# Dimensionality Reduction for Clustering

**Yunshuang Tao; Ningxiao Tang**

**Github Link: https://github.com/Ningxiao-Tang/BigDataProj**

## 1. Problem Definition

In the real world, we sometimes need to assign items into different clusters. These items are considered as points and sometimes points can have a very high dimension. Clustering high-dimensional points can take up a considerable amount of time and space. If we just want to do the clustering and still maintain the cost, dimensionality reduction can be performed to reduce workload.

## 2. Methodology

In our project, we first perform dimensionality reduction on a dataset and then use k-means++ clustering to assign points into different clusters. Compute the cost of each dimension with the original dimension.

### 2.1. Dimensionality Reduction (FastMap)

FastMap Algorithm:

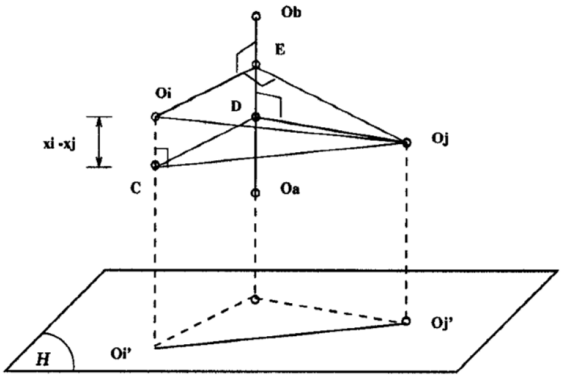
–Original dimension:N reduced to dimension k

–Initialize column number to 0 (In practice it should be -1)

–Compute distance matrix

–Repeat the following steps for k times

- –Increase column number by 1
- –Choose pivot object Oa and Ob and record their ids
- –Project objects on a plain that is perpendicular to the line connecting Oa and Ob
- –Update distance matrix
- –Decrease k by 1




Choose-Distant-Object Algorithm

–Choose arbitrarily an object and let it be the second pivot object Ob

–Let Oa be the object that is farthest apart from Ob

–Let Ob be the object that is farthest apart from Oa

–Report the object Oa and Ob as the desired pair of objects

### 2.2. Clustering (k-means++)

1. Randomly select the first means of size N
2. Select k centers from means
3. Update cluster in while loop  until the euclidean distance between new centroids and old centroids becomes zero
    a. Assign each value to its closest cluster
    b. Sorting the old centroid values
    c. Finding the new centroid by taking the average value

## 3. Experiments

Original dataset: http://archive.ics.uci.edu/ml/machine-learning-databases/00401/

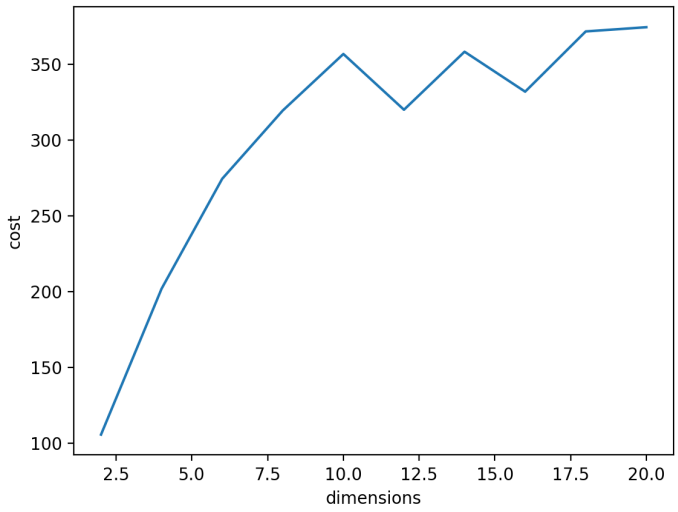
A medical dataset with more than 800 items and each item has more than ten thousand fields.

### 3.1. Experiment 1

Dataset: 100 points of 20 dimensions.

Dimensions: Decreased from 20 to 2 with a step of 2.

Result:

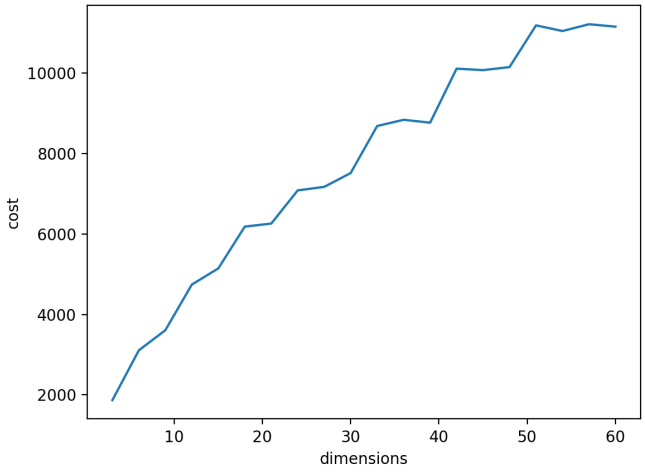



### 3.2. Experiment 2

Dataset: 800 points of 60 dimensions

Dimensions: Reduced from 60 to 3 with a step of 3

Result:

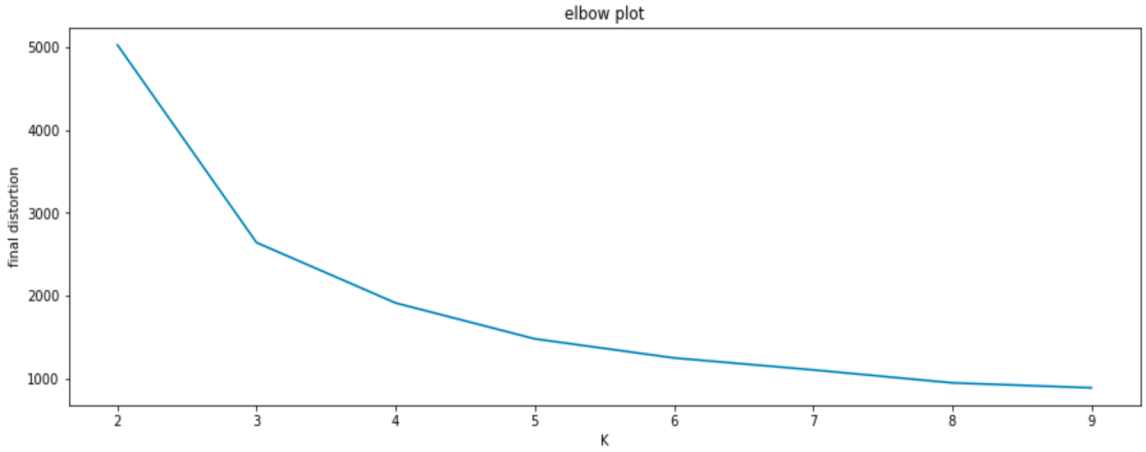



### 3.3. Experiment 3

Dataset: an image with dimensions of 6000* 3376

Clustering: segment the image with center from 2 to 9

Result:




## 4. Discussion

According to the experiments above, we can see that if we want to reduce points dimension while still keeping the cost of clustering, we can reduce points dimension to no smaller than 90% percent of its original dimension. When dimension keeps decreasing, cost may not be kept since we use euclidean distance. However, if we just want to have similar results of clusters, we can reduce points dimension on a larger scale.

## 5. References

Christos Faloutsos and King-Ip Lin. 1995. FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. SIGMOD Rec. 24, 2 (May 1995), 163–174. DOI:https://doi.org/10.1145/568271.223812