Minimal Parameter Clustering of Complex Shape Dataset with High Dimensional Dataset Compatibility

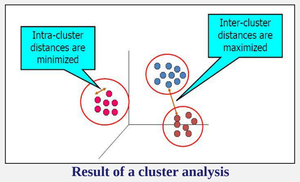
Ahmed Al Muzaddid (0805053) and K.M.A. Solaiman (0805116)

1. Introduction

Clustering Analysis is defined as a way to group relatively homogeneous cases or observations in a separate set which is different from other sets or objects formed outside this particular group. Clustering is a form of learning by observation, rather than learning by examples. In machine learning, clustering is an example of unsupervised learning.

Clustering do not rely on predefined classes and class-labeled training examples. By automated clustering, we can identify dense and sparse regions in object space and, therefore, discover overall distribution patterns and interesting correlations among data attributes. Cluster analysis has been widely used in numerous applications, including market research, pattern recognition, data analysis, and image processing.

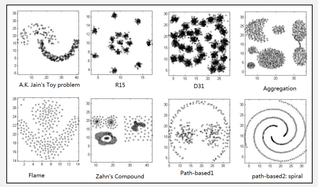
Our approach focuses on developing a parameter less clustering technique which is effective for clustering complex shapes and types of data. We also focus on formulating a high dimensional clustering technique along the way.



Result of a cluster analysis

2. Problem Definition

- Existing clustering algorithms generally use Euclidean or Manhattan distance measures for forming clusters. Algorithms based on such distance measures tend to find spherical clusters with similar size and density. But a cluster could be of any arbitrary shape.
- Most of the existing clustering algorithms require certain input parameters to be explicitly incorporated. The clustering results can be quite sensitive to input parameters. Parameters are difficult to determine for data sets containing high dimensional objects.



Complex Data Sets

- Some clustering algorithms cannot incorporate newly inserted data (i.e., database updates) into existing clustering structures and, instead, must determine a new clustering from scratch.
- Some clustering algorithms are sensitive to the order of input data.
- Most real-world databases contain outliers, missing data, unknown data or erroneous data. Some clustering algorithms are sensitive to such data and may lead to poor quality clusters.
- A database or a data warehouse can contain several dimensions or attributes. Finding clusters of data objects in high dimensional space is challenging, especially considering that such data can be sparse and highly skewed.

3. Objectives

Our objective is to design a new clustering algorithm that fulfills the following qualities:

- Minimal External Parameter for Clustering Analysis
- Improving scalability of online clustering methods
- Incremental clustering and insensitivity to the order of input records
- Finding methods for clustering complex shapes and types of data
- Clustering in High dimensional data space

5. Simulation Output

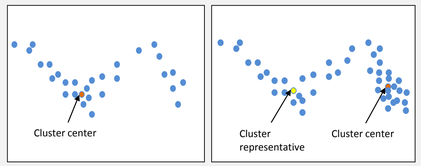
Some snapshots of a simulation run of our proposed algorithm is shown here. As new sample arrives from the input dataset, the augmented distribution changes and it forms a new cluster or joins any existing clusters.

4. Our Method

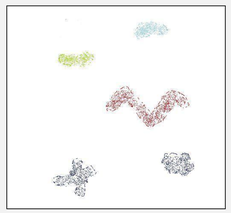
To handle the outlier and complex shape cluster data we have initially developed an "augmented" density function $f(x)$ using Fourier transformation such that if $P(x_i) > P(x_j)$ then $f(x_i) > f(x_j)$, where $P(x)$ is the probability density function of sample distribution. To update this augmented density function for every new sample we have to do a little calculation since $F(u)=F^*(u) + f(x)e^{-j2\pi x/M}$, where $F^*(u)$ is previous Fourier coefficient and $F(u)$ is the updated one . To filter the outlier we have only used some low frequency component of Fourier transformation. Once "augmented" density function is ready, algorithm 1 is invoked to do the rest of the clustering procedure. In algorithm 1, we have used algorithm 2 to find out in which cluster any sample point is situated.

Algorithm 1: merge and span cluster(S)

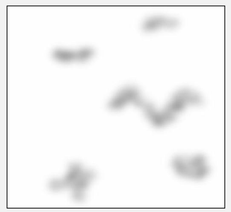
1. Set of Cluster C←*Null*
2. Set of Cluster representative R←*Null*
3. For each sample S
4. If π(S) > t
5. k ← in _which_cluster(S)
6. If k >= 0
7. If π(S) > π(k.center)
8. Make S_i as new cluster center of cluster C
9. Else
10. Create a new cluster Q with S_i as cluster center
11. C ← C U {Q}
12. End if
13. End if
14. Add sample S_i to the augmented distribution
15. End for



Changing of Cluster Center



Complex Colored Cluster Samples

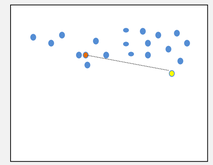


Augmented Distribution

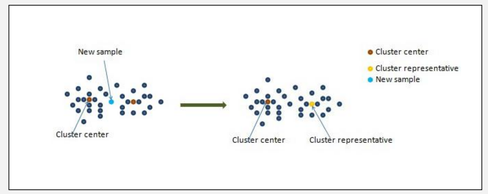
Algorithm 2: in_which_cluster(S)

Input: new sample*(s)*
Output: which cluster contains input sample

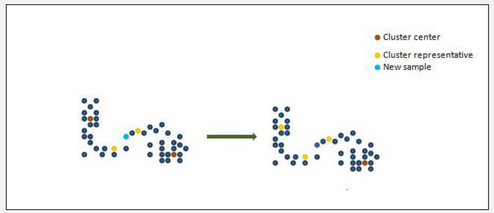
Belonging set, B←*Null*
For all neighbor cluster C
σ ← line_intregal(s,C.center)
ι← length(s,C)
difPi = | π(c.center) σ/ ι |
if *difPi* < ε
B ← B U {C}
End if
End for
If B has single element C'
return C'.cluster_index
else if B has more then single element
merge those cluster
return cluster index
else
for all neighbor representative r of S from R
σ ← line_intregal(S,r)
ι← length(S,r)
difPi = | π(r) σ/ ι |
if *difPi* < 2ε
R← R U {r}
B← B U {r}
End if
End for
If B has single element C'
return C'.cluster_index
else if B has more then single element
return most probable one



Linear path does not inscribe within augmented distribution. So line integral σ is small and ε<*difPi*<2ε will hold. So new sample will be a representative for this cluster.



Merging using Cluster Center



Merging using Cluster Representative

Summary

Our algorithm can handle complex & arbitrary shapes of clusters. In our approach, parameter t and ε will not affect the clustering performance, only run time will vary a little. For fitting augmented distribution to dataset no parameter or prior knowledge is required. It helps online clustering to be more reliable & parameter independent. In our current algorithm, categorical data are not handled. But there is a scope to introduce categorical data within this algorithm. Time complexity increases with the feature dimension for dependent features. But we can preprocess the data to distinguish the main features using principle component analysis to reduce the feature dimension and as a result time complexity will be also improved.

References

1. Kaufman, L., & Rousseeuw, P. (1990). Finding groups in data: An introduction to cluster analysis. John Wiley & Sons.
2. Hieu T. Nguyen , Arnold Smeulders, Active learning using pre-clustering, Proceedings of the twenty-first international conference on Machine learning, p.79, July 04-08, 2004, Banff, Alberta, Canada
3. Clustering Datasets
4. Aldenderfer, M.S. and Blashfield, R.K. 1984. Cluster Analysis. Beverly Hills, CA: Sage Press
5. Dasgupta, S. and Hsu, D., Hierarchical Sampling for Active Learning, Department of Computer Science and Engineering, University of California, San Diego
6. Brzezinski, D. and Stefanowski, J., Reacting to Different Types of Concept Drift: The Accuracy Updated Ensemble Algorithm, IEEE Transactions on Neural Networks and Learning Systems.

Department of Computer Science and Engineering (CSE), BUET