

SUMMARY

Introduction:

The process of grouping a set of physical or abstract objects into classes of similar objects is called Clustering, i.e. within the same cluster, the similarity between objects is more and dissimilar objects stay in other clusters.

In data mining there are many categories of clustering methods, such as partitioning methods, hierarchical methods, Density based clustering etc. Among these, partitioning methods and agglomerative methods are the most popular. In this paper an improvement of these methods is introduced.

Problems:

Among agglomerative methods, Greedy Agglomerative Clustering method initializes each object as a cluster of size one, compute the distances between all pairs of clusters, sort the distances from smallest to the largest and then repeatedly merge two clusters which are closes together until one is with the desired number of cluster. The computational complexity of this method is in the order of n^2 . But when the number of objects is very high, usually in millions or billions, the or when the number of clusters is very high the GAC method's complexity becomes massive.

In k -Means clustering method, first arbitrary k objects as cluster centers and (re)assign each object to the cluster to which it is the most similar. Based on the mean value of the objects in the cluster, update the center of the cluster means, until there is no change on cluster centers. But the complexity increases drastically when the number of cluster is high and the number of objects is very high, where it is obvious while data mining.

The other problem is when the features dimensionality is high for an object to be clustered, finding the similarity is troublesome and takes a lot of computation time.

Summarizing the conditions that may cause problems:

1. The number of objects might be very high
2. The number of clusters might be high
3. Each object might have a large number of features.

Solution:

This paper presents a technique to deal with the three cases. The key idea of this technique is to deferring the main clustering procedure till the number of datasets

on which the clustering method has to be applied is reduced. That means the clustering method is applied only on the datasets that are candidates of being in a cluster. This is performed by effectively dividing the data into overlapping subsets, which are called “Canopies”. An extremely inexpensive cheap distance metric can be used and to find the data objects near the center of a region, and use an expensive clustering method on the lowered number of datasets. This expensive distance measurement is found only between the objects that are in a common canopy. Eliminating all the distance comparisons among points that do not fall within a common canopy saves the significant computation.

Methodology:

The canopy algorithm uses two distance measurements, one cheap and one expensive distance measurement.

For example if the clustering algorithm is trying to find the groups of motor insurance policy holders, the cheap distance metric can be the address or the words in common between two addresses. The expensive distance measure can be field-by-field comparison between the attributes of policyholders.

A canopy is a simple subset of data objects that are in some distance threshold according to an approximate similarity. An object may appear in more than one canopy, and every element must appear in at least one canopy. The number of analyzable clusters are reduced by the assumption, that objects that are not in common canopies are far dissimilar and need not be considered for further distance measurements. The authors presented two conditions under which the canopies procedure preserves the accuracy of the traditional clustering. We are stating them as it is.

For every traditional cluster, there exists a canopy such that all Elements of the cluster are in the canopy.

This is true for partitioning methods in which the distance to a cluster is measured to the center of the cluster, i.e. k -means clustering or EM Clustering.

For every cluster, there exists a set of canopies such that The elements (or objects) of the cluster “connect” the canopies.

This is true for Greedy Agglomerative Clustering algorithm.

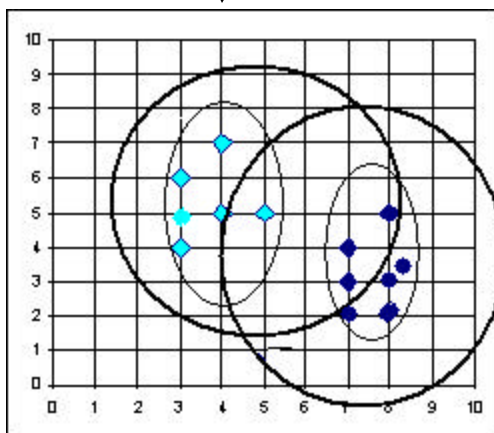
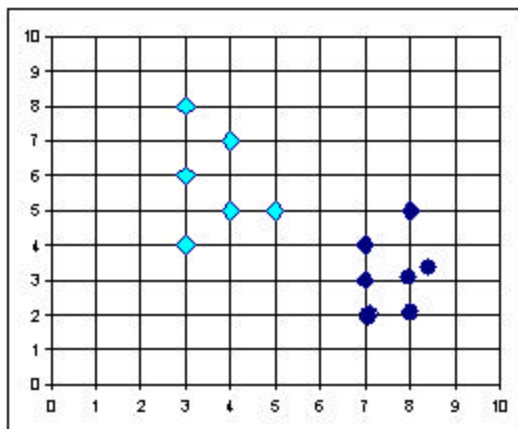
How to create canopies?

The first stage of this methodology is to create the canopies. Two distance thresholds T_1 , and T_2 , where $T_1 > T_2$ can be defined by the user and O is the list of all objects. When there are n objects, Canopy creation algorithm is as follows:

Create_Canopy (O, n, T_1, T_2)

```
{  
  while ( $O \neq \emptyset$ )  
  {  
    1. Pick an object off the list randomly  
    2. Approximately measure the distance from all other objects (Here  
       cheap distance metric is used to measure the distance).  
    3. Put all the objects that are within the distance of  $T_1$  into a canopy  
    4. Remove the objects that are within the distance  $T_2$  from the list  $O$ .  
  }  
}
```

In the above algorithm, first find the objects that are within the threshold T_1 and encapsulate them within a canopy and remove those objects that are within the threshold T_2 where $T_1 > T_2$. To find the distances, a cheap distance metric can be used, so that a significant amount of computation can be reduced. In this paper, the authors described the algorithm for reference matching in research papers.

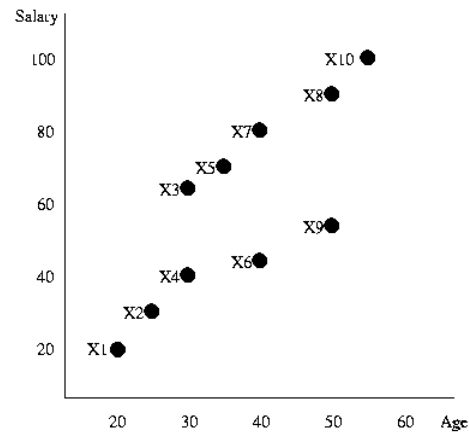


In the above example, dark lines represent the canopy with threshold T_1 and the inner circle represent the objects those are in threshold T_2 . The expensive algorithm is applied to only those objects that are in the same canopies.

An Example:

Here we are presenting an example to show which problems the traditional methods have and how to solve them with this canopy method.

	Age	Salary
X1	20	20
X2	25	30
X3	30	65
X4	30	40
X5	35	70
X6	40	45
X7	40	80
X8	50	90
X9	50	55
X10	55	100

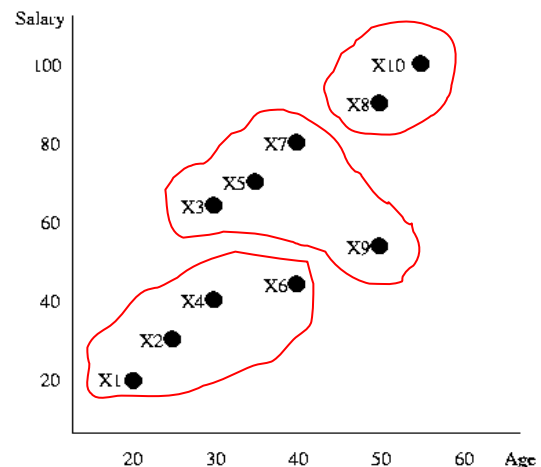


Let the Datamining objective of the above dataset is to find the position of an employee and how his/her salary varies.

With traditional k -means algorithm, let the number of clusters is 3 and the initial partition of the salary ranges are 30-35k, 40-70k, and 50-90k. Then the data objects belong to different clusters and the centroids are calculated in the next iteration. After first iteration the clusters look like this.

	K1(30,35)	K2(40,70)	K3(50,90)	
X1	25	70	100	K1
X2	10	55	85	K1
X3	30	15	45	K2
X4	5	40	70	K1
X5	40	5	35	K2
X6	20	25	55	K1
X7	55	10	20	K2
X8	75	30	0	K3
X9	40	25	35	K2
X10	90	45	15	K3

New K1(29,34) K2(39,68) K3(53,95)



After second iteration the data points are as follow:

	K1(29,34)	K2(39,68)	K3(53,95)	
X1	23	67	108	K1
X2	8	52	93	K1
X3	32	12	53	K2
X4	7	37	78	K1
X5	42	6	43	K2
X6	22	24	63	K1
X7	57	13	28	K2
X8	77	33	8	K3
X9	42	24	43	K2
X10	92	48	7	K3

But as the problems are mentioned before, when the problem size is very high the number of calculations and comparisons increase to a very high. The solution is reducing the number of calculations. This can be achieved by canopies method. The following pictures give the insight of canopy method.

	K1(30,35)	K2(40,70)	K3(50,90)	
X1	25	X	X	K1
X2	10	X	X	K1
X3	X	15	X	K2
X4	5	40	X	K1
X5	X	5	X	K2
X6	20	25	X	K1
X7	X	10	20	K2
X8	X	X	0	K3
X9	40	25	X	K2
X10	X	X	15	K3

New K1(29,34) K2(39,68) K3(53,95)

■ Canopy

- C1={X1, X2, X4, X6, X9}
- C2={X3, X4, X5, X6, X7, X9}
- C3={X7, X8, X10}

To make the example simple, we have taken the same dataset for this method as that of traditional method. In the first stage the canopies are created using a cheap distance metric, in this case the salaries, without considering all the features of the dataset. In traditional method, to find the distances we have to consider all the features, but here we have considered only one feature to reduce the number of comparisons.

In the second stage distances are measured only between the data objects that are in the same canopy and the overlapping canopies. Here the distance is

computed based on all the features. Thus the number of expensive distance computation is reduced.

	K1(30,35)	K2(40,70)	K3(50,90)	
X1	23	X	X	K1
X2	8	X	X	K1
X3	X	12	X	K2
X4	7	37	X	K1
X5	X	6	X	K2
X6	22	24	X	K1
X7	X	13	28	K2
X8	X	X	8	K3
X9	42	24	X	K2
X10	X	X	7	K3

After the second iteration the data objects are as above. This procedure continues in the traditional method from here.

Computational Complexity:

As shown above the complexity for greedy agglomerative clustering (GAC) method is $O(n^2)$ and the complexity for EM is $O(nk)$. But when canopy technique is used the complexity reduces substantially. If there are c canopies and each object belongs to f canopies on average, then the complexity each canopy contains fn/c objects. Then it requires at most $O(c * (fn/c)^2) = O(f^2 n^2 / c)$ distance measurements, when GAC is used. That means there is f^2/c times reduction. When EM is applied to measure the expensive measurements, the complexity is $O(nk^2/c)$ and the complexity is the same as that of GAC.

Conclusion:

This paper proposed a technique that can drastically reduce the number distance calculation between the datasets in the case of large number of objects, large number of clusters, and for the higher feature dimensional objects. This is proved for reference matching application and can be applied for many other number massive datasets.

Our opinion:

We rate this paper at 7 in the scale of 10, where 10 is the best or “strongly agree”. We agree that this paper is introducing one of the simplest methods and effective methods to reduce the computation overhead that is in traditional clustering algorithms. The complexity can be reduced extensively, for example by a factor of 1000 or 10000, when the creation of canopies is done carefully. But, the problem with this approach is how to decide on the cheap distance metric

that is used to create a canopy. The problem relies in deciding the cheap distance metric, where a user should have proper insight of the data he is mining. This cheap distance metric varies for each dataset.

We believe that this paper is presenting a methodology that can be implemented, but hasn't given an exact algorithm that can be used generally, such as *a priori* algorithm we studied in our course.

In this paper the authors, used an application of reference matching, which introduces the methodology neatly and simply. We agree with this method and believe it is very obvious solution to reduce the complexity of traditional clustering algorithm.

Suggestion of Improvement:

However, we find one more problem after applying this method, and here we also give the improvement. The problem is, we might find it difficult to make smaller canopies based on the current cheap distance metric. Our improvement for this method is, when there are a large number of objects in each canopy, we can implement canopy creation within those canopies based on another distance metric and further reduce the number of comparisons (recursive till we get the expected number of canopies or some other threshold). Because of this method cheap distance metric decision can be automated to some extent. For example, let the distance be calculated based on one feature of the dataset. If a feature is unable to create the number of canopies we are expecting, next iteration of creating the canopies could be run using another feature of the dataset. But the user's choice of cheap distance metric should be supported, which could result more accurate canopies. We haven't proved this, but this is just our intuition.