman E D, Kuhn L A, Jain A K 2000 *IEEE Transactions on Evolutionary Computation* **4**(2) 164-71

Authors

Fei Song, born on October 31, 1982, Chengdu, Sichuan, China

Current position, grades : PhD student in Chengdu Institute of Computer Application, Chi Doctor
University studies: PhD studies Chengdu Computer Institute of Chinese Academy of Science
Scientific interest: Data Disaster Recovery, Reliability Engineering and Network coding
Publications : 4
Experience: Academy of Sciences, China. He has published more than four articles in reputed international journals and International Conferences
Research interests: Data Disaster Recovery, Reliability Engineering and Network coding

Zhe Cui, born on September 20, 1997, Chengdu, Sichuan, China

Current position, grades: Professor in Chengdu Institute of Computer Application, Chinese Academy of Sciences, China.
University studies: Chengdu Computer Institute of Chinese Academy of Science
Scientific interest: Trusted Computing, Embedded Systems and Reliability Engineering
Research interests: Trusted Computing, Embedded Systems and Reliability Engineering