AN EFFICIENT HIGH DIMENSIONAL CLUSTER METHOD AND ITS APPLICATION IN GLOBAL CLIMATE SETS

Ke Li*, Fan Lin, and Kunqing Xie

Department of Intelligence Science, State Key Laboratory of Machine Perception, Peking University, Peking University, Beijing100871, China
*Email:**like@cis.pku.edu.cn, *linfan@cis.pku.edu.cn, kunqing@cis.pku.edu.cn*

ABSTRACT

Because of the development of modern-day satellites and other data acquisition systems, global climate research often involves overwhelming volume and complexity of high dimensional datasets. As a data preprocessing and analysis method, the clustering method is playing a more and more important role in these researches. In this paper, we propose a spatial clustering algorithm that, to some extent, cures the problem of dimensionality in high dimensional clustering. The similarity measure of our algorithm is based on the number of top-k nearest neighbors that two grids share. The neighbors of each grid are computed based on the time series associated with each grid, and computing the nearest neighbor of an object is the most time consuming step. According to Tobler's "First Law of Geography," we add a spatial window constraint upon each grid to restrict the number of grids considered and greatly improve the efficiency of our algorithm. We apply this algorithm to a 100-year global climate dataset and partition the global surface into sub areas under various spatial granularities. Experiments indicate that our spatial clustering algorithm works well.

Keywords: Clustering methods, High dimensionality, Global climate datasets, Spatial data

1 INTRODUCTION

Because of the development of modern-day satellites and other data acquisition systems, the amount of spatial data being collected is increasing exponentially. Global climate research often involves overwhelming volume and complexity of high dimensional datasets. The complexity of the data contained in databases means that it is not possible for humans to completely analyze the data being collected. Data mining techniques have been used to discover unknown information, searching for unexpected results and correlations.

Clustering, according to Han, Kamber, and Tung (2001), is the method that groups similar objects into classes. It is an important component of spatial data mining, which can generalize data into a higher conceptual level and is of great importance in spatial data preprocessing. Spatial clustering is quite useful in many applications: it can be used in the identification of areas of similar land usage in an earth observation database or in merging regions with similar weather patterns, and so on (Han, et al., 2001).

So far, although many clustering methods have been used for spatial data, many of which claim that their methods are spatial clustering methods (Kolatch, 2001), few of them treat the spatial dimensions carefully and find good quality clusters (e.g., clusters of different sizes, shapes, and densities in noisy, high dimensional data). According to Ertoz, Steinbach, and Kumar (2002), this is because most of them use direct similarity (e.g.,

K-means, DBSCAN), and they employ a new definition of similarity based on the shared nearest neighbors (Jarvis, & Patrick, 1973). This new algorithm can find clusters of different sizes, shapes, and densities in noisy, high dimensional data, but its run-time complexity of our algorithm is $O(n^2)$, where n is the number of points if a similarity matrix is constructed.

To reduce the complexity of the algorithm, we employ Tobler's "First Law of Geography" and add a spatial window constraint upon each point to restrict the number of other points to be considered as neighbors. The following experiments show that this optimization greatly improves the efficiency of the algorithm and keeps the quality in an acceptable state.

The rest of the paper is organized as follows. Section 2 presents the related work on spatial clustering. Section 3 introduces Tobler's "First Law of Geography" and its optimization. The algorithm is elaborated in section 4, and the experimental results are shown in Section 5. Section 6 is the conclusion.

2 RELATED WORK

2.1 Shared Nearest Neighbor

It is known that climate data is of high dimension and most of the similarity measures do work well in such a high dimensional datasets. For example, consider the five-dimensional climate data points shown in Table 1. It is not hard to judge that there is higher similarity between P3 and P4 than between P1 and P2 because there are three shared attributes between the former pair and none between the latter pair.

Table 1. Datasets with five attributes

Point	A1	A2	A3	A4	A5
P1	3	0	0	0	0
P2	0	0	0	0	4
P3	3	1	2	3	0
P5	0	1	2	3	4

In high dimensional data sets, the traditional Euclidean notion of density, which is the number of points per unit volume, is meaningless. To see this, consider that as the number of dimensions increases, the volume increases rapidly, and unless the number of points grows exponentially with the number of dimensions, the density tends to 0. Thus, as dimensionality increases, it becomes increasing difficult to use a traditional density based clustering method, such as the one used in DBSCAN, which identifies core points as points in high density regions and noise points as points in low density regions.

An alternative to direct similarity is to define the similarity between a pair of points in terms of their shared nearest neighbors. That is, the similarity between two points is “confirmed” by their common (shared) nearest neighbors. If point P1 is close to point P2 and if they are both close to a set of points S, then we can say that P1 and P2 are close with greater confidence because their similarity is “confirmed” by the points in set S. The shared nearest neighbor approach was first introduced by Jarvis and Patrick (1973). A similar idea was later presented in ROCK (Guha, Rastogi, & Shim, 1999).

In the Jarvis-Patrick scheme, a shared nearest neighbor (SNN) graph is constructed from the similarity matrix as follows. A link is created between a pair of points, p and q, if and only if p and q have each other in their k-nearest neighbor lists. This process is called k-nearest neighbor sparsification. The weights of the links between two points in the SNN graph can either be simply the number of nearest neighbors the two points share or can take the ordering of the nearest neighbors into account, specifically, if p and q are two points.

Then, the strength of the link between p and q, i.e., their similarity, is defined by the following equation:

$$Similarity(i, j) = Size(NN(i) \cap NN(j))$$

In the above equation, $NN(i)$ and $NN(j)$ are, respectively, the nearest neighbor lists of p and q. At this point, clusters can be obtained by removing all edges with weights (similarities) less than a user specified threshold and taking all the connected components as clusters (Jarvis & Patrick, 1973). We will refer to this as Jarvis-Patrick clustering.

2.2 SNN Algorithm

Some researchers provide a shared k nearest neighbors-based algorithm as in Ertoz, et al. (2002). The algorithm uses shared nearest neighbors to redefine the neighbor and density and run a DBSCAN clustering algorithm on those definitions. The structure of the algorithm is as follows:

1. Compute the similarity matrix.
2. Sparsify the similarity matrix by keeping only the k most nearest neighbors. (Construct the KNN graphic)
3. Construct the shared nearest neighbor graph from the KNN graphic. (Construct the SNN graphic)
4. Find the SNN density of each point. Using user specified parameters, Eps, find the number points that have an SNN similarity of Eps or greater to each point. This is the SNN density of the point.
5. Find the core points. Using a user specified parameter, MinPts, find the core points, i.e., all points that have an SNN density greater than MinPts.
6. Form clusters from the core points. If two core points are within a radius, Eps, of each other, then they are placed in the same cluster.
7. Discard all noise points. All non-core points that are not within a radius of Eps of a core point are discarded.
8. Assign all non-noise, non-core points to clusters. We can do this by assigning such points to the nearest core point. (Note that steps 4-8 are DBSCAN.)

We can see that the algorithm consists of two parts: first, construction of an SNN graphic and second, running a DBSCAN clustering on the SNN graphic. As each of the points has no more than k neighbors, the DBSCAN has a basic time complexity of $O(k*n)$ .On the other hand, to find the nearest neighbors of every point, the algorithm has to calculate every other point in the dataset and keep the top-k nearest point as the nearest neighbor. It means that the algorithm has to maintain an $n*n$ distance matrix and generally it will take about $O(n^2)$ time to construct such a matrix. But only an $n*k$ matrix is saved, and most of the computation is wasted considering that n is much larger than k. So what has happen to the wasted computation, or what makes the few top-k nearest neighbors saved useful to construct the SNN graphic in a geospatial environment?

3 TOBLER'S FIRST LAW

Tobler's First Law tells us the answers to the above questions: everything is related to everything else, but near things are more related than distant things (Miller, 2004). Spatial autocorrelation is a measure of the correlation among spatial objects. As a result, we prefer to believe a continuous region with similar non-spatial attributes that we are interested in is generated under the same mechanism. To discover the relationship between the top-k nearest neighbors saved and spatial autocorrelation, we do a test using a box to describe the continuous region and the centralization index to tell how those points in a continuous region are similar to a core point.

Definition 1: ***window*** of a core point.

We define a threshold R and if the distance between a point and the core point is smaller than R, then we say the point is close to the core point. The ***window*** of a core point is the area where every point in it is close to the core point.

Definition 2: ***coverage*** of a core point.

As the top-k nearest neighbors means that they are similar to the core point, those points described as being in the ***window*** of a core point means that they are close to the core point. The ***coverage*** of a core point describes how those close points are similar to the core point.

That is $$\text{coverage} = \frac{(\text{points in the window}) \cap (\text{the saved top - k nearest neighbors})}{(\text{the saved top - k nearest neighbors})}$$

Using the ***coverage*** we can test how the windows contain most of the top-k nearest neighbors. Figure 1 shows the average ***coverage*** of a core point with different ***window*** sizes. We can see that as the window grow bigger, it contain more than 90% of the top-k nearest neighbors, which will determine the result of the original clustering algorithm. It means that the spatial data follow Tobler's "First Law of Geography," and we can utilize the spatial autocorrelation to reduce the computation of the top-k nearest neighbors. In this way we can reduce the running time of the original algorithm while keep a satisfactory result.

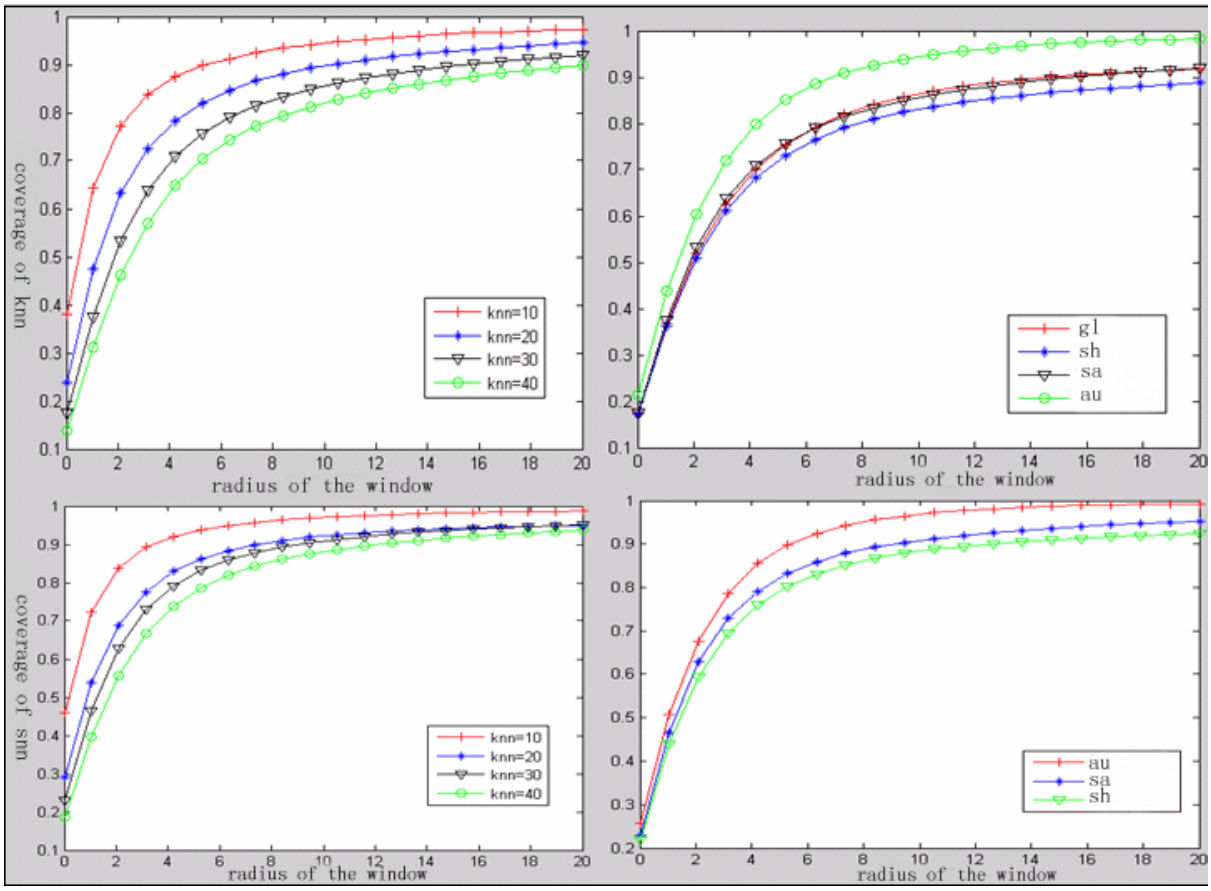

Figure 1. Average coverage with different data. gl means globle dataset, sh means south hemisphere, sa means south America, au means Australia.

4 ALGORITHMS

With the restriction of the ***window,*** the algorithm narrows down the search space for finding the top-k nearest neighbors. Using grids to store the climate data will let the algorithm find the ***window*** of a point in a constant time. The improve algorithm is described as follows:

1. Store climate data in grids format
2. For every point in the datasets, compute the similarity with every point in the ***window*** via a given R.
3. Construct the shared nearest neighbor graph from the KNN graphic. (Construct the SNN graphic)
4. Find the SNN density of each point. Using a user specified parameter, Eps, find the number points that have an SNN similarity of Eps or greater to each point. This is the SNN density of the point.
5. Find the core points. Using a user specified parameter, MinPts, find the core points, i.e., all points that have an SNN density greater than MinPts.
6. Form clusters from the core points. If two core points are within a radius, Eps, of each other, then they are placed in the same cluster.
7. Discard all noise points. All non-core points that are not within a radius of Eps of a core point are discarded.
8. Assign all non-noise, non-core points to clusters. We can do this by assigning such points to the nearest core point. (Note that steps 4-8 are DBSCAN)

For every point in the datasets, it takes constant time to find its top-k nearest neighbors, so the running time of constructing the SNN graphic is $O(R^2n)$. As we have discussed in section 2.2, the DBSCAN has a basic time complexity of $O(k*n)$, so the running time of the algorithm is $O(R^2n+kn)$. Generally, to get a satisfactory result, the R should be big enough to cover more than 95% of the top-k nearest neighbors. This means that $R^2>>k$, so we can say that the running time of the algorithm is $O(R^2n)$.

5 DATA AND EXPERIMENT

The Climate Research Unit, UK, offers the dataset "CRU TS 2.0," which gives a detailed description of the data. It contains a set of 102 years of global climate data, and it is a 0.5 degree grid dataset with five variables: cloud cover percentage, diurnal temperature range, precipitation, temperature, vapor and pressure, covering global land surface and measured on a monthly basis from the year 1901 (Mitchell, 2003).

In our experiment, we treat the temperature of every month as the attribute value of a dimension. This means that it is has 1224 dimensions. As each cell is of $0.5^{\circ}\times0.5^{\circ}$, we have 720×360 cells but with many cells with null value. The clustering result is illustrated in Figure 5, when there are 190 clusters left (including many isolated islands).

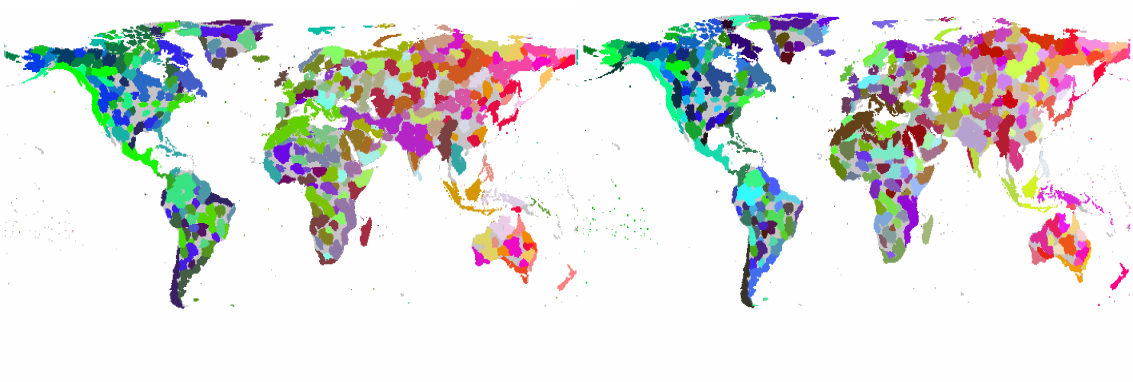

Figure 2. The left part is the original clustering result, and the right part is the improve clustering result

Table 2. The statistical data of the clustering

	Number of clusters	Number of core points	Number of outer
The Original SNN Clustering	16901	316	13732
The Improved SNN Clustering	19099	335	10805

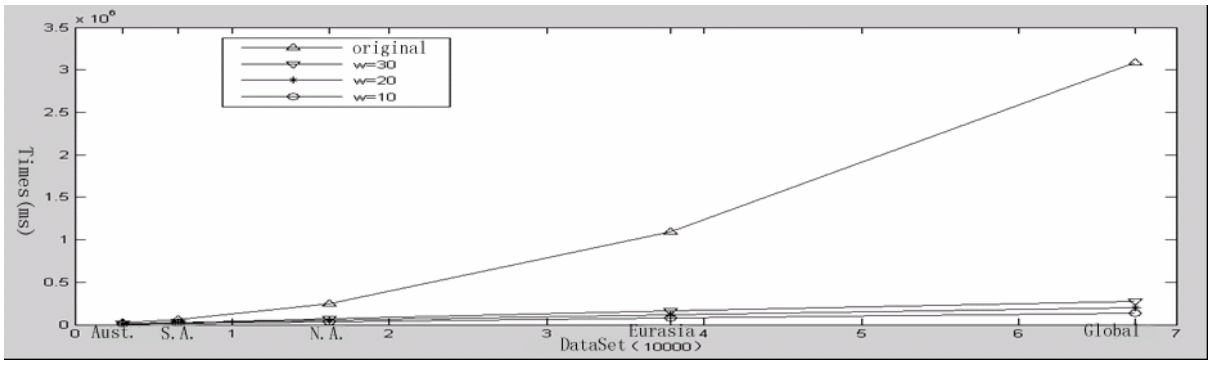

Figure 3. Running time of the original SNN clustering versus the improved one.

We can see from Figure 2 and Table 2 that the improved clustering algorithm can arrive at almost the same result as the original SNN algorithm. However, Figure 3 tells us that the improved algorithm reduces the running time from $O(n^2)$ to $O(R^2n)$.

6 CONCLUSION

In this paper, we have proposed an improved efficient high dimensional spatial clustering method based on Tobler's "First Law of Geography." We managed to reduce the relatively high time complexity of the SNN method to $O(R^2n)$, which is widely acceptable as a scalable complexity. By carefully choosing the R value, our algorithm can produce good quality and will generate spatial clusters that are continuous in space. Compared with algorithms that only consider attributes space, such as k-mean, our algorithm ignores outliers and noise and produces clusters that have a smoother boundary.

7 REFERENCES

Ertoz, L., Steinbach, M., & Kumar, V. (2002) *Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data.* USA Technical Report. Department of Computer Science, University of Minnesota, Minneapolis, MN,

Guha, S., Rastogi, R., & Shim, K. (1999) Rock: A robust clustering algorithm for categorical attributes. *Proceedings of the 15th International Conference on Data Engineering.*

Han, J., Kamber, M., & Tung, A. (2001) Spatial Clustering Methods in Data Mining: A Survey. In Miller, H. and Han, J. (eds.), *Geographic Data Mining and Knowledge Discovery*. Taylor and Francis.

Jarvis, R. & Patrick, E. (1973) Clustering using a similarity measure based on shared nearest neighbors. *IEEE Transactions on Computers C-22 (11)*, 1973.

Kolatch, E. (2001) *Clustering Algorithms for Spatial Databases: A Survey*. Department of Computer Science, University of Maryland, College Park.

Miller, H. (2004) Tobler's First Law and Spatial Analysis. *Annals of the Association of American Geographers 94(2)*, 284–289.

Mitchell, T. (2003) *CRU TS 2.0: Introduction*. Tyndall Centre for Climate Change Research, School of Environmental Sciences.