Density Peak clustering algorithm based on t-SNE Optimization

Yanbing Wu1, Xiwei Feng1*, Yuzi Dou1, Rui Zhu1, Lei Ma1 and Tianzhu Gao1

1School of Computer and Communication Engineering, Liaoning Shihua University, Fushun City, Liaoning Province, 113006, China
feng.xiwei@163.com

Abstract: Aiming at the defect that the fast peak search clustering does not adapt to high-dimensional data sets, a T-DPC optimization algorithm is proposed. The algorithm is based on the t-SNE dimensionality reduction method, and also optimizes the calculation method of Gaussian kernel function, using a uniform metric when solving the density. Finally, the T-DPC algorithm and the DPC algorithm are compared in the artificial data set and the UCI standard data set, respectively. The experimental results show that the T-DPC algorithm not only adapts to the high dimensional dataset, but also improves the efficiency of the DPC algorithm.

1. Introduction
In 2014, Alex Rodriguez [11] published an article called < Clustering by fast search and find of density peaks > on < Science >. It can solve any shape of clustering, and does not need strong dependent parameters. Win the unanimous praise of all of us. But then people found a lot of problems. For example, he does not work well with high-dimensional datasets and needs to select appropriate functions when calculating density $\rho$. In order to solve these problems, a new T-DPC algorithm is proposed in this paper, which can solve the problem of poor performance in dealing with high-dimensional datasets.

2. DPC algorithm
DPC is a distance measure of any two points in the data set and is not affected by high-dimensional data. The cluster center of the algorithm has the following characteristics: (1) surrounded by points with low local density, and (2) the distance between the centers of each cluster is relatively far. The density peak clustering algorithm introduces two quantities, one is the local density $\rho$ and the other is the distance. For a random sample i of the data set, the formula of the local density is shown in (1). In addition, Alex Rodriguez proposed another way to calculate the local density, which is to use the Gaussian kernel function to calculate the density, as shown in (2).

$$\rho_i = \sum_{j \neq i} \chi(d_{ij} - d_c)$$

$$\rho_i = \sum_{j \neq i} e^{-\frac{(d_{ij})^2}{2\sigma^2}}$$

Distance $\delta$ is the closest distance between point i and its higher density point, as follows:

$$\delta_i = \min_{j \neq i, \rho_j > \rho_i} (d_{ij})$$
3. Improved T-DPC algorithm description

3.1. Description based on t-SNE algorithm

In this paper, the t-SNE dimensionality reduction method based on radiation transformation is used to standardize the data. First of all, the t-SNE algorithm is a nonlinear dimensionality reduction method for high-dimensional data. The core of the SNE algorithm is to combine data into the probability distribution by means of affine transformation.

SNE calculation process: First, the conditional probability will be used instead of the Euclidean distance to express the similarity between objects. For example, given a high-dimensional data set \( X = (x_1, x_2, \ldots, x_n) \). The calculation formula (4) of the conditional probability is as follows. The low-dimensional data points correspond to the high-dimensional data points, and the conditional probability distribution calculation

\[
p_{ij} = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)}{\sum_{i \neq k} \exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma^2}\right)}
\]

(4)

\[
q_{ij} = \frac{\exp\left(-\frac{\|y_i - y_j\|^2}{2\sigma^2}\right)}{\sum_{i \neq k} \exp\left(-\frac{\|y_i - y_k\|^2}{2\sigma^2}\right)}
\]

(5)

Among them, the above two formulas are representative similarities, so both \( p_{ij} \) and \( q_{ij} \) are zero. Then the distance KL divergence between the two distributions is optimized, and his objective function is as follows:

\[
C = KL(P \parallel Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}
\]

(6)

The core principle of t-SNE is to transform a method for finding KL divergence by replacing the conditional probability distribution by joint probability distribution. Its optimized objective function becomes as follows:

\[
C = KL(P \parallel Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}
\]

(7)

The formulas for \( p_{ij} \) and \( q_{ij} \) shown in the above equation have also changed, and their calculation formulas become:

\[
p_{ij} = \frac{p_{ij} + p_{ji}}{2}
\]

(8)

\[
q_i = \frac{1}{\sum_{x \neq i} \left(1 + \|x - y_i\|^2\right)^{-1}}
\]

(9)

Finally, the gradient calculation is performed. The complete gradient formula after t-SNE optimization is as follows:

\[
\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij}) \left(1 + \|y_i - y_j\|^2\right)^{-\frac{3}{2}}
\]

(10)

Because the above formula is easy to fall into the local optimal solution during the optimization process, a large momentum parameter needs to be added in the process of gradient descent. It is added as follows:

\[
Y^{(i)} = Y^{(i-1)} + \eta \frac{\partial C}{\partial Y} + \alpha(t)(Y^{(i-1)} - Y^{(i-2)})
\]

(11)

\( Y^{(i)} \) refers to the solution of iterations \( t \) times, \( \eta \) represents the learning rate, and \( \alpha(t) \) represents the iteration of \( t \) momentum. At this point, the standardization process of the data is completed, and the standardized data retains the characteristics of the original data set to the greatest extent and is adapted to the DPC algorithm.
Algorithm 1:
Input: High dimensional data set: X = x1, ..., xn
Output: The target result is low dimensional data representation y^T = y1, ..., yn
1. Calculate the conditional probability according to formula (4)\( P_i^j \)
2. Calculate \( q^j \) according to formula (8)
3. Iterate, from t=1 to T, do the following:
   3.1 Calculate \( q^j \) in low dimensions according to formula (9)
   3.2 Calculate the gradient according to formula (10)
   3.3 Find new low-dimensional data according to formula (11)
4. End.

3.2. Unified guidelines for density calculation:
According to the mathematical properties of the Gaussian kernel function [9], when the data point distance is higher than \( 3d_i / \sqrt{2} \), the contribution value of the sample point will drop rapidly to 0, so it is not necessary to calculate more than \( 3d_i / \sqrt{2} \) sample points when calculating the Gaussian kernel function. Among them, this can improve the performance of the algorithm, overcome the shortcomings of high complexity, and solve the problem of inconsistent density calculation.

Improved description: (1) Define a list \( Z \). (2) After calculating \( dc \), save the distance from any point \( j \) to \( i \) less than the distance to the list \( Z \). (3) New density calculation method:
\[
\rho_i = \sum_{j \neq i} e^{-\frac{d(i, j)^2}{2dc^2}}
\]

(12)

3.3. T-DPC algorithm flow
1. Use algorithm 1 for data standardization.2. Read the point, calculate the distance between the point and the point, find \( dc \). 3. The local Gaussian kernel function (12) can be used to find the local density \( \rho \). 4. Using the formula (3) to find \( \delta \). 5. Drawing a decision graph. 6. Manually select cluster centers. 7. Distribution point. 8. Calculate noise points and complete clustering.

4. Experimental simulation and analysis

4.1. Experimental simulation environment
All experiments were done with the software pycharm, using the language python3. The test was run on a laptop. (i5-3337U CPU, 1.80GHz memory 8G Win7)

4.2. Experimental data
In order to verify the authenticity and accuracy of the T-DPC algorithm, three UCI data sets [14] and five standard [15] UCI data sets were selected for testing. The cluster center is manually selected, The experimental data are shown in Table 1.

| Dataset    | record | attributes | clusters |
|------------|--------|------------|----------|
| aggregation| 788    | 2          | 7        |
| D31        | 3100   | 2          | 31       |
| R15        | 600    | 2          | 15       |
| PID        | 768    | 8          | 2        |
| wine       | 178    | 13         | 3        |
| iris       | 150    | 4          | 3        |
| waveform   | 5000   | 21         | 3        |
| seed       | 210    | 7          | 3        |
4.3. Evaluation index of clustering results

In order to verify the authenticity of the T-DPC algorithm clustering effect, we use two effective indicators to evaluate, namely the F-measure indicator and the NMI indicator. Then compare the time efficiency of T-DPC algorithm and DPC algorithm, and finally evaluate the two according to the test results.

4.3.1. F-measure indicator

The F-measure indicator is a commonly used evaluation standard in information retrieval. The reconciliation and re-weighted average are determined by precision and recall. Let \( P_i \) be the known cluster manually labeled, and \( C_i \) be the cluster formed by the end of the cluster.

\[
\begin{align*}
(1) \quad \text{Accuracy rate} & \quad P(C_i \cap P_i) = \frac{|P_i \cap C_i|}{|C_i|} \\
(2) \quad \text{Recall rate} & \quad R(P_i, C_i) = \frac{|P_i \cap C_i|}{|P_i|} \\
(3) \quad \text{F-measure} & \quad F(P_i \cap C_i) = \frac{2 \cdot P(C_i \cap P_i) \cdot R(P_i, C_i)}{P(C_i \cap P_i) + R(P_i, C_i)}
\end{align*}
\]

Table 2 Comparison of F evaluation indicators of clustering results

| Dataset  | DPC  | T-DPC |
|----------|------|-------|
| aggregation | 0.999 | 0.998 |
| D31      | 0.968 | 0.970 |
| R15      | 0.997 | 0.997 |
| PID      | 0.518 | 0.651 |
| wine     | 0.677 | 0.729 |
| iris     | 0.788 | 0.897 |
| Waveform | 0.358 | 0.412 |
| seed     | 0.317 | 0.298 |

4.3.2. NMI evaluation index

Standard mutual information is a concept in the description of information theory, which is used to measure the degree of agreement between the two data distributions. Suppose \( X \) and \( Y \) are the allocations for \( N \) samples.

\[
\begin{align*}
H(X) & = \sum_{i=1}^{P} P(i) \log(P(i)) \\
H(Y) & = \sum_{j=1}^{P'} P'(j) \log(P'(j))
\end{align*}
\]

among them, \( P(i) = |X|/N, P'(j) = |Y|/N \). The mutual information (MI) between \( X \) and \( Y \) is defined as:

\[
MI(X,Y) = \sum_{i=1}^{P} \sum_{j=1}^{P'} P(i,j) \log \left( \frac{p(i,j)}{P(i)P'(j)} \right)
\]

among them, \( P(i,j) = |X \cap Y|/N \). The standardized mutual information (NMI) [16] is:

\[
NMI(X,Y) = \frac{MI(X,Y)}{\sqrt{H(X)H(Y)}}
\]

Table 3 Comparison of NMI evaluation indicators

| Dataset  | DPC  | T-DPC |
|----------|------|-------|
| aggregation | 0.996 | 0.997 |
4.3.3. Algorithm operation efficiency analysis:
In order to reflect the accuracy of the experiment, the average of 20 run times was taken as the final run time.

Table 4 Run time comparison of the two algorithms (Unit: S)

| Dataset    | DPC  | T-DPC |
|------------|------|-------|
| aggregation| 22.2 | 17.2  |
| D31        | 310.2| 194.1 |
| R15        | 14.2 | 11.2  |
| PID        | 23.5 | 20.5  |
| wine       | 3.5  | 3.5   |
| iris       | 3.1  | 3.1   |
| Waveform   | 846.7| 512.2 |
| seed       | 3.7  | 3.5   |

4.4. Experiment analysis
It can be seen from Table 2 that the F-measure index of the DPC algorithm does not change much more than the F-measure index of the T-DPC algorithm for the three-person synthetic dataset, because it does not need to undergo dimensional reduction of t-SNE. So there is basically no change in accuracy. For the high-dimensional UCI dataset, the change is relatively larger. After the t-SNE dimension reduction, the four data sets increased by 7.0%, 5.2%, 1.1%, and 5.4%, respectively, before the dimension reduction.

Obviously, the data after t-SNE dimensionality reduction is better.

Table 3 reflects the degree of agreement between the clustered dataset and the original dataset. The PID dataset, wine dataset, and waveform dataset are 4.9% and 3.9% lower than the NMI in the DPC algorithm under the T-DPC algorithm, 3.6%, which indicates that the four data sets are less consistent with the original data set. In the remaining two synthetic sets and the iris data set, respectively, the aggregation data set and the D31 data set increased by 0.1% and 0.2%, respectively, and the iris data set increased by 7.3%. But there is no change in the R15 data set. The combination can show that the consistency of the data set will decrease in the case of higher dimensions, and the better in the lower dataset.

Table 4 reflects the time efficiency comparison between the T-DPC algorithm and the DPC algorithm. Due to the new Gaussian kernel function, the overall runtime has been improved. In the artificial data set D31, aggregation and R15, the running time of the T-DPC algorithm is increased by 116.1s, 5s and 3s, respectively, than the running time of the DPC algorithm. At the same time, 334.5s and 3s have been improved in the standard set waveform and PID respectively. However, it may be because the amount of data is too small, there is no change in the wine dataset and the iris dataset. As can be seen from the above experimental results, the T-DPC algorithm is more suitable for large data sets than between small data sets and large data sets. Because this can increase more efficiency and is more suitable for real-world data.

5. Conclusion
The T-DPC algorithm proposed in this paper has no obvious change in the effect of DPC algorithm on low-dimensional data, but it has a better improvement on high-dimensional data sets. The T-DPC
algorithm not only unifies the density calculation formula, but also improves the computational efficiency under large data sets. However, this paper continues to select the dataset length of 1% - 2% as dc, and the NMI index of high-dimensional data is not good, the degree of coincidence has decreased, although the efficiency is improved, the degree is still far from enough, then you can consider putting it in a distributed environment for calculations. In short, how to adaptively choose dc and continue to improve efficiency is the next issue.

References
[1] Frey, B.J., Dueck, D. (2007) Clustering by passing messages between data points[J]. Science. 5: 972-976.
[2] Xu, R.C., Wunsch, D. (2005) Survey of clustering algorithms[J]. IEEE Transactions on Networks. 16:645-678.
[3] Sambasivam, S., Theodosopoulos, N. (2006) Advanced data clustering methods of mining Web documents[J]. Issues in Informing Science and Information Technology. 3:563-579.
[4] Wu, Y., Wang, T., Li, J.D. (2016) Parameter selection algorithm based on density partitioning clustering process[J]. Control and Decision. 1: 21-29.
[5] Xia, N.X., Su, Y.D., Tan, X. (2010) An efficient K-medoids algorithm [J]. Application Research of Computers. 12: 4571-4519.
[6] Guha, S., Rastogi, R., Shim, K. (2008) CURE: A new efficient clustering algorithm for large databases[C]. Proc of the ACM SIGMOD Int Conf on Management of Data. New York: ACM Press. 1998: 73-84.
[7] Karypis, G., Han, E.H., Kumar, V. (1999) CHAMELEON: A hierarchical clustering algorithm using dynamic Modeling[J]. IEEE Computer. 8: 68-75.
[8] Agrawal, R., Gehrke, J.E., Gunopoulos, D. (1998) ACM sigmod Int Conf on Management of Data. ACM Press, Seattle.
[9] Ester, B.M., Kriegel, H.P., Sander, J. (1996) eACM: A density based algorithm for discovering clusters in large spatial databases[C]. Proc of Int Conf on Knowledge Discovery and Data Mining. Protland. 165-178.
[10] Rodriguez, A., Laio, A. (2014) Clustering by fast search and find density peaks[J]. Science. 344:1492-1496.
[11] Xie, X.N., Gao, S.P., Wu, H.H. (2018) Improved density peak clustering algorithm combining K-nearest neighbors[J]. Computer Engineering and Applications. 2018 (7):36—43.
[12] Gao, S.Y., Zhou, X.F., Li, Sh. (2017) Peak clustering algorithm based on density ratio [J]. Computer Engineering and Applications. 16:10—17.
[13] Gan, W.Y., Liu, Ch. (2017) An improved clustering algorithm for search density peaks [J]. Journal of Intelligent Systems. 2: 229-235.
[14] Chang, H., Yeung, D.Y. (2008) Robust path-based spectral clustering. Pattern Recog. 4: 191-203.
[15] Lichman M. (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml. Irvine:University of California.