International Journal of Hybrid Information Technology
Vol.9, No.4 (2016), pp. 417-426
http://dx.doi.org/10.14257/ijhit.2016.9.4.36

A Self-adaptive Spectral Clustering Based on Geodesic Distance and Shared Nearest Neighbors

Chunmiao Yuan, Kaixiang Fan, Xuemei Sun

School of Computer Science &Software, Tianjin Polytechnic University, Tianjin, China
cm_yuan@163.com

Abstract

Spectral clustering is a method of subspace clustering which is suitable for the data of any shape and converges to global optimal solution. By combining concepts of shared nearest neighbors and geodesic distance with spectral clustering, a self-adaptive spectral clustering based on geodesic distance and shared nearest neighbors was proposed. Experiments show that the improved spectral clustering algorithm can fully take into account the information of neighbors, but also measure the exact distance and better process the geodetic data.

Keywords: *Spectral clustering, Shared nearest neighbors, Geodesic distance, Subspace clustering*

1. Introduction

High-dimensional data processing has always been a hot and difficult subject in the field of data mining. With the number of dimensions increasing, curse of dimensionality is a very common problem. Due to excessive data attributes, a lot of uncertain factors increase, which will make data too sparse, thus increasing the difficulty of processing. To address this issue, many scholars have done a lot of research to varying degrees and reduce the impact caused by the curse of dimensionality in which dimensionality reduction and subspace clustering are currently hot researched.

Spectral clustering [1-2] is a method of subspace clustering, widely applied in the field of pattern recognition and data mining in recent years. Many traditional clustering algorithms, such as k-means algorithm, can yield better clustering results on the globular data, but not satisfactory performance on the data of other shapes, and easy to converge to local optimal solution. In contrast, the spectral clustering method [3] does not make the assumption about the global structure of the data, but directly solve the characteristic decomposition of graph Laplacian matrix to achieve clustering on the data of any shape and converge to the global optimal solution, also applied to non-convex data sets [4].

Based on research and analysis of spectral clustering algorithm, this paper obtains shortcomings of traditional spectral clustering algorithms, and combines concepts of shared nearest neighbors and geodesic distance with spectral clustering to propose a self-adaptive spectral clustering based on geodesic distance and shared nearest neighbors. The improved spectral clustering algorithm can fully take into account the information of neighbors, but also measure the exact distance. Experiments show that the algorithm can better process the geodetic data.

ISSN: 1738-9968 IJHIT

2. Shared Nearest Neighbors

2.1. The concept of shared nearest neighbors

Under normal conditions, the similarity measurement of two points adopts the Gaussian kernel function, $(x_i, x_j) = \exp(-\| x_i - x_j \|^2 / 2\sigma^2)$ and it is clear that two most important parts of the formula are the scale parameter σ, and the distance measure formula $\| x_i - x_j \|^2$. Here is the Euclidean distance. σ is a parameter, needed to be set. The final similarity derived from the formula only varies with changes of the Euclidean distance between two points. Once the distance is determined, regardless of the distribution of neighbors around the two points, the similarity is determined, regardless of the impact of neighbors between two points on the similarity. In fact, two points have neighbors, and their common neighbors are many, we can consider these two points may be more similar, and this is the very intuitive concept of shared nearest neighbors.

The traditional spectral clustering does not consider the impact of neighbors on the similarity of the two points, so the algorithm cannot process multi-scale data sets. To solve this problem, Manor et al proposed self-adaptive (self-adjusting) Gaussian kernel function, defined as follows:

$$S_T(x_i, x_j) = \exp(-\| x_i - x_j \|^2 / \sigma_i \sigma_j) \quad (1)$$

Wherein, σ_i and σ_j represent the Euclidean distance from the point x_i to the neighbor p. Using this formula to calculate the similarity relationship can more easily find the real cluster structure, and has important reference value for the precise definition of the similarity between the two data points. This spectral clustering algorithm is called self-tuning spectral clustering, referred to as SSC. This algorithm takes into account the impact of surrounding data points on the data point, to better process multi-scale data sets, able to accurately identify clusters with larger density difference.

However simple self-adaptive spectral clustering still cannot get a better clustering effect, so the similarity measure based on shared nearest neighbors appears.

Shared nearest neighbors was proposed by Jarvis and Patrick et al, referred as to SNN. SNN can be used to characterize the local density between two points. Combining the concept of shared nearest neighbors to spectral clustering forms spectral clustering based on the shared nearest neighbors. This spectral clustering re-defines the similarity of the two points, thus eliminating the impact of artificial value σ on the algorithm [5].SNN is defined as follows:

Set *kd* points nearest to the point x_i to construct the set $N(x_i)$, *kd* points nearest to the point x_j to construct the set $N(x_j)$,then the *kd* shared nearest neighbors of point x_i and x_j is $SNN(x_i, x_j) = |N(x_i) \cap N(x_j)|$.

2.2 Self-adaptive spectral clustering based on shared nearest neighbors(SSC-SNN)

The concept of SNN was added to the improved self-adaptive Gaussian kernel function in self-adaptive spectral clustering based on shared nearest neighbors, which enable it to characterize the local density information [6]. When a lot of data points connect between two points, the local density is relatively high, and SNN value will be relatively large, the similarity between two points should be also larger, namely the value of SNN and similarity is proportional.

Definition of self-adaptive Gaussian kernel function [7] based on shared nearest neighbors is as follows:

$$S_N(x_i, x_j) = \exp(-\frac{\| x_i - x_j \|^2}{\sigma_i \sigma_j (SNN(x_i, x_j) + 1)}) \quad (2)$$

Wherein, $\| x_i - x_j \|$ is the Euclidean distance between two data points, σ_i and σ_j are respectively the Euclidean distance from x_i and x_i to the neighbor P.

The difference between the newly constructed Gaussian kernel function and self-adaptive Gaussian kernel function is the introduction of SNN as part of the description of the data density, taking into account the local density information of data [8]. From the equation we can see that when two data points are located in sparse clusters, appropriately increase the similarity of the two points by adjusting the parameters, points in the coefficient cluster can be more easily and accurately to be clustered which will get the correct cluster result and avoid classifying uniform data points in sparse cluster into different clusters, and two points relatively close or located in the same cluster on the same points have high similarity.

There are a lot of kinds of spectral clustering algorithm, in which NJW [3] is a popular one proposed by Ng, Jordan and Weiss and et al. NJW uses the top *k* numbers of eigenvectors corresponding the maximum eigenvalues of Laplacian matrix to construct a new vector space *R* in which a corresponding relationship with the original data is set up, then clusters the data. We will use NJW to cluster in this paper.

SSC-SNN is the spectral clustering algorithm based on shared nearest neighbor, and the basic procedure of the algorithm is as follows:

Step1: Calculate $SNN(x_i, x_j)$, the number of shared kd neighbors of any point x_i and x_j.

Step2: According to S_N, calculate the similarity matrix *W* of the sample point, construct the similarity connection diagram and make *W* as the weight matrix of the similarity connection diagram.

Step3: Calculate non-normalized Laplacian matrix L_{sym}, $L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$.

Step4: Calculate the eigenvalue and eigenvectorof L_{sym}, set $\lambda_1, \lambda_2, ..., \lambda_k$ as top *k* largest eigenvalues ranking in order. $v_1, v_2, ..., v_k$ is the corresponding eigenvector of these *k* eigenvalues.

Step5: Rank eigenvectors $v_1, v_2, ..., v_k$ in rows to construct the matrix $V = [v_1, v_2, ..., v_k]$.

Step6: Normalize each row of the matrix V, make the norm of each row as 1, get the matrix U, namely $u_{ij} = v_{ij} / (\sum_k v_{ik}^2)^{1/2}$.

Step7: See each row of *U* as one point in the R^k space, and use *K*-mean value to cluster the rows into *k* categories.

The inputs of the algorithm are the sample similarity matrix *W* and the number of clusters *k*. From the process, it can be seen that the process of NJW spectral clustering does not change, and only the definition of similarity changes.

3. Geodesic Distance

3.1. The concept of geodesic distance

Geodesic distance [9-11] is an important concept in the mathematical morphology, mainly for watershed segmentation (also called catchment area, which means that the water flow and other substances expel from a common discharge outlet to form a

centralized drainage area). The concept of geodesic distance is applied in many algorithms, primarily to process some special data forms.

In machine learning, the Euclidean distance is the most common way to define the distance, also the easiest to understand, originating from the distance formula between two points in Euclidean space. The Euclidean distance between two n-dimensional vectors $a(x_{11}, x_{12}, ..., x_{1n})$ and $b(x_{21}, x_{22}, ..., x_{2n})$ is defined as:

$$d_{ab} = \sqrt{\sum_{k=1}^{n} (x_{1k} - x_{2k})^2} \tag{3}$$

Although the Euclidean distance is simple and easy to understand and implement, it still has significant disadvantages. It treats equally the difference between different attributes (or known as variables) of the sample, so different distance definitions may be used sometimes. In addition to the Euclidean distance, common distance measure methods include Manhatten distance, Chebyshev distance, Mahalanobis distance, Hamming distance, the distance of correlation coefficient, etc. These distance measures all have their own advantages and disadvantages.

But in real life, we will encounter many distance measure problems that Euclidean distance cannot deal with, such as driving on the mountain road, or laying the pipeline in an area, which forces us to seek another distance measure method, thus "avoiding" similar problems, and geodesic distance can achieve that [12].

The basic idea of geodesic distance [13] is: When two points are very close, geodesic distance equals to the Euclidean distance, while when two points are relatively distant, geodesic distance is accumulative based on geodesic distance between neighboring points, which is an iterative distance measure method. Geodesic distance is the accurately measured distance between two points in Euclidean space, able to accurately reflect the true distance distribution of data in the continuous data space. For two data points relatively close, the geodesic distance is the Euclidean distance generated along the curved surface of the data distribution; for two data points relatively distant, firstly, calculate the distance between the sample point and its adjacent points and then adopt iteration to calculate the distance between these two data points. Therefore, when the sample point distribution is curved or manifold, the distance derived from the geodesic distance is more authentic.

3.2 The calculation of geodesic distance

Geodesic distance calculation is as follows:

Step1: Define the neighborhood size *k*, according to neighborhood size to structure geodesic distance, and go to the second step;

Step2: The distance between the observation point x and its adjacent point *k* equals to Euclidean distance, or regarded as the infinite, that is

$$d_G(x_i, x_j) = \begin{cases} d_E(x_i, x_j), & if\ (x_i, x_j)\ is\ neighbors \\ \infty, & if\ (x_i, x_j)\ is\ non-neighbors \end{cases} \tag{4}$$

Step3: Use data point *t* as the relay point, calculate the geodesic distance between the observation point *x* and relatively distant points, and iteratively calculate the geodesic distance between the observation point and all other points.

$$d_G(x_i, x_j) = \min\{ d_G(x_i, x_j), d_G(x_i, x_t) + d_G(x_t, x_j) \} \tag{5}$$

4. Self-adaptive spectral clustering based on geodesic distance and shared nearest neighbors(SSC-GD&SNN)

This paper introduces the geodesic distance into SNN self-adaptive spectral clustering in order to compensate for the deficiency of the previous algorithm about geodetic data processing and the kernel function used by the algorithm is as follows:

$$S_{GN}(x_i, x_j) = \exp\left(-\frac{d_G(x_i, x_j)}{\sigma_i \sigma_j (SNN(x_i, x_j) + 1)}\right) \tag{6}$$

$$\text{Meets } d_G(x_i, x_j) = \begin{cases} d_E(x_i, x_j) & if\ (x_i, x_j)\ are\ neighbors \\ \infty & if\ (x_i, x_j)\ are\ not\ neighbors \end{cases}$$

In addition to advantages of SNN self-adaptive kernel function, the new kernel function can process geodetic data. For spectral clustering, we still use the popular NJW algorithm. The algorithm process of self-adaptive spectral clustering based on geodesic distance and SNN is as follows:

Step1: Get data.

Step2: Use geodesic distance to calculate the distance between data points.

Step3: Use SNN self-adaptive kernel function to calculate similarity matrix.

Step4: Use NJW spectral clustering algorithm to complete clustering work.

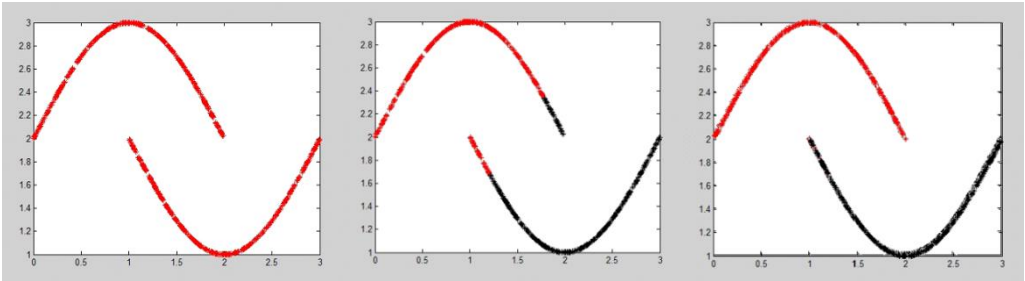
5. Experiment result

5.1 Experiment environment

In order to verify the algorithm result, we conducted simulation experiment, compared and analyzed the traditional spectral clustering, SSC-SNN and SSC-GD&SNN. All experiments use Matlab 6.0 programming, running on Windows7 operating system, executed on a PC with 2.1GHZ CPU and 2G memories. Respectively, use the circular data, bi-moon data, and hat-shaped data as experimental data.

5.2 Bi-moon data

SSC-SNN can self-define the scale parameter, but also take advantage of the information of neighbors. In order to verify the algorithm result, we constructed a similar bi-moon shaped data for the experiment. The initial shape of data is shown as Figure 1 (a). The clustering result obtained from traditional spectral clustering (NJW) is shown in Figure 1 (b), and the clustering result obtained from SSC-SNN is shown in Figure 1 (c).



(a) The initial shape (b) Clustering result of NJW (c) Clustering result of SSC-SNN

Figure 1. Bi-moon shaped data

From Figure 1 (b), it can be seen that, the traditional spectral clustering does not consider the relationship between neighbors, and simply clusters the two sets of data

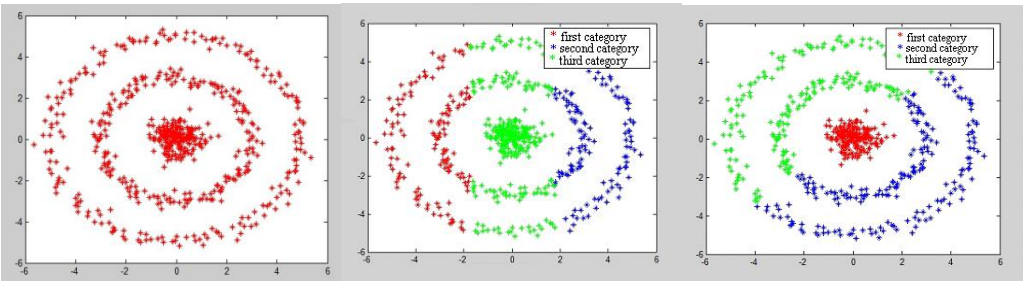
based on the distance, which results in the problem that the clustering of the middle part is not exact. But in Figure 1 (c), by using SSC-SNN to cluster bi-moon shaped data, and it can be found that the spectral clustering using the concept of shared nearest neighbors can better process bi-moon shaped data. SSC-SNN takes full advantage of the neighbor information between two data points, and gets a more accurate similarity measure, which can prove that SSC-SNN has a better data processing performance than traditional spectral clustering.

5.3 Circular data

In 5.2, it has been proved that shared nearest neighbor based self-adaptive spectral clustering can more accurately process data. But for some data, self-adaptive spectral clustering becomes helpless. We construct the circular data, composed of three ring data, and the initial data shape is as shown in Figure 2 (a).

We first use the traditional spectral clustering algorithm to cluster them, and get the experiment result of Figure 2 (b). Clearly, if traditional spectral clustering processes circular data, we can not get the desired results, because traditional spectral clustering is defined based on Euclidean distance. It is natural that the red part of the left, the green part of the center and the blue part of the left will be clustered together, because this measurement is difficult to determine data characteristics of between circular data, i.e., the true degree of similarity between the data.

Next, we use SSC-SNN to cluster the above data and the clustering result is shown in Figure 2 (c). It can be seen from the experimental results, SSC-SNN still has flaws, still not able to process circular data well. The central part of the circular data can be classified well, but for ring data, SNN can keep neighbor information of the data, but cannot accurately measure the degree of correlation between data points. When there are three categories, two outer ring data cycle has to be divided into two parts, resulting in unexpected division. So the definition of the distance is still the urgent problem of SSC-SNN. The circular data has a total of 600 data and the correct rate statistics of two spectral clustering algorithms is as shown in Table 1.



(a) The initial shape (b) Clustering result of NJW (c) Clustering result of SSC-SNN

Figure 2. Circular data

Table 1. The circular data processing results

Spectral clustering algorithms	Correct	Wrong	Total	Correct rate
NJW	325	275	600	54.17%
SSC-SNN	400	200	600	66.67%

For the above problem, we can think of using the geodesic distance to process. We have already discussed the concept of geodesic distance, and the use of geodesic distance instead of traditional Euclidean distance can process circular, half-moon shaped and specific manifold data, more accurately measure the distance between two data points, thereby enhancing the clustering effect of such data. Next, we use SSC-GD&SNN to process circular data and the obtained result is shown in Figure 3.

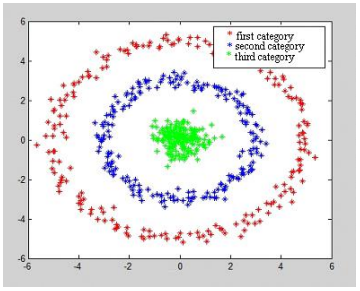


Figure 3. Clustering result of SSC- GD&SNN

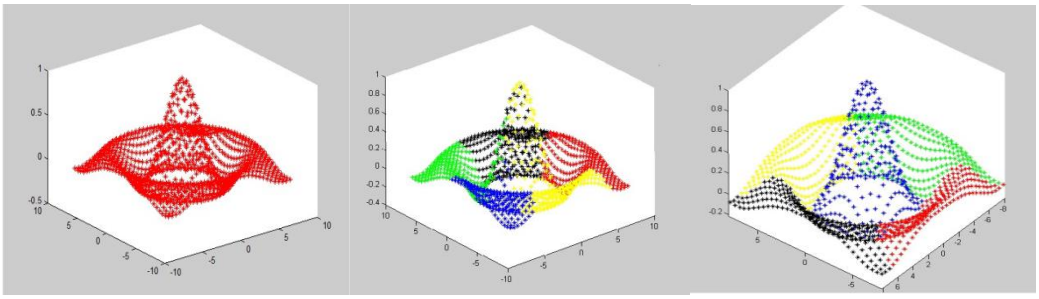
We can see from the figure, after the application of geodesic distance instead of distance measure in the similarity matrix, the spectral clustering effect has been improved, three ring data are accurately clustered and the impact of human factors on the σ value is eliminated.

5.4 Hat-shaped data

Next, we use hat-shaped data to further verify SSC- GD&SNN.

The shape of the initial data is shown as Figure 4 (a). Such data is much like a fashion hat, but with the four hat brims and one crown. Our clustering work clusters them into five categories. Ideally, the four hat brims should be classified into one category and one crown into one category. The clustering effect of traditional spectral clustering and SSN are not good. The clustering result is shown in Figure 4 (b).

Due to the inaccuracy of distance measure, although the algorithm clusters them into five categories, the data on the top is not well characterized, mixed with the following data points, so the clustering effect is not ideal. Here we use SSC-GD&SNN to cluster and the observed clustering result is shown in Figure 4 (c). Experimental results show that after the introduction of geodesic distance, GD&SSC-SNN can better process circular and manifold data.



(a) The initial shape (b) Clustering result of SSC-SNN (c) Clustering result of SSC-GD&SNN

Figure 4. Hat-shaped data

6. Conclusions

Clustering algorithm is an important branch in machine learning, and plays a very important role. The purpose is to cluster high similar data together. Because the clustering algorithm does not require category label, it can take full use of its superior characteristics when processing many issues. However, current clustering algorithm performance restricts the development and applications of clustering algorithms. In recent years, scholars have done sufficient efforts on the clustering algorithm, and the study on clustering algorithms is becoming increasingly hot.

Subspace clustering is a method for processing subspace. As a subspace clustering algorithm, spectral clustering is a relatively new clustering algorithm, also very competitive. Spectral clustering is no longer confined to the circular data processing, and the clustering complexity only relates with the number of data points, not the number of dimensions. Spectral clustering can map highly non-linear data points to the linear subspace, thus becoming a simple clustering problem of linear subspace. Spectral clustering is simple to achieve, but needs a solid theoretical foundation.

This paper analyzed and discussed the traditional spectral clustering, SSC-SNN and SSC-GD&SNN. For reliance problem of traditional spectral clustering on Gaussian kernel function, use shared nearest neighbors of two points to access the implicit information in the cluster structure, and use this information for similarity calculation, which is self-adaptive spectral clustering based on shared nearest neighbors. For the deficiency of manifold data processing of SSC-SNN, the concept of geodesic distance was introduced into SSC-SNN, and proposed SSC-GD&SNN.

Although this paper has some improvement result for spectral clustering, but for spectral clustering there is still a lot of work to be done. The main researches are about the practical application, improvement of the clustering accuracy and spectral clustering algorithm.

References

[1] J. Y. Li, J. G. Zhou and J. H. Guan, "A survey of clustering algorithms based on spectra of graphs", CAAI Transactions on Intelligent Systems, vol.6, no.5, (**2011**), pp. 405-414.
[2] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering", Proceedings Neural Information Processing Systems (NIPS 2004), (**2004**).
[3] A.Y. Ng, M. I. Jordan and Y. Weiss. "On spectral clustering: analysis and an algorithm". Proceedings Neural Information Processing Systems (NIPS 2001), (**2001**).
[4] U. V. Luxburg, "A tutorial on spectral clustering, statistics and computing", vol.17, no. 4, (**2007**), pp.395-416.
[5] Kursun, O. I. Univ, "Spectral clustering with reverse soft k-nearest neighbor density estimation", Neural Networks, (**2010**), pp.1-8.
[6] A. Hamzaoui, A. Joly, N. Boujemaa, "Multi-source shared nearest neighbours for multi-modal image clustering" , vol. 51, no.2, (**2011**),pp 479-503.
[7] Z. M. Pan, Y. L. Chen, "Research on shared nearest neighbor clustering for large dataset", Journal of Chinese Computer Systems", vol.35, no.1, (**2011**), pp.51~55.
[8] X. Y. Liu, J. W. Li, H. Yu, "Adaptive spectral clustering based on shared nearest neighbors", Journal of Chinese Computer Systems, vol. 32, no.9, (**2011**), pp.1876-1880.
[9] W. T. Wu, Y. X. Li, B. H. Wei, S. L. Zheng, "Incremental ISOMAP method based on locally estimated geodesic distance", Journal of Shanghai Jiao Tong University, vol. 47, no. 7, (**2013**), pp. 1082-1087.
[10] A. Murari，P. Boutot，J. Vega, "Clustering based on the geodesic distance on Gaussian manifolds for the automatic classification of disruptions", Nuclear Fusion, vol.53, no.3, (**2013**), pp. 33006-33014(9).
[11] A. Criminisi, T. Sharp, C. Rother, "Geodesic Image and Video Editing", ACM Transactions on Graphics(TOG), vol.29, no.5, (**2010**), pp.1-5.
[12] P. J. Toivanen, "New geodesic distance transforms for gray-scale images", Pattern Recognition Letters, vol.17, no.5, (**1996**), pp.437-450.
[13] G. h. Wen, L. J. Jiang, J. Wen, "Using locally estimated geodesic distance to optimize neighborhood graph for isometric data embedding", Pattern Recognition, vol. 41, no. 7, (**2008**), pp. 2226–2236.

Authors

Chunmiao Yuan, born in September, 1975, Tianjin city, P.R. China
Current position, grades: Associate professor of Tianjin polytechnic university of China
University studies: M.Eng. and Ph.D. of Computer Science from Xi'an Electronic and Engineering University and Tianjin University in China
Scientific interest: Computer network and Data mining
Publications <number or main>: more than 10 papers published in various journals and academic conferences
Experience: teaching experience of 11 years

Kaixiang Fan, born in November, 1992, Zhumadian City, P.R.China

Current position, grades: Graduate students of Tianjin Polytechnic University
University studies: B.Eng. of Computer Software from Tianjin Polytechnic University in China
Scientific interest: Computer application technology
Publications <number or main>: null
Experience: Pursuing a master's degree of one and a half years

Xuemei Sun, born in October, 1971, Fengning County, Chengde city, P.R. China

Current position, grades: Associate professor of Tianjin polytechnic university of China
University studies: M.Eng. and Ph.D. of Computer Science from Tianjin University in China
Scientific interest: Computer application technology
Publications <number or main>: more than 20 papers published in various journals
Experience: teaching experience of 14 years, 5 scientific research projects