High-Dimensional Shared Nearest Neighbor Clustering Algorithm*

Jian Yin[1], Xianli Fan[1], Yiqun Chen[1,2], and Jiangtao Ren[1]

[1] Zhongshan University, Guangdong, P.R. China, 510275
[2] Guangdong Institute of Education, Guangdong, P.R. China
issjyin@zsu.edu.cn

Abstract. Clustering results often critically depend on density and similarity, and its complexity often changes along with the augment of sample dimensionality. In this paper, we refer to classical shared nearest neighbor clustering algorithm (SNN), and provide a high-dimensional shared nearest neighbor clustering algorithm (DSNN). This DSNN is evaluated using a freeway traffic data set, and experiment results show that DSNN settles many disadvantages in SNN algorithm, such as outliers, statistic, core points, computation complexity etc, also attains better clustering results on multi-dimensional data set than SNN algorithm.

1 Introduction

Cluster analysis mainly originates from many research domains, including data mining, statistics, biology, machine learning and so on. Most data mining researchers focus on the scalability and effectivity on clustering algorithm. Except that, researchers also try to find out methods for cluster data with complex figures and types, high-dimension cluster analysis technology, and methods for hybrid numerical and systematic data.

Clustering is one of the challenging research domains in data mining and is widely applied on all kinds of application. But different application has its special requirement, such as flexibility, domain knowledge minimum to decide input parameter, ability to deal with outliers, not sensitive to input record sequence, high dimensionality, usability and so on.

In this paper, we describe a high-dimensional shared nearest neighbor clustering (DSNN) algorithm, and evaluate it on multi-dimensional spatio-temporal data set. The rest of the paper is organized as follows: In section 2, we will introduce the challenge of high dimensional data and classical shared nearest neighbor clustering algorithm, and analysis its disadvantages. Then by introducing new

* This work is supported by the National Natural Science Foundation of China (60205007) , Natural Science Foundation of Guangdong Province (031558,04300462), Research Foundation of National Science and Technology Plan Project (2004BA721A02), Research Foundation of Science and Technology Plan Project in Guangdong Province (2003C50118) and Research Foundation of Science and Technology Plan Project in Guangzhou City(2002Z3-E0017).

L. Wang and Y. Jin (Eds.): FSKD 2005, LNAI 3614, pp. 494–502, 2005.

relevant similarity and density definitions, we bring out the DSNN algorithm in Section 3. Finally, in Section 4, we evaluated DSNN on American freeway traffic dataset. The experiment results prove that DSNN algorithm has better performance than SNN algorithm on correlative measures, such as in-cluster similarity, between-cluster dissimilarity, spatio-temporal complexity and so on.

2 Related Works

The clustering algorithm is mainly applied on spatio-temporal dataset, especially for traffic dataset.

2.1 Challenge from High Dimensionality

Observe the data samples showed in table 1 and table 2, their Euclidean distance is five. A attribute with a "0" value means it is a useless data(the experiment got no test data) or it is a free traffic situation. Attributes with a value from "1"to "4" stands for a more and more busy traffic situation. That is attribute with a value "4" means the most busy traffic situation. But it is not hard to judge that there is higher similarity between Sample Three and Sample Four than between Sample One and Sample Two, because there are eight shared attributes between the former pair, and none between the latter pair. Generally speaking, if there are redundant "0" attributes in samples, it is sparse; we often define the similarity by ignoring those useless attributes between two samples.

Other measures, such as cosine measure, Jaccard modulus etc, can be used to deal with this problem. If we compute the cosine measures between Sample One and Sample Two, Sample Three and Sample Four respectively, the former is zero and the latter is 0.8286.

Even if cosine and Jaccard measure can provide relevant similarity information about whether there are intact attributes or not, they can't handle high dimensionality very well. By computing cosine measure between high dimensional data, we can find that, if only observing 15-20 percents of attributes of Sample A, those data, which is very similar to Sample A, may belong to other clusters without Sample A, as described in [1]. It doesn't derive from the defective similarity measure, but from the un-trusted direct similarity of high dimensional data, since the similarity between any sample pair is all very low.

Another very important problem derived from similarity measures is *the triangle inequality* doesn't hold. Consider the following example in table 3: Sample

Table 1. data sample 1∼2

Point	Att1	Att2	Att3	Att4	Att5	Att6	Att7	Att8	Att9	Att10	Att11	Att12
1	4	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	3

Table 2. data sample 3∼4

Point	Att1	Att2	Att3	Att4	Att5	Att6	Att7	Att8	Att9	Att10	Att11	Att12
3	4	2	0	1	4	3	0	2	3	4	1	0
4	0	2	0	1	4	3	0	2	3	4	1	3

Table 3. data sample 5∼7

Point	Att1	Att2	Att3	Att4	Att5	Att6	Att7	Att8	Att9	Att10	Att11	Att12
5	3	4	2	4	1	4	3	3	2	2	1	4
6	0	0	2	4	1	4	2	1	4	3	2	3
7	4	2	3	3	0	2	2	1	4	3	3	1

5 is close to sample 6, sample 6 is close to sample 7, but sample 5 and sample 7 have a similarity of 0. The similarities come from different sets of attributes.

2.2 SNN Algorithm

Some researchers provide a *shared k nearest neighbors-based* algorithm (SNN algorithm) [2,3]. Its procedure can be described as: firstly compute the similarity matrix, then sparsify the similarity matrix by keeping only the k most similar neighbors, finally construct a correlative shared nearest neighbor graph. Subsequently, it will find out the SNN density of each point to specify the core points, and form clusters with the core points. Except that, it also discards all noise points and assigns all non-noise, non-core points to clusters.

With the synthesis analysis on this algorithm, it is not difficult to find that there are many disadvantages as follows:

- There are no enough processes about outliers, which results in redundant pointless computations.
  Apparently, there are only a few calculations about outliers in SNN, until finishing constructing SNN graph with all samples. Moreover, you have to begin judging whether it is an outlier until computing link strengths between all points. By analyzing its algorithm, we can find that the complexity of computing similarity matrix and constructing SNN graph is $0(M^2)$, which lead to a high complexity of this algorithm.
- Scientist definitions of thresholds for core points, outliers and filter those link strengths are not clearly provided.
  With the statistics to the sample data, we can find out the thresholds to define core points and outliers. However, there is high spatio-temporal measure in statistics, which undoubtedly add complexity in this total algorithm.
- The procedure of defining core points is not good enough.
  It is not very exact to define core points directly by thresholds, which lead to a vital problem: defined core points may belong to identical cluster. Thus,

if clustering with these core points, it is hard to avoid partitioning born clusters. In other word, in order that the latter clustering can be as accurate as possible, the core points had better to be as disperse as possible.

3 High Dimensional Shared Nearest Neighbor Clustering Algorithm (DSNN)

In order to overcome the disadvantages of the classical SNN algorithm, we bring out the refined DSNN clustering algorithm.

3.1 "Elementary Deletion for Outliers"

Given a distance measure on a feature space, there are many different definitions of distance-based outliers. The simplest type of algorithm based on nested loops in conjunction with randomization and a pruning rule gives state-of-the-art performance [4]. The algorithm not only computes the distance between any two examples using, for example, Euclidean distance for continuous features and Hamming distance for discrete features, but also can be any monotonically decreasing function of the nearest neighbor distances such as the distance to the k^{th} nearest neighbor, or the average distance to the k neighbors.

The main idea in our nested loop algorithm is that for each example in a data set, we keep track of the closest neighbors found so far. When an example's closest neighbors achieve a score lower than the cutoff, we remove the example because it can no longer be an outlier. As we process more examples, the algorithm finds more extreme outliers and the cutoff increases along with pruning efficiency.

What needs to attend is that samples can be put into random order in linear time and constant main memory with a disk-based algorithm. One repeatedly shuffles the data set into random piles and then concatenates them in random order.

As far as the algorithm complexity, because of the nested loops, it could require $0(M^2)$ distance computations and $0(M^2/\text{blocksize})$ data accesses.

3.2 Core Points and Outliers

SNN Algorithm can define the core points and outliers with the help of correlative threshold by users. Those points with a higher link strength than the threshold is defined as core points, vice versa. This method with threshold often have a inferior efficiency, since users are required to have a deep understand on spatio-temporal data set, and also require to take abundant correlative experiments for thresholds setting. There apparently exists the possibility of threshold data error, and it often reflects on spatio-temporal data set, which reduces clustering precision and maneuverability by a long way.

Furthermore, there exists another important problem: All or some of these core points may possibly belong to identical cluster.

DSNN Algorithm has a new idea for defining core points. Firstly, we can confirm a candidate set of core points, which must make certain that all possible core points are included in it. Then, we can choose a core point randomly, and define a candidate point as a new core point by distance measure, whose distance is more than distance threshold. After defining new core points as a seed, we can continuously define other candidate core points in that candidate set until judging all candidate points.

Aimed at outliers, the method in DSNN is based on SNN graph and candidate core point set. If some sample has fewer SNN of other samples, it is defined as an outlier.

3.3 DSNN Algorithm

After discussing the deficiencies of the classical SNN algorithm and providing correlative solutions, we have succeeded in discovering the following DSNN algorithm.

Algorithm 1 DSNN clustering algorithm

Input: spatio-temporal data set
Output: every individual cluster

1.Transfer an algorithm to find distance-based outliers as above, in order to achieve better precision with refined sample set.
2.Based on this sample set, construct the similarity matrix.
3.Sparsify the similarity matrix using k-nn sparsification.
4.Construct the shared nearest neighbor graph from k-nn sparsified similarity matrix.
5.For every point in the graph, calculate the total strength of links coming out of the point.
6.Define noise points by choosing the points that have low total link strength and remove them, and gain a refined dataset again.
7.Based the new dataset, re-implement Step 2 to Step 5 once.
8.Define the candidate set of core points by choosing the points that have high total link strength.
9.For this candidate set, to do some refining as follows: saving those whose relative distance is very high as defined core points, and deleting the remainder.
10.Based on these core points and new SNN graph, to form clusters, where every point in a cluster is either a core point or is connected to a core point.

4 Experiments

This traffic dataset is from American freeway network, which is recorded as a matrix, where every row indicates each checkpoint and column to time segment. For every data set $D_{i,j}$, it records traffic state in the j^{th} time segment within the i^{th} checkpoint. Moreover, it has been preprocessed, and records traffic states with 0 to 4, where 0 indicates free, 4 indicates busy.

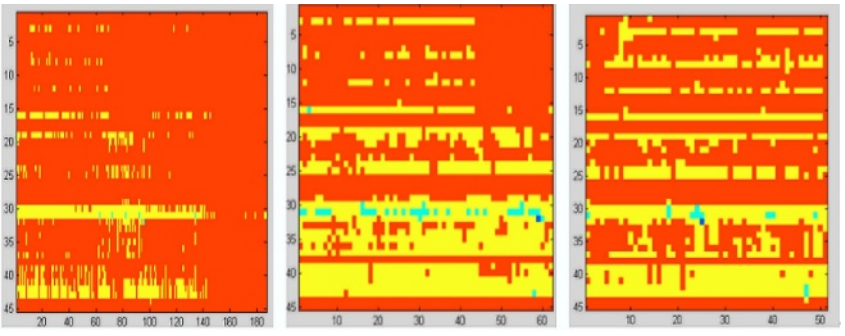

Fig. 1. the same Cluster Results with SNN and DSNN

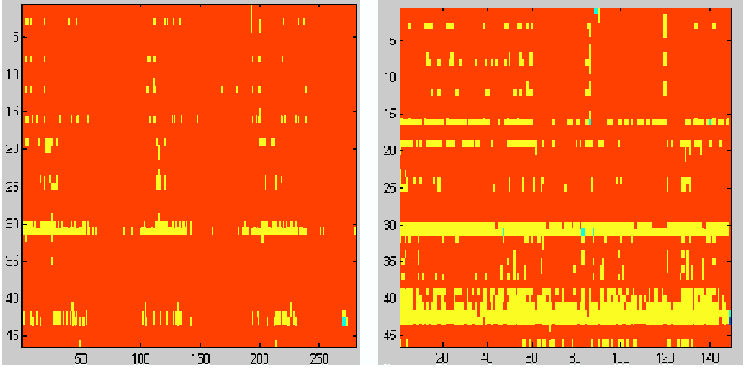

Fig. 2. the Cluster with SNN Algorithm (Left) and DSNN Algorithm (Right)

In this section, we evaluate our DSNN by applying SNN and DSNN on this data set and compare their results on different experiment environment. We will try different attempts with this experiment dataset, such as different amount of data set, and compare their experiment results.

We randomly chose 300 samples from the set, and the clustering result with DSNN method was same as one with SNN method in same time: As the three figure showed in Figure 1, they respectively show a cluster. The number of abscissa stands for the experiment day number, while the ordinate stand for the time period of a day. As we can see from the three figures, the amount and sample data was also same respectively, which shows that they can be used to handle small data set with same precision. We can get the traffic situation from the tint area in the figure: tint area show a free traffic situation, others is a busy traffic situation. For example, the first left figure shows that in the time periods 16, from 29 to 32, from 41 to 44(those tint area), the traffic is in a normal situation, it also shows that the traffic situation has a high similar in these time period.

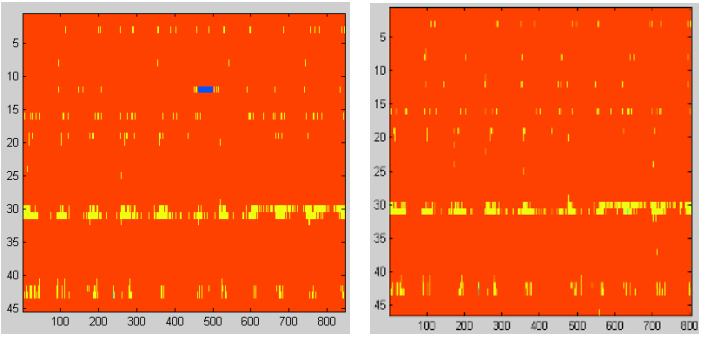

Fig. 3. the Supreme cluster with SNN (left) and DSNN (right) method

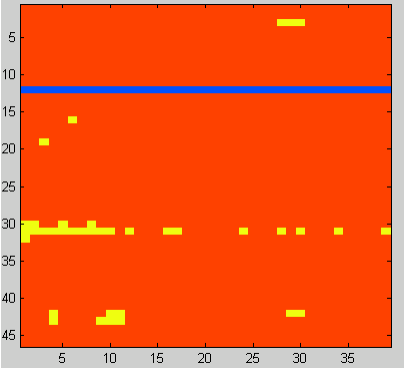

Fig. 4. The partial outliers with DSNN method

What is showed in Figure 2 is the most primary cluster with classical SNN algorithm and DSNN method respectively and about 600 samples. The meaning of abscissa and ordinate is the same with Figure 1. The left figure and the right figure stand for a cluster result respectively. By computing their cluster-in similarities respectively, we can gain the clusters with high cluster-in similarities if adopting the DSNN method. In other word, DSNN clustering algorithm can help with obtaining more accurate clusters on big data sets.

Perhaps there is an incident in the clustering result with six hundred samples, so we decided to cluster with more samples continuously in order to evaluate the performance of the DSNN algorithm.

Figure 3 shows the supreme cluster graphs with the classical SNN method and the DSNN method respectively. The meaning of abscissa and ordinate is the same with Figure 1. The left figure include the data in the right figure, but each of them stand for a cluster result respectively. Obviously, they are same on the whole, but there are more outliers in the former. Moreover, with careful feedback of algorithm implement, we can find that the difference is primarily included in

the outliers during the course of deleting outliers, as showed in Figure 4. All the above shows that DSNN has a much better adaptability and performance than SNN.

5 Conclusion

In this paper, we firstly analyzed the classical SNN algorithm in the fields of spatio-temporal data cluster analysis. Then we brought forward the high dimensional nearest neighbor clustering algorithm (DSNN) step by step to overcome SNN's shortcomings. This refined algorithm can reduce the spatio-temporal complexity effectively, and refined many performances, such as outliers, core points, clustering results and so on. With the experiments on American freeway traffic dataset, we prove that DSNN can reduce computation effectively, at the same time, it can accurately judge core points and outliers, and gain better clustering performance than SNN algorithm with better clustering methods.

There are still some limits in DSNN. For example, it can't deal with data flow; it must be used to preprocess sample data. And how to expand its application domains, how to improve its validity and how to delete sensitive data better such as outliers, etc, are our future work.

References

1. Clarke, F., Ekeland, I.: Nonlinear oscillations and boundary-value problems for Hamiltonian systems. Arch. Rat. Mech. Anal. **78** (1982) 315–333
2. S. Guha, R. Rastogi, K. Shim. Cure: An efficient clustering algorithm for large databases. In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data(SIGMOD'98).(1998)73-84.
3. Levent Ertoz, Michael Steinbach, Vipin Kumar. Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. In Proceedings of Third SIAM International Conference on Data Mining, San Francisco, CA, USA, May 2003.
4. Levent Ertoz, Michael Steinbach, Vipin Kumar. A New Shared Nearest Neighbor Clustering Algorithm and its Applications. Workshop on Clustering High Dimensional Data and its Applications, Second SIAM International Conference on Data Mining, Arlington, VA, USA, 2002.
5. Stephen D. Bay, Mark Schwabacher. Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule. In Conference on Knowledge Discovery in Data archive Proceedings of the ninth ACM SIGKDD International Conference (KDD). (2003)29-38.
6. E. Eskin, A. Arnold, M. Prerau, L. Portnov, S. Stolfo. A framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. Applications of Data Mining in Computer Security, number 10/320,259, filing date: December 16, Kluwer 2002.
7. A. Strehl, J. Ghosh, R. Mooney. Impact of Similarity Measures on Web-page Clustering. In Proceedings of the 17^{th} National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search, AAAI/MIT Press. (2000)58-64.

8. P.-N. Tan, M. Steinbach, V. Kumar, S. Klooster, C. Potter, and A. Torregrosa. Finding spatio-termporal patterns in earth science data. In KDD Temporal Data Mining Workshop, San Francisco, California, USA, August 2001.
9. M. Steinbach, P.-N. Tan, V. Kumar, S. Klooster, and C. Potter. Temporal data mining for the discovery and analysis of ocean climate indices. In Proceedings of the KDD Temporal Data Mining Workshop, Edmonton, Alberta, Canada, August 2002.
10. S. Shekhar, C. T. Lu, S. Chawla, P. Zhang. Data Mining and Visualization of Twin-Cities Traffic Data. University of Minnesota Academic report, 2001.
11. Vipin Kumar, Michael Steinbach, Pang-Ning Tan. Mining Scientific Data: Discovery of Patterns in the Global Climate System. PAKDD, May 7, 2002.
12. Kitamoto Asanobu. Data mining for Typhoon Image Collection. Journal of Intelligent Information Systems,(2002)Vol.19, No.1.25-41.
13. P.-N. Tan, M. Steinbach, V. Kumar, S. Klooster, C. Potter, and A. Torregrosa. Finding spatio-termporal patterns in earth science data. In KDD Temporal Data Mining Workshop, San Francisco, California, USA, August 2001.