Clustering Algorithms

Xiangyu Luo (20374111), Yufan Lu (20373248)

Abstract

This report provides a literature survey result conducted with a focus on one of the most commonly used unsupervised learning algorithms. To be specific, clustering algorithms. In this report, two main categories of the clustering algorithms are discussed: hierarchical clustering and k-means clustering. Hierarchical clustering and k-means clustering are two different kinds of clustering algorithm since they use different methods for choosing and generating clusters. Both of the algorithms are appropriate for a high-dimensional dataset. K-means algorithm is efficient while hierarchical clustering runs slower since it produces a hierarchy or structure of data.

1 Introduction

Clustering is one of the most commonly used unsupervised learning algorithms. Its applications are wide across different disciplines[1]. Without any prior knowledge, clustering algorithms are often used to generate a general picture of the data set.

Intuitively, clustering is a process of grouping the data into clusters according to some distance measurements. The result of clustering is a collection of groups or called clusters. Objects in the same cluster are more similar to each other while objects from different cluster share few things in common.

Various distance measurements can be used in clustering algorithms. For example, under Euclidean space, common Euclidean distance measurement is commonly used in cluster analysis. Under non-Euclidean space, alternative distance measurements such as numbers of features, memory allocations, and other customized criteria are used. Moreover, it is easy to shown that different measures can produce different sets of clusters. Thus, the selection of distance measure is important since it influence the outcome of clustering.

This report gives brief discussions of hierarchical clustering and k-means algorithms in Euclidean space. It then compares the two cluster algorithms through a brief example. All the graphs presented in the report are generated using Python. And two clustering algorithms are used for generating the graphs are implemented by SciPy.

2 Overview of Clustering Algorithm

When an agent learning from provided examples, there are typically three types of feedback, which determine the three main types of learning: supervised learning, reinforcement learning, and unsupervised learning[2]. In supervised learning problems, an agent tries to come up with a function that approximates the true function in the hypothesis space. However, there are not any functions that we try to find in unsupervised learning problem. Instead, we wish to organize the data in a meaningful way.

Hierarchical clustering and k-means clustering use different strategies for grouping the input objects based on different distance measurements [1]. The former starts with each point as a cluster and then clusters are combined based on their "closeness", or distance from each other. The later has a different strategy: given the initial guess of clusters, each point is assigned to the cluster into which it best fits. Despite their differences, they share the following common setup[5]:

- **Input** - a set of elements, $\mathcal{X}$, and a distance measurement function over it, where

$$d : \mathcal{X} \times \mathcal{X} - > \mathbb{R}+$$

- **Output** - a partition of the domain set $\mathcal{X}$ into subsets. $C = (C_1, ..., C_k)$ where

$$\cup_{i=1}^{k} C_i = X \text{ and } C_i \cap C_j = \emptyset \text{ for } i \neq j$$

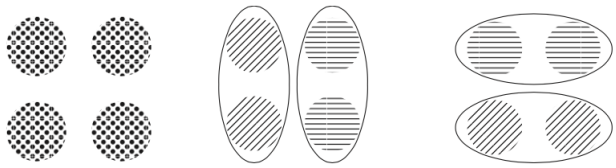

Figure 1: A Simple Input and Two Possible Outputs after Clustering

3 Hierarchical Clustering

3.1 Hierarchical Clustering in a Euclidean Space

Hierarchical clustering is one of the agglomerative algorithms. Each cluster is represented by its centroid or average of the points in the cluster. It compares the distance between centroids and then merges the two clusters at the shortest distance until some requirements, such as number of clusters, are met. The hierarchical algorithm can be described as the

following pseudo code[1]:

```
WHILE none of the stop requirements are met DO
      Pick the best two clusters to merge
      And combine those two clusters into one cluster
END
```

The figures provided below show a simple process of hierarchical clustering in the Euclidean Space.

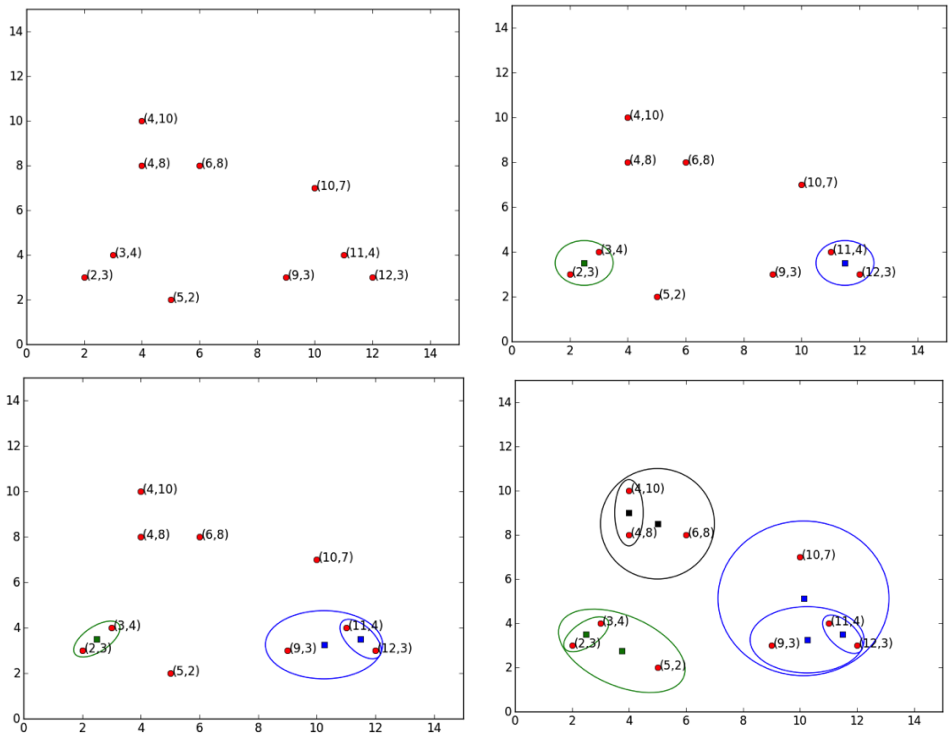

Figure 2: Simple Process of Hierarchical Clustering

3.2 Efficiency of Hierarchical Clustering

Overall, the basic algorithm for hierarchical clustering is not very efficient. The initial step takes $O(n^2)$ times, and $(n-1)^2$, $(n-2)^2$, ... for subsequent steps. Thus, overall, the basic hierarchical clustering algorithm takes $O(n^3)$. If we use a priority queue to store the their distance, the algorithm takes $O(n^2 \log n)$

4 K-Means Clustering

4.1 K-Means Clustering in Euclidean Space

K-means algorithm is one of the famous point-assignment algorithms. It assumes a number of clusters, k, is known in advance. Each point other than those select points, which represent different clusters, is assigned to the closest cluster. It may also deduce number of clusters by trial and error. The k-means algorithm can be described as the following pseudo code[1]:

```
Initially choose k points as clusters
FOR each unassigned point DO
      add this point to the closest cluster
      adjust the centroid of that cluster to account this point
END
```

The figures provided below show a simple process of k-mean clustering in the Euclidean Space.

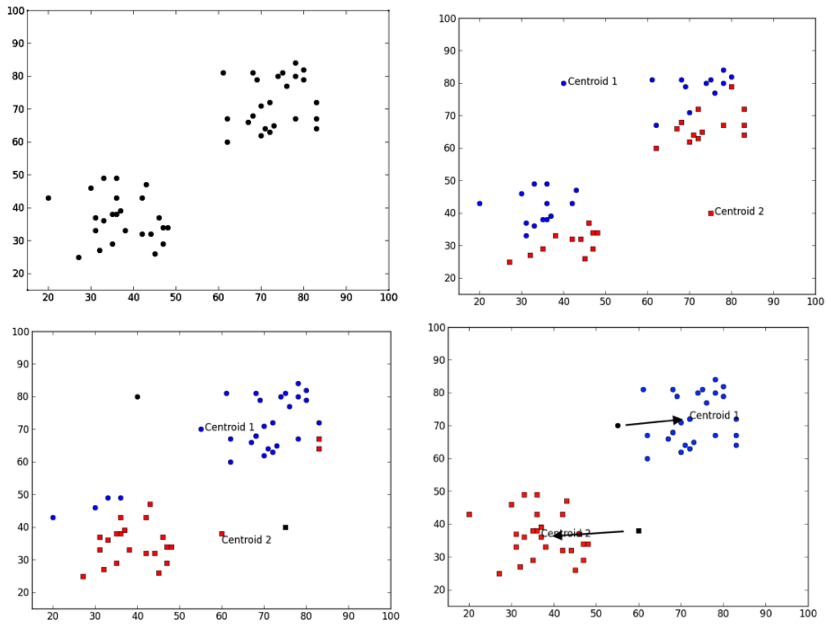

Figure 3: Simple Process of k-mean Clustering

K-means objective function is one of the most popular clustering objectives. The data is grouped into k disjoint set $C_1, \cdots, C_k$ where each C_i is represented by a centroid μ_i.

Mathematically, given input set $\mathcal{X}$ and $\mu_i \subseteq \mathcal{X}$. So, the centroid of C_i is defined as[5]

$$\mu_i(C_i) = argmin_{\mu \in \mathcal{X}'} \sum_{x \in C_i} d(x, \mu)^2$$

Thus, the k-means objective is[5]

$$G_{k-means}((\mathcal{X}, d), (C_1, \cdots, C_k)) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i(C_i))^2$$

4.2 Choosing the Right Value of k

The k-means algorithm assumes k is known in advance. However, picking the right value of k is difficult sometimes. If we can measure the quality of the clustering for various k, then we can usually guess what the right value of k is[1]. Normally, average radius or diameter of clusters are used. The relation between average diameter and number k is given as below

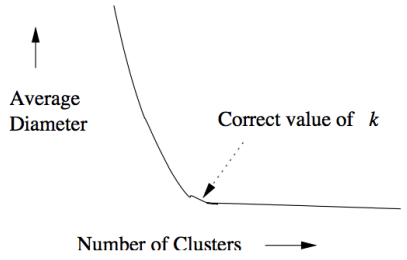

Figure 4: Relationship between k and average diameter

A practical method is to begin with $k = 1, 2, 4, 8, \cdots$, and we choose the value v where there is very little decrease in average diameter between v and $2v$[1].

5 Handling Large and High Dimensional Data

5.1 The BFR Algorithm

Normally, target datasets are large, complex, and high-dimensional. Thus, it is harder to compute "distance" between "objects", and grouping data into clusters since the size of the data is too large to fit in the main memory[5].

As a variant of k-means clustering, the BFR algorithm is designed to cluster data in high-dimensional Euclidean space[1]. However, it makes a strong assumption about the shape of the cluster: they must be normally distributed about a centroid. The BFR algorithm

begins by selecting k points, using the method shown in 4.2. Then the points of the data file are read in chunks. Each chunk consists three types of objects[1] as shown in the Figure 5:

1. *The Discard Set* — simple summaries of the clusters themselves.
2. *The Compressed Set* — sets of points that have been found close to one another, but not close to any cluster.
3. *The Retained Set* — a collection of certain points can neither be assigned to a cluster nor are they sufficiently close to any other points in a compressed set.

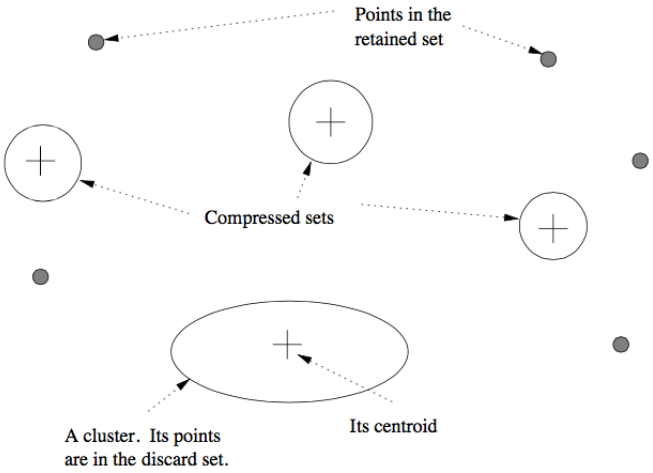

Figure 5: Three types of objects[1]

With this mechanism, the BFR algorithm can handle large data set without having memory related issues. However, due to this mechanism, the distance measurement used in BFR algorithm is different from the normal k-means algorithms. To be specific, Mahalanobis distance is used in BFR to measure the distance between a point p and a centroid of a cluster c. But, the process of BFR still follows the similar methods of assigning points to clusters.

5.2 The CURE Algorithm

The CURE algorithm also handles large dataset. It assumes Euclidean space, but it does not require any assumptions on the shape of the cluster[4]. The process of a CURE algorithm is shown as following:

1. take a small sample of the data and cluster it in main memory

2. select a small set of point as representative points
3. move each of the these points a fixed fractions of the distance between its location and the centroid of its cluster
4. Merge two clusters if they have a pair of representative points that are sufficiently close
5. For each point p that is stored in the secondary storage, compare it to the representative points and assign it to the closest representative point.

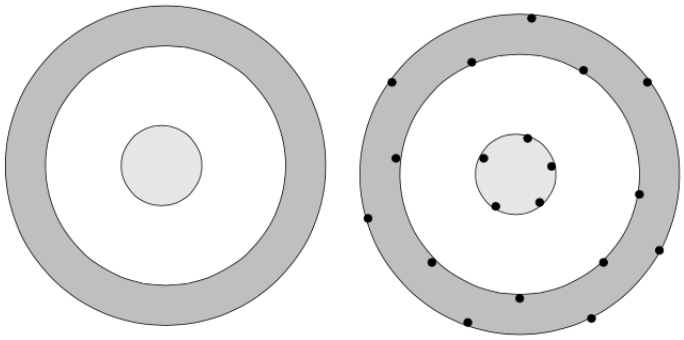

Figure 6: Two Clusters with Picked Representative Points

6 Clustering in Non-Euclidean Spaces: GRGPF Algorithm

Under Non-Euclidean Space, "distance" between each "point" is computed using a different mechanism. GRGPF algorithm is often used under Non-Euclidean space. It uses a tree to organize the clusters hierarchically[6]. By doing this, a new points can be assigned to the appropriate cluster by passing it down the tree. Leaves of the tree hold summaries of some clusters reachable through that node.

The GRGPF algorithm uses the similar ideas like CURE to represent cluster by sample points in the main memory. Then it load the points in the secondary storage and process them by passing each on to the cluster "tree". For each points p, it uses follows the process:

1. Starting at the root, it compares the point p to the children of the root.
2. Choose the one, denoted as q, that is the closest to p
3. Compare the children of q to p, repeat this cursively until reaches the leaves
4. At leaf node, choose the cluster that is closest to p

A Reference

1. Rajaraman, A., & Ullman, J. (2012). Mining of massive datasets. New York, N.Y.: Cambridge University Press.

2. Russell, S., & Norvig, P. (2010). Artificial intelligence: A modern approach (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

3. P.S. Bradley, U.M. Fayyad, and C. Reina, "Scaling clustering algorithms to large databases"

4. S.Guha, R.Rastogi, and K.Shim, "CURE: An efficient clustering algorithm for large databases"

5. Shwartz, S., & David, S. (2014). Understanding machine learning. New York, NY: Cambridge University Press.

6. V. Ganti, R. Ramakrishnan, J. Gehrke, A.L. Powell, and J.C. French:, "Clustering large datasets in arbitrary metric spaces"

7. B. Babcock, M. Datar, R. Motwani, and L. OCallaghan, Maintaining variance and k-medians over data stream windows

8. Ackerman, M. and Ben-David, S., Measures of clustering quality: A working set of axioms for clustering