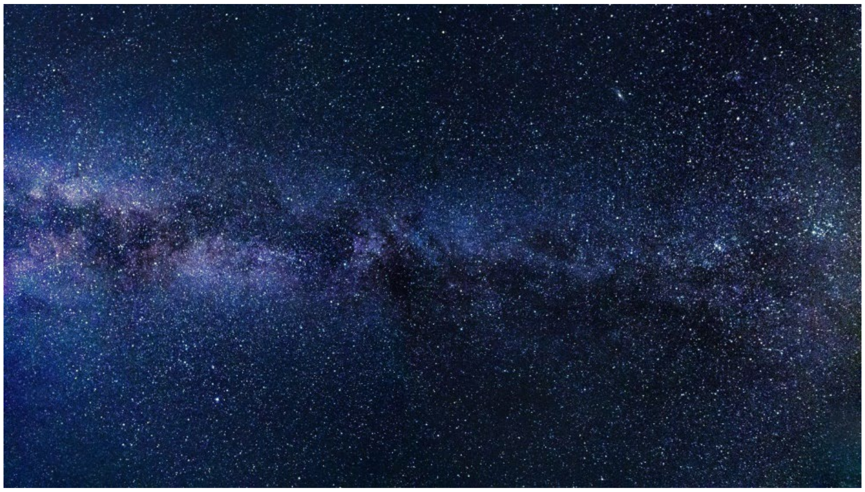


Clustering with the CURE Algorithm

Diego Freire | Data Mining | May 2022

Problem Description

CLUSTERING

According to the course, Mining of Massive Datasets textbook, “Clustering is the process of examining a collection of “points,” and grouping the points into “clusters” according to some distance measure.” (Leskovec, Rajaraman, & Ullman)

Clustering is an unsupervised Machine Learning technique used to group which automatically groups unlabeled data, looking for similar patterns in the data set, then it divides groups in presence or absence of similar patterns.

To use a clustering algorithm, the dataset should be a collection of points that belong to a space where the points can be located.

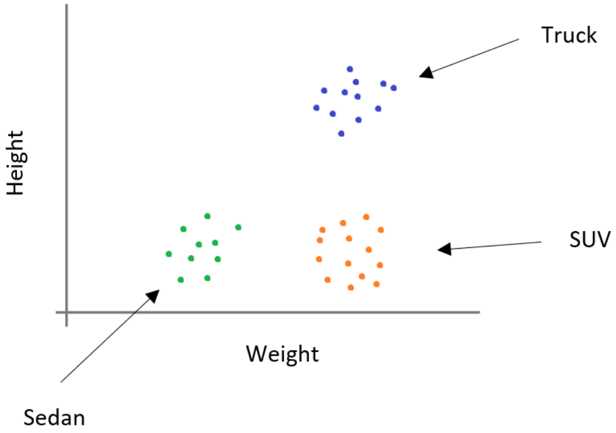


Figure 1

The most common uses of clustering are:

- Market Segmentation
- Statistical data analysis
- Social network analysis
- Image segmentation
- Anomaly detection
- Recommendation systems (Amazon, Netflix)

Clustering Strategies

- Hierarchical: It is a type of clustering starts with a single point cluster, and moves to merge with another cluster, until the desired number of clusters are created. There are two types of hierarchical clusters, combined based on their distance:
    - Agglomerative
    - Divisive
- Point Assignment: Considers the points in order, assigning each to the cluster that fits best. The process can be preceded by an initial cluster estimation.
  Some variations combine or split clusters and remove unassigned points if they are outliers.

Clustering algorithms group points from a data set, with a notion of distance between points, in a way that:

- Members of a cluster are close when the look like each other.
- Members of different clusters are dissimilar.
- Usually points are in a high-dimensional space.
- Similarity defined using a distance measure like Euclidean, Cosine or Jaccard.

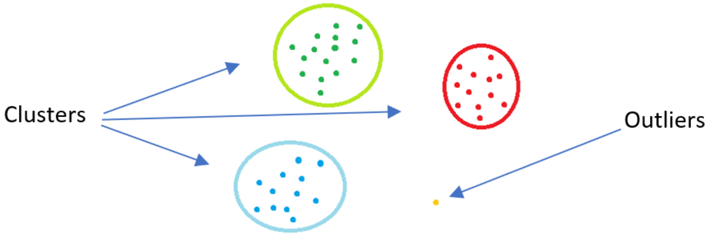


Figure 2

The Curse of Dimensionality

It is the phenomenon that occurs when handling high dimensional data. It manifests when the distance of the points is equal from one another, or when two vectors are orthogonal.

Description of the algorithm

CURE - CLUSTERING USING REPRESENTATIVES

The CURE algorithm is a hierarchical based clustering technique, which adopts a middle ground between the centroid based and the all-point extremes. (geeksforgeeks, 2021)

This algorithm is used for identifying spherical and non-spherical clusters. Allows clusters to take any shape, in other words the CURE algorithm assumes Euclidean space and identifies clusters of any space. CURE uses a collection of representative points for efficiently handling the clusters and eliminating the outliers.

The CURE algorithm can be good for discovering groups and identifying interesting distributions in the underlying data.

Analysis of the algorithm performance

Why does it work?

The main point of the CURE algorithm is to create partitions starting with a random sample from the data set. This is performed recursively, allowing to remove outliers, create the partial clusters and finally labeling the data.

Algorithm Steps

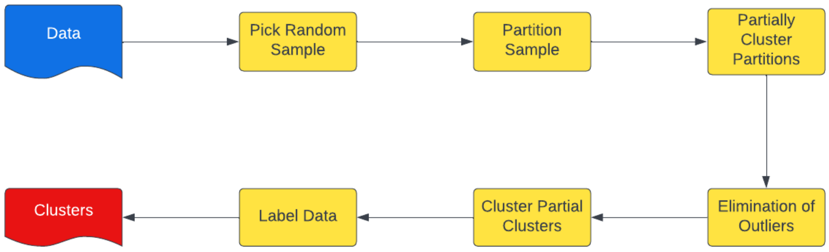


Figure 3

Initialization

- Take a small random sample and cluster it to the main memory using hierarchical clustering
- Select a small set of points from each cluster to be representative points. The points will be selected as far as possible from one another.
- Move the partition containing representative points to a fixed fraction of the distance between its location and the centroid of its cluster.

Next Steps

- After the initialization completes, cluster the remaining points and create an output cluster.
- Finally, merge two clusters if they representative points sufficiently close, defining a distance threshold. Repeat this step with all the clusters until there are no more close points.

What is its cost?

The cost of using hierarchical clustering could be high because this type of algorithms requires to compute distances between pairs of clusters. For example, let's think about a basic hierarchical clustering algorithm:

- The first step takes O(n^2) time.
- Subsequent steps take proportional time (n – 1) ^2, (n– 2) ^2, …
- The algorithm ends up being cubic O(n^3).

This could limit the use of the algorithm only for a small data set.

Using random samples and partitions help to improve the efficiency of the CURE algorithm. The cure algorithm was designed using a combination of partition based and hierarchical algorithms which improves the performance, allowing it to be used with large scale data sets.

Pros and cons?

- The CURE algorithm is capable to identify arbitrary shape clusters.
- The algorithm is robust in the presence of outliers.
- It is suitable for handling large data sets.
- CURE is not good for noise handling.
- Given the quantity of the data it can handle the algorithm is slow
- The algorithm cannot be applied to large databases because of its high execution time

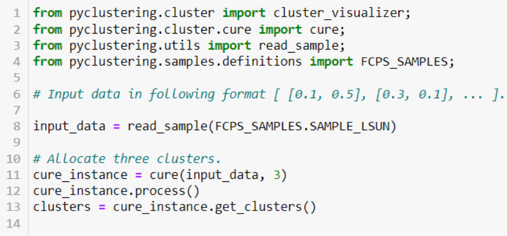
Example of how the algorithm works:

Attachment contains a python notebook

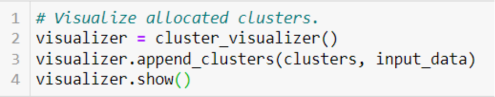
- Choose a suitable dataset or generate one
- Implement the algorithm
- Show the results obtained from the algorithm

Results using Pycluserting cure implementation

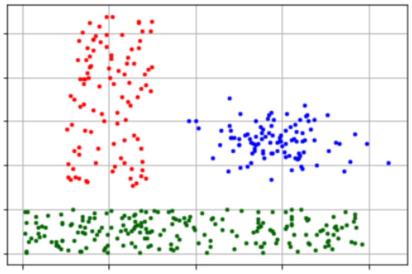
I used the SAMPLE_LSUN dataset provided by the Pyclustering library.



```
from pyclustering.cluster import cluster_visualizer;
from pyclustering.cluster.cure import cure;
from pyclustering.utils import read_sample;
from pyclustering.samples.definitions import FCPS_SAMPLES;

# Input data in following format [ [0.1, 0.5], [0.3, 0.1], ... ].

input_data = read_sample(FCPS_SAMPLES.SAMPLE_LSUN)

# Allocate three clusters.
cure_instance = cure(input_data, 3)
cure_instance.process()
clusters = cure_instance.get_clusters()

```



```
# Visualize allocated clusters.
visualizer = cluster_visualizer()
visualizer.append_clusters(clusters, input_data)
visualizer.show()
```



For didactic purposes, I created a notebook using Kun Chu's implementation of the CURE algorithm in python with one of the provided datasets. (Kun, n.d.)

Attachments

Cure.ipynb (Pyclusering)
Cure2.ipynb (Kun Chu's implementation)
data.txt

Bibliography

CR, A. (2021). *Clustering Algorithms*. Retrieved from Neptune Blog: https://neptune.ai/blog/clustering-algorithms

geeksforgeeks. (2021). Retrieved from geeksforgeeks.org: https://www.geeksforgeeks.org/basic-understanding-of-cure-algorithm/

Khan, H. (2020). *Clustering in Machine Learning*. Retrieved from https://ai.foobrdigital.com: https://ai.foobrdigital.com/clustering-in-machine-learning/

Kun, C. (n.d.). *CURE cluster python*. Retrieved from github.com: https://github.com/Kchu/CURE-cluster-python

Leskovec, J., Rajaraman, A., & Ullman, J. (n.d.). *Mining of Massive Datasets, 3rd Ed.*