Parameter free clustering algorithm based on density and natural nearest neighbor

Yulun Wu
Guilin University of Technology
College of Information Science and Engineering
Guilin, China
e-mail:773912529@qq.com

Abstract—**The purpose of clustering algorithm is to explore the correlation of a large number of unlabeled data, so as to find valuable information in chaotic data. There are many kinds of clustering algorithms, but most of them need one or more parameters to be selected based on experience, and the selection of parameters will directly affect the accuracy of clustering algorithm, for example, DBSCAN needs domain radius and number of radius points; k-means algorithm needs to know the number of clusters in advance, k-nearest neighbor algorithm needs to be selected. Choose the appropriate number of neighbors, etc. In order to realize parametric clustering, we adopt the concept of natural nearest neighbor, and let data points discover neighbors independently by iteration. To make up for the problem that the selected K value may not be appropriate, in the process of clustering, we proposes a method of similarity between clusters, which is used to modify the misclassified clusters. Finally, the similarity of observation points is proposed. To distinguish boundary points from outliers. By comparing with DBSCAN, BIRCH and K-MEANS, proved that our algorithm can achieve good performance than other algorithms.**

Keywords—cluster; parameter free;

I. Introduction

In data mining algorithm, clustering algorithm is one of the main research fields. The so-called "people clustering, things clustering", the purpose of clustering algorithm is to divide a group of disorderly data into several clusters without any prior knowledge, so as to make the similarity process between the data objects in each cluster. The degree should be as large as possible, and the smaller the similarity between different clusters, the better. In today's highly developed network, people's work and life are constantly generating a large amount of data at all times. For example, from small to user-friendly browsing pages to consumer shopping, these data seem to have no relevance. In fact, these small traces contain the interests and habits of each user. Detailed information, and these details are creating endless value for enterprises. With the above examples, it is common to say that in clustering analysis, different users constitute different data objects, and users' interests and habits constitute different attributes of data objects, and excellent clustering algorithm can accurately achieve "people clustering". Because clustering algorithm can obtain useful knowledge without supervision, more and more people focus on clustering algorithm, eager to discover the rules and values of cluttered data through clustering algorithm. Traditional clustering algorithms are mainly divided into several categories: Partition-based cluster, Hierarchical-based clustering, Density-based clustering.

The general idea of partition-based clustering algorithm is to divide the data into k parts, among them, the classical algorithms are K-means[1] and K-medoids[2], in 2007 X-means[3] is proposed by Pelleg et al. which is an improved K-means algorithm, it only needs a parameter range, not a specific value as a parameter.

In hierarchical clustering, a hierarchical nested clustering tree is created by calculating the similarity between data points of different categories. In the clustering tree, the original data points of different categories are the lowest level of the tree, and the top level of the tree is the root node of the cluster. There are two ways to create clustering tree: bottom-up merging and top-down splitting.

Cure[4] and Birch[5] are the representative algorithm of this kind. In order to consider the proximity of the nearest neighbor node and the size of the adjacent region, the chameleon [6] is proposed. To solve the instability of clustering algorithm to noise, Balcan et al.[7] proposed robust hierarchical clustering.

Density-based clustering defines clusters as the largest set of densely connected points,it and can divide regions with sufficient densities into clusters, it can find clusters of arbitrary shape. DBSCAN [8] is the most typical representative algorithms in this kind of methods, it needs eps, minpts two parameters, if there are minpts points in the radius of eps, they are regarded as high density points.

D.Lian et al. proposed the LDBSCAN[9] with techniques for handling data with heterogeneous. To improve efficiency APSCAN[10] and DSet-dbscan[11] was proposed, because DBSCAN parameters are difficult to set, OPTICS[12] has been developed by Ankerst et al.

Although the clustering algorithm has been improved in many aspects, the setting of parameters will always affect the clustering performance in the practical application of the clustering algorithm. Different parameters will lead to different clustering results. For a data set without prior knowledge and labels, it is difficult to give a reasonable parameter directly.

In this paper, we introduce the concept of natural nearest neighbor(3N) which proposed by Dr. Zhou et al.[13] to adaptively generate nearest neighbor eigenvalues and reverse nearest neighbors of each point. Then we extend the cluster by density-based clustering, and regard the points whose number of reverse nearest neighbors is larger than the natural eigenvalues as the core points. We also propose the method of similarity between clusters. To correct the misclassification of clusters caused by inappropriate eigenvalues. Finally, the outlier points are judged as boundary points or noise points by the similarity of points.

The rest of this paper is organized as follows: In Section 2,we introduce how DBNNN works. Experimental results and performance analysis are presented in Section 3. Finally some conclusion are included in Section 4.

II. Algorithm

A. Natural Nearest Neighbor

Natural nearest neighbor is proposed by Dr.Zou Xianlin et al, a branch of nearest neighbor algorithm. In Nearest Neighbor Based Algorithms, K-nearest neighbor algorithm is widely used in machine learning and data mining. K-nearest neighbor algorithm needs to artificially give a parameter K selected according to experience, which represents k data points with the closest distance or the highest similarity for each data point. In the natural nearest neighbor algorithm, it can be understood that there is such a K value named natural eigenvalue,we express it as NE_k, but it is self-adaptive and does not need to be given artificially. The specific process is as follows: first, we take NE_k = 1, then traverse the data set, get the first nearest neighbor of each data point and the current reverse nearest neighbor of each point, and then NE_k = NE_k + 1, the k-nearest neighbor and the reverse-nearest neighbor of each point are supplemented, and the above steps are repeated until each point has at least one reverse-nearest neighbor. When traversing to the i-th data point, we express the k-nearest neighbor of data[i] as NE_k NN (i), and RNN(i) indicate the reverse neighbors of data[i] . Finally NE_k stays at one value. At this time, we can find that the denser the area, the moref reverse neighbors each point has, and conversely, the fewer reverse neighbors. the most outlying point will only have one point as reverse neighbor.

Simply describing NE_k 、 NE_k NN and RNN by algorithm is:

Algorithm 2.1

input DATASET

```
1)  k = 1
2)  Flag = 1
3)  RNN = $\varnothing$
4)  While(Flag):
5)    For i in data:
6)      ik = k_nearest(i, $NE_k$ )
7)      RNN(ik) = RNN(ik) + i
8)      $NE_k$ NN = $NE_k$ NN + ik
9)      If({RNN(i)| i $\in (data[1] \sim data[N])$ != $\varnothing$ }):
10)       Flag = 0
11)       $NE_k$ = k
12)     Else:
13)       Flag = 1
14)       k = k+1
15)  Return $NE_k$ , $NE_k$ NN, RNN
```

In algorithm 2.1 ik represent NO.k nearest neighbor of point i , RNN(i) is a subset of data set, it represents all points that regard point i as their neighbors, commonly known as the reverse neighbors of point i. Flag = 1 by default, when each observation point in the data set has at least one inverse neighbor, Flag is set to 0, then the algorithm terminates operation. algorithm 2.1 return NE_k , NE_k NN, RNN finally.

B. Expand Cluster

After automatically generating eigenvalues, this algorithm uses density-based method to distinguish different clusters. The number of reverse neighbors of observation points is used as the density of observation points. The more the number of reverse neighbors, the greater the density of observation points. We regard the observation points whose number of reverse nearest neighbors is larger than the eigenvalue as the core observation points, while those whose number is smaller than the eigenvalue as the noise points for the time being.

Random traversal starts from a data point x. If x is a core point and the number of reverse neighbors of x is greater than the number of neighbor eigenvalues NE_k , it is allocated to one of the clusters, and the core points of its neighbors are placed in a queue, at the same time, they are allocated to the same cluster as x. Before queues are listed as empty sets, we delete the first point in the queue in turn. If it is also the core point, we still assign their NE_k -nearest neighbors to the same cluster. Here is a brief description of the algorithm of expand clusters:

Algorithm 2.2 expand cluster

Input D, NE_k

```
1)  assign[ $x \in D$ ] = 0
2)  cluster = 1
3)  for x in D:
4)    If count(RNN(x) < $NE_k$ ):
5)      assign[x] = 0
6)    Else:
7)      assign[x] = cluster
8)      assign[queue] = cluster
9)      queue = $NE_k$ NN(x)
10)     While(queue $\neq \varnothing$ ):
11)       y = queue.popleft()
12)       If count(RNN(x) < $NE_k$ ):
13)       For i in $NE_k$ NN(y):
14)         If (assign[i] == 0):
15)         assign[i] = cluster
16)     cluster = cluster + 1
17)  Return assign,cluster
```

C. Similarity of clusters

Because the generation of eigenvalues depends heavily on the degree of data set regularity: if the data set has more outliers, the generated eigenvalues will be particularly large. If the data is very neat and the density is very uniform, the eigenvalues will be very small. In density-based clustering, if the eigenvalues are too large, many different clusters may be allocated together. If the eigenvalues are too small, one cluster will be divided into several clusters. In this chapter, experiments on multiple datasets show that the problem caused by larger eigenvalues is much less than that caused by smaller eigenvalues. So we mainly discuss how to solve the problem that a cluster is divided into multiple clusters. So we propose cluster similarity, which is consistent with the similarity of two clusters. We misread them as a whole. Similarity of two clusters is described as follows: for any two clusters C_1 and C_2 belong to data set, there are more than two pairs of data points belonging to two clusters, but they belongs to NE_{k}-nearest neighbor each other. For instance, suppose there are data points x_1, x_2 belong to cluster C_1, y_1, y_2 belong to cluster C_2, if the following conditions are satisfied:

$$y_1 \in NE_k NN(x_1),\ y_1 \in RNN(x_1)$$

$$y_2 \in NE_k NN(x_2),\ y_2 \in RNN(x_2)$$

We will merge C_1 and C_2 two clusters into one cluster. In order to ensure that each cluster is complete and can not be combined with any other cluster to generate a new cluster, we iterate the similarity step until the allocation of each data point in the data set does not change. Similarity is simply described by the algorithm:

```
Algorithm 2.3 Similarity
Input assign
1)  Flag = 1
2)  this_cluster = 1
3)  pair_count = 0
4)  While(Flag):
5)      For k in assign = this_cluster:
6)          For j in assign $\neq$ this_cluster:
7)              If has $\mathrm{k} \in NE_k NN(j)$, $\mathrm{k} \in RNN(j)$
8)              pair_count++
9)              if (pair_count = 2):
10)                 assign[$C_{\mathrm{k}}$] = assign[$C_{\mathrm{j}}$]
11)     if notChange(assign):
12)         Flag = 0
```

Finally, when all points other than noise are allocated, a similarity judgment is made for the remaining noise points. In order to determine a noise point x, we first traverse its NE_{k}-nearest neighbor and find the nearest point y which is not a noise point. If they are NE_{k}- nearest neighbors, we consider the noise point x and y as the same cluster. If there is no such y point, then x is a noise point.

III. Experiment And Result

A. Date sets and algorithm

In order to verify the effectiveness of the algorithm, we conduct experiments on four synthetic datasets with various shapes and densities are found on UCI for experiments.

The algorithms used for comparison are DBSCAN, K-Means, Birch. DBSCAN is the most famous density-based clustering algorithm. It identifies the core part by the sparseness of samples. Therefore, clusters of arbitrary shape can be found. It requires two parameters EPS and minpts to represent the points contained in the neighborhood radius and radius, respectively.

K-means clustering is the most famous partition clustering algorithm. Because of its simplicity and efficiency, it has become the most widely used clustering algorithm. Given a set of data points and the number of clusters required, K is specified by the user. The k-means algorithm divides the data into K clusters repeatedly according to a distance function.

The full name of BIRCH is the balanced iteration protocol and clustering based on hierarchical method. It only needs to scan the data set one time to cluster. The parameter of Birch's number of optional categories, if the user specifies the number of categories, the clustering results will return the corresponding number of clusters.

B. Experiment And Result

Figure 1 shows the clustering results of each approaches on data 'flame'. flame set is divided into upper and lower two clusters, K-Means(c) and Birch(d) two algorithms, although we provided the correct number of categories 2, it still can't cluster correctly, and it can't recognize two noise points. as for DBSCAN(b), set the parameters *eps*=1, *minpts* = 2, Because there is no obvious distinction between the two clusters. it can't recognize two clusters, DBNNN(a) obtain the right number of clusters and correctly clusters all normal points.

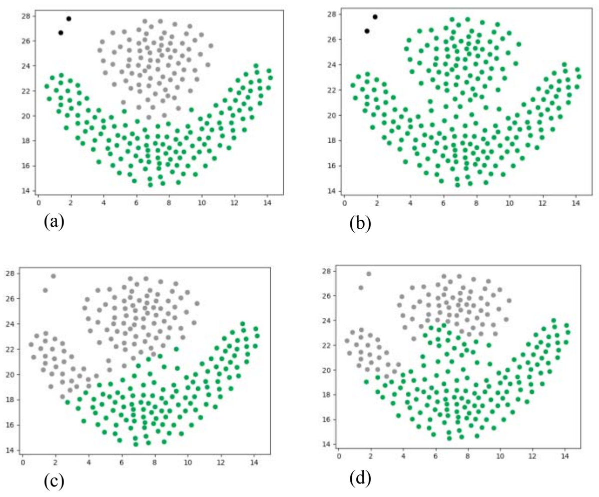

Figure 1. approaches on flame set

Figure 2 shows the clustering results of each approaches on data 'compound'. DBNNN(a) can reasonably classify all data points, but the uneven part on the right can not be distinguished. Because EPS and minpts can not be changed once they are set, it is difficult to recognize the difference of density correctly. Here we set EPS = 1.5, minpts = 2. DSBCAN algorithm ignores the sparse area directly and regards it as a noise point. As for K-Means(c) and Birch(d), because they are partition-based clustering methods, they do not have good performance in irregular shape data sets.

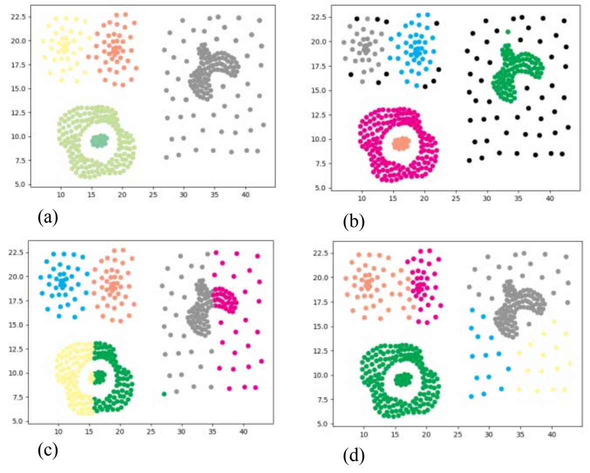

Figure 2. approaches on compound set

Figure3 shows the clustering results of each approaches on data ‘R15’, R15 has 15 clusters in total, no noise in the data set, and each cluster is regular shape, so with reasonable parameters, all 15 clusters can be identified by the four algorithms. DBNNN(a) , K-Means(c) and Birch(d) have good performance and high accuracy in noise recognition, but DBSCAN treats many boundary points as noise points, here we set *eps* = 0.5, *minpts* = 15.

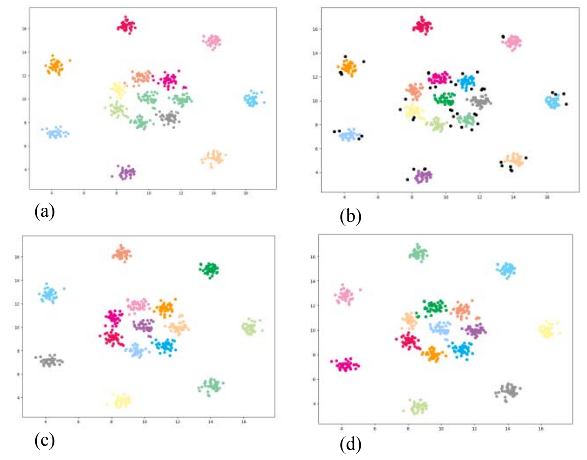

Figure 3. approaches on R15 set

Figure4 shows the clustering results of each approaches on data ‘D31’. The parameters of DBSCAN in this data set are very difficult to set, many parameters we were tried and all clusters could not be found, on the way, we set eps = 0.5, minpts = 4, it can be seen that many clusters have not been found by DBSCAN(b), and many boundary points have been identified as noise points. As for K-means(c) and Birch(d), we tell it the correct number of clusters, These two algorithms can recognize all clusters on such regular shape and clear boundary data sets, but still a cluster is misclassified in their respective results. DBNNN(a) not only can all clusters be identified, but also the accuracy is high. Only a few outlier boundary points are identified as noise points.

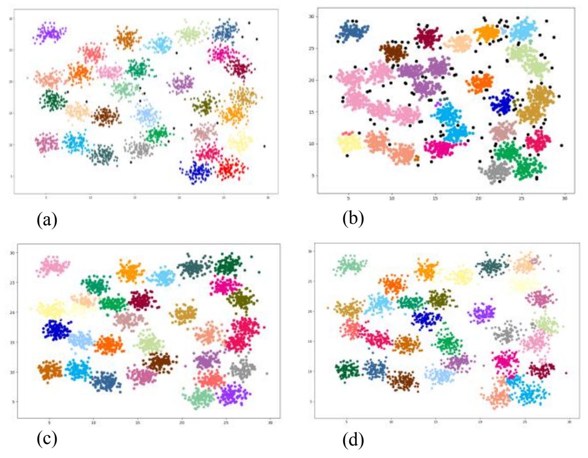

Figure 3. approaches on D31 set

IV. Conclusions

A parametric clustering algorithm based on natural nearest neighbor (3N) is proposed. First, the DBNNN algorithm adaptive generates a natural eigenvalue NE_k according to the concept of natural nearest neighbor. With the natural eigenvalue NE_k , the density of each observation point is determined by comparing the number of reverse nearest neighbors with NE_k . Then, the algorithm expands the density by the way of density expansion. To generate each cluster, in order to solve the problem of unreasonable eigenvalues, we propose the concept of similarity between clusters, which is used to synthesize misdivided clusters. Finally, we determine the attribution of boundary points by similarity between points. Experiments show that the algorithm can perform well in many data sets without providing parameters artificially, and it outperforms some classical algorithms in some data sets.

References

[1] Macqueen J. Some Methods for Classification and Analysis of MultiVariate Observations[C]// Proc of Berkeley Symposium on Mathematical Statistics & Probability. 1965.

[2] Vinod H D. Integer Programming and the Theory of Grouping[J]. Publications of the American Statistical Association, 1969, 64(326):14.

[3] Dan P, Moore A W. X-means: Extending K-means with Efficient Estimation of the Number of Clusters[C]// Seventeenth International Conference on Machine Learning. 2000.

[4] Guha S, Rastogi R, Shim K, et al. CURE : An Efficient Clustering Algorithm for Large Databases[J]. Information Systems, 1998, 26(1):35-58.

[5] Zhang T, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for var large databases. ACM SIGMOD Rec, 25(2):103-104. 1996

[6] Karypis G . Chameleon : Hierarchical Clustering Using Dynamic Modeling[J]. IEEE Computer, 1999, 32.

[7] Balcan, Maria-Florina, Liang Y, Gupta P. Robust Hierarchical Clustering[J]. Eprint Arxiv, 2014.

[8] Ester M, Kriegel, Hans-Peter, Sander J, et al. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise[C]// International Conference on Knowledge Discovery & Data Mining. 1996.

[9] Lian D, Xu L, Feng G, et al. A local-density based spatial clustering algorithm with noise[J]. Information Systems, 2007, 32(7):978-986.

[10] Chen X, Liu W, Qiu H, et al. APSCAN: A parameter free algorithm for clustering.[J]. Pattern Recognition Letters, 2011, 32(7):973-986.

[11] Jian H. DSets-DBSCAN: A Parameter-Free Clustering Algorithm[J]. IEEE Transactions on Image Processing, 2016, 25(7):3182-3193.

[12] Ankerst M, Breunig M M, Kriegel H P. OPTICS:ordering points to identify the clustering structure[C]// Acm Sigmod International Conference on Management of Data. 1999.

[13] Zou X L, Zhu Q S, Yang R L. Natural Nearest Neighbor for Isomap Algorithm without Free-Parameter[J]. Advanced Materials Research, 2011, 219-220:994-998.