Machine Learning CS60050
Instructor: Dr. Aritra Hazra
Department of Computer Science and Engineering Indian Institute of Technology,Kharagpur

Scribed by: Nabajyoti Das (20CS60R01), Braj Bihari(20CS60R02)

Notes for the class on - 10th March 2021

1 Supervised and Unsupervised Learning

We have two methods to learn machine learning but both technique is applied on different data sets.

1.1 Supervised Machine Learning:

In supervised learning method model are trained using labeled data. Where model needs to find the mapping function to map the input variable (X) with the output variable (Y).
mapping function:

$$Y = f(X)$$

it is similar as a student learn in the presence of a teacher same thing can also be applied here like Supervised learning needs supervision to train the model.it have basically two types of problems:
1. Classification
2. Regression

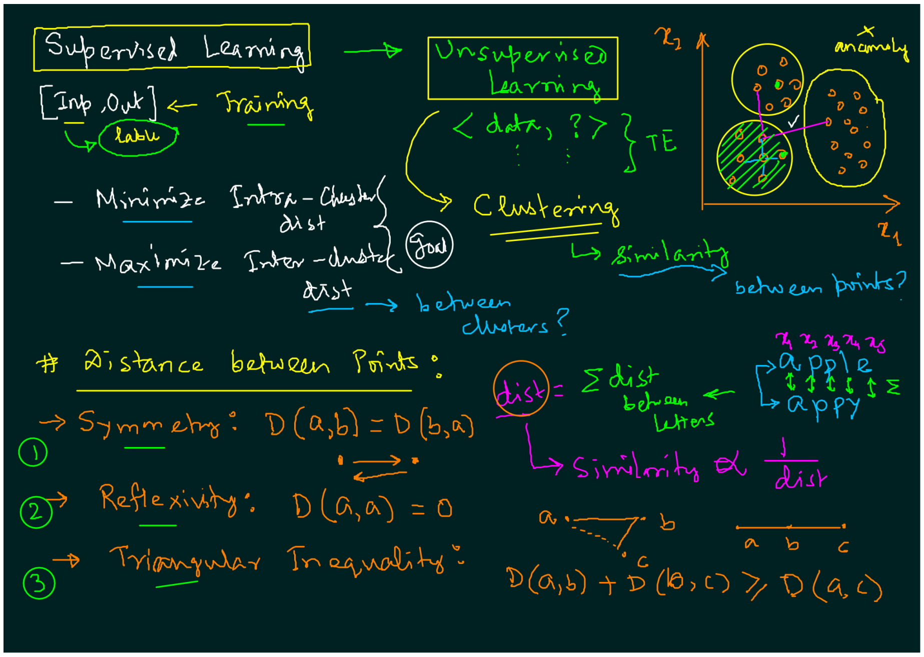
1.2 Unsupervised Machine Learning:

In this method also pattern inferred from the unlabeled input data. The main goal is to we need to find the structure and pattern from input data. unlike supervised learning it doesn't need any supervision it find pattern by itself only. no supervision is required. Unsupervised Machine Learning:

2 Clustering

In clustering we basically divide the point in different group like the we have similar point in one group and dissimilar point in other group. we grouping points on the basic of similarity and dissimilarity.

it is kind of unsupervised learning method. here we use the reference from the input data sets points and finds some meaningful structure,explanatory underlying processes, generative features, and groupings inherent in a set of examples

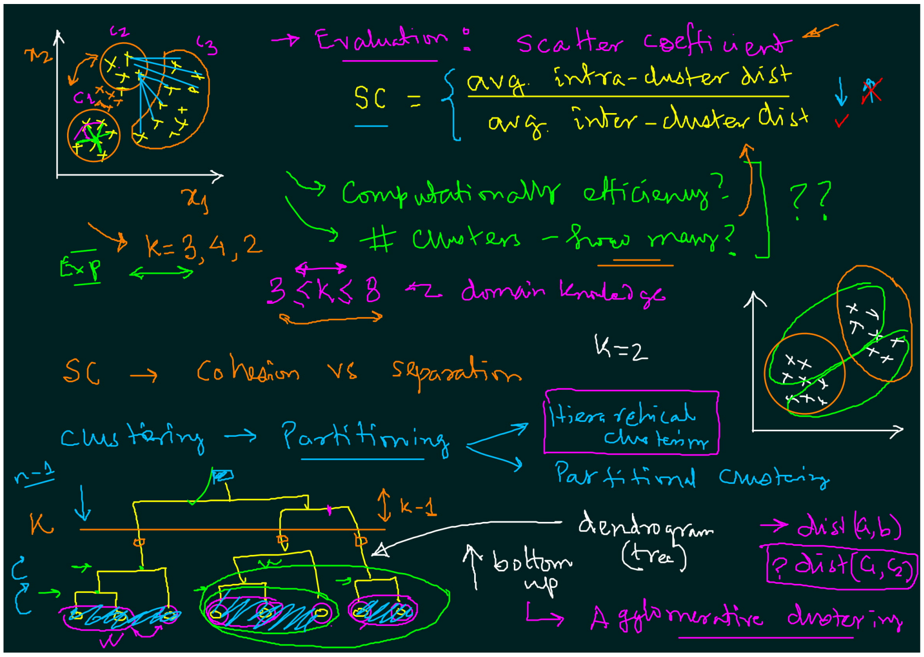


Supervised Learning
Unsupervised Learning
[Inp, Out] ← Training
label
< data, ? > } TE
Clustering
↳ similarity
between points?
x_2
x_1
anomaly
− Minimize Intra-Cluster dist
− Maximize Inter-cluster dist
Goal
→ between clusters?
# distance between Points :
→ Symmetry : D(a,b) = D(b,a)
①
② → Reflexivity : D(a,a) = 0
③ → Triangular Inequality :
dist = Σ dist between letters
x_1 x_2 x_3 x_4 x_5
apple
appy
Σ
↳ Similarity ∝ 1/dist
a b c
a b c
D(a,b) + D(b,c) ≥ D(a,c)

Figure 1: Reference - 1

For example in below figure we have some points. these points can be classified into different groups according similarity.
Our goal is to find the minimize intra cluster distance and maximize inter cluster distance.
There are some property based on distance:

1. Symmetry :
$$D(a,b) = D(b,a)$$
2. Reflexivity :
$$D(a,a) = 0$$
3. Triangular inequality :
$$D(a,b) + D(b,c) > D(a,c)$$



→ Evaluation : scatter coefficient
SC = avg. intra-cluster dist / avg. inter-cluster dist
c1
c2
c3
x2
x1
Computationally efficiency?
# clusters - how many?
??
Exp
k=3,4,2
3≤k≤8 ← domain knowledge
k=2
SC → Cohesion vs separation
clustering → Partitioning
Hierarchical clustering
Partitional clustering
n-1
K
k-1
dendrogram (tree)
↑ bottom up
→ dist(a,b)
? dist(c1,c2)
↳ Agglomerative clustering

Figure 2: Reference - 2

In case of co-linear point:

$$D(a,b) + D(b,c) = D(a,c)$$

Using scatter coefficient we measure the how well we done clustering. scatter coefficient=(average of intra-cluster distance) /(average of inter-cluster distance)
if the lower the scatter coefficient it will be good. and high the scatter coefficient it will bad for clustering. scatter coefficient is the measure between cohesion and separation. Now the question is how much clustering efficiency is required and how many cluster required for perfect clustering.

clustering is basically partitioning of points into different groups. it is basically two types.

1. Hierarchical clustering
2. Partitional clustering

Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

2.1 Agglomerative:

Initially consider every data point as an individual Cluster and at every step, merge the nearest pairs of the cluster. (It is a bottom-up method). At first every data set is considered as individual entity or cluster. At every iteration, the clusters merge with different clusters until one cluster is formed.
Algorithm for Agglomerative Hierarchical Clustering is:
1. Calculate the similarity of one cluster with all the other clusters (calculate proximity matrix)

2. Consider every data point as a individual cluster
3. Merge the clusters which are highly similar or close to each other.
4. Recalculate the proximity matrix for each cluster
5. Repeat Step 3 and 4 until only a single cluster remains.

2.2 Divisive:

We can say that the Divisive Hierarchical clustering is precisely the opposite of the Agglomerative Hierarchical clustering. In Divisive Hierarchical clustering, we take into account all of the data points as a single cluster and in every iteration, we separate the data points from the clusters which aren't comparable. In the end, we are left with N clusters.

2.3 Distance between cluster:

Avg. distance: It is defined as the average of distances between the centroid of the cluster c1 and c2.
Min. distance: It is defined as the minimum distance of two points from c1 and c2.
Max. distance: It is defined as the maximum distance of two points from c1 and c2.
And there are other distances like Cosine distances, Euclidean distances, Manhattan distances as we have seen in K-mean clustering.

2.4 Clustering algorithms:

According to distances measurement we choose in our algorithm, name of our algorithms changes.
Avg. linkage clustering: It follows Avg. distance to calculate between two cluster.
Single linkage clustering: If it is a agglomerative then it follows bottom-up min distance. Similarity of two clusters is based on the two most similar (closest) points in the different clusters. Determined by one pair of points, i.e., by one link in the proximity graph.
Complete linkage clustering: It follows Max. distance to measure distance between two points. Similarity of two clusters is based on the two least similar

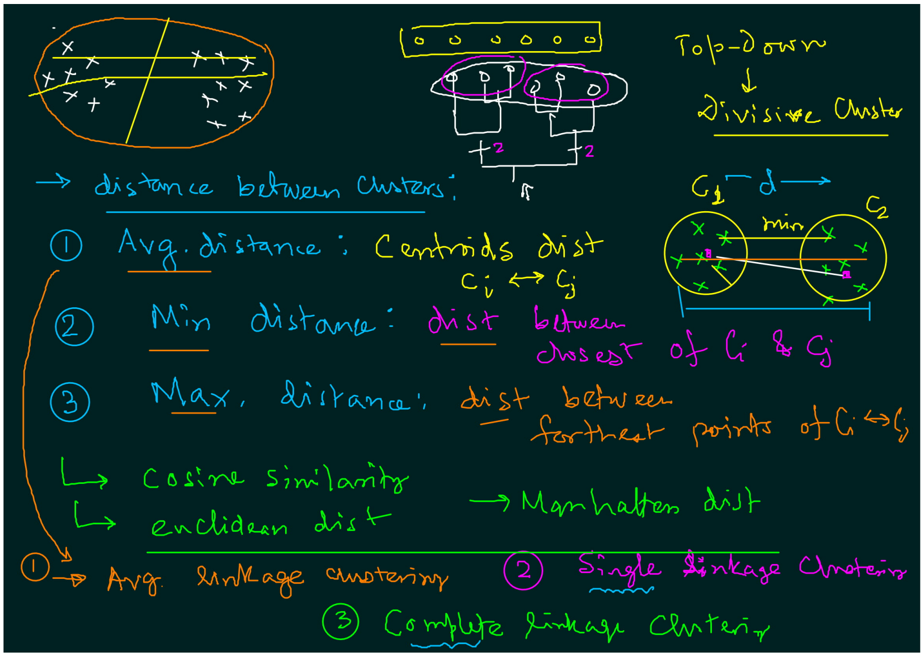

Top-down
↓
Divisive Cluster
2
2
→ distance between Clusters:
C1 d →
min
C2
① Avg. distance : Centroids dist
Ci ↔ Cj
② Min distance : dist between closest of Ci & Cj
③ Max. distance: dist between farthest points of Ci ↔ Cj
↳ cosine similarity
↳ euclidean dist
→ Manhattan dist
① → Avg. linkage clustering
② Single linkage Clustering
③ Complete linkage Clustering

Figure 3: Reference - 3

(most distant) points in the different clusters. Determined by all pairs of points in the two clusters.

2.5 Merits and Demerit:

One of the merits of above algorithm is it can form arbitrary shaped clusters. So it is easy for this algorithm to find any kind of clusters. Computationally inefficiency and noise in data are two massive demerit of this algorithm.

3 Partitional Clustering

To overcome above limitation we choose most widely used partitional clustering algorithm i.e. K-means algorithm. This algorithm is computationally very efficient.

3.1 K-means algorithm:

1. Let us assume we have K number of buckets where K can be any chosen predefined number and there are n number of points.

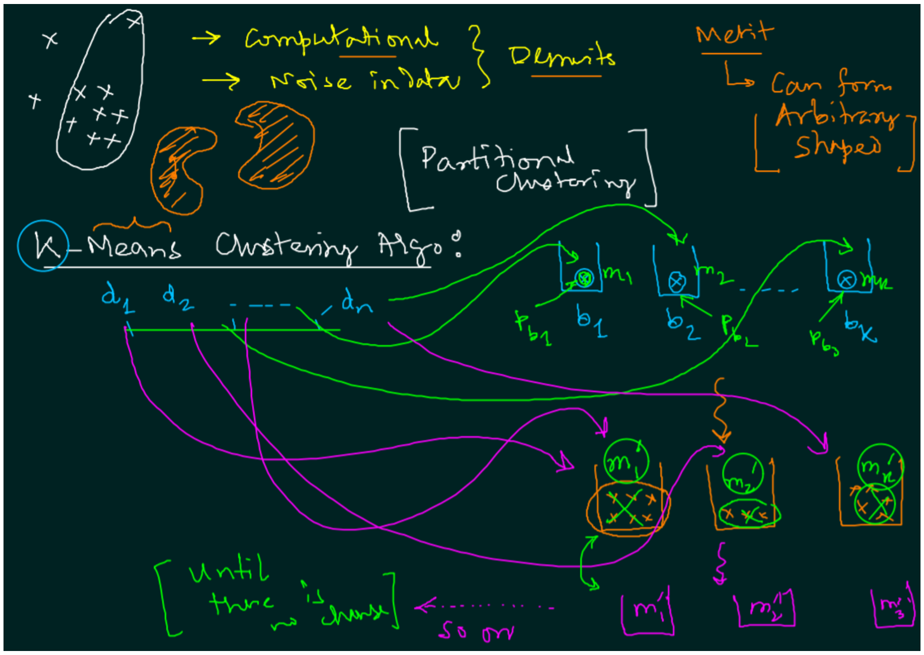


Figure 4: Reference - 4

2. Then choose any arbitrary K number of points from n and put it into buckets.
3. Then for each points in n try to compute distance w.r.t. pivot point in all the bucket then accordingly push that data point into the bucket with closest pivot point.
4. Once all the data points are pushed into the bucket then compute mean or avg. for all the buckets using the points inside the buckets and update the pivot point.
5. Then repeat step 3 and 4 until there is no change between previous and current data set.

3.2 Example:

For K=2
Let us consider data point (0,0),(0,1),(1,0),(1,3),(2,3),(2,2). and two(K=2) buckets B1,B2.
Now arbitrarily chose two points (0,0),(0,1) and put them into two buckets.
Iteration:1
B1=(0,0)
B2=(0,1)
For rest data points pick them one by one and put them into closets bucket.
B1=(0,0),(1,0)
B2=(0,1),(1,3),(2,3),(2,2)

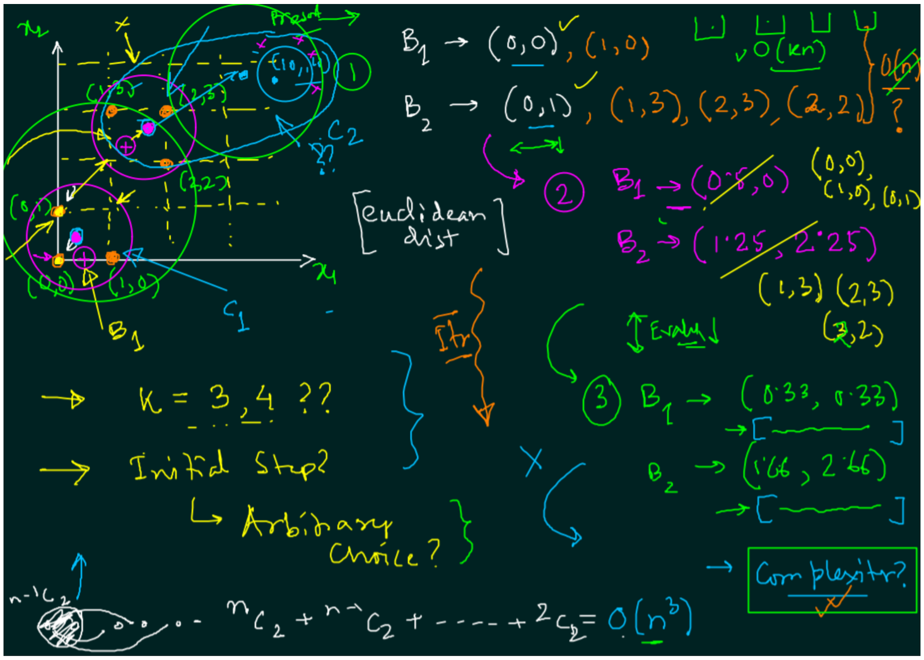


Figure 5: Reference - 5

Iteration:2
Now calculate mean point of both the bucket:
for B1:(0.5,0)
for B2:(1.25,2.25)
Find the closest bucket for a data point w.r.t. new pivot points. New points in the bucket will be:
B1=(0,0),(1,0),(0,1)
B2=(1,3),(2,3),(2,2)
Iteration:3
Now calculate mean point of both the bucket:
for B1:(0.33,0.33)
for B2:(1.66,2.66)
Again find the closest bucket for a data point w.r.t. new pivot points. New points in the bucket will be:
B1=(0,0),(1,0),(0,1)
B2=(1,3),(2,3),(2,2)
We can see data set in bucket for previous and current iteration are same. We can stop our algorithm and return B1,B2.