→ "UNSUPERVISED - LEARNING"

↳ "Clustering" :⇒ (Another set of problems in ML)

We have already learnt about ["classification"] & ["Regression"] problems.

In classification & regression, we are given a dataset "D" comprising of $\{x_i's$ & $y_i's\}$

In "classification" "y_i" are the "class labels"

Where as in case of "regression" "y_i" are real values.

In classification & Regression, given a value of ["x_i"] we want to predict the value of ["y_i"]

We basically trying to find a function

$y = f(x)$ in both "Classification & Regression"

Note:⇒ The only difference b/w "classification & Regression" is, In case of classification $y_i \in$ {set of values (a finite set)} whereas in case of regression $y_i \in$ real values.

Eg:⇒ Classification $y_i \in \{0,1\}$ → 2 class classification

Regression $y_i \in \mathbb{R}$

Now let us move to clustering & see what it is?

In case of clustering, we are just given a bunch of datapoints without any class labels ($y_i's$)

Scanned with CamScanner

In Clustering there are no "yi's"

Where as in case of regression & classification, we are given ["xi's"] & we want to predict the corresponding ["yi's"].

"$y = f(x)$"

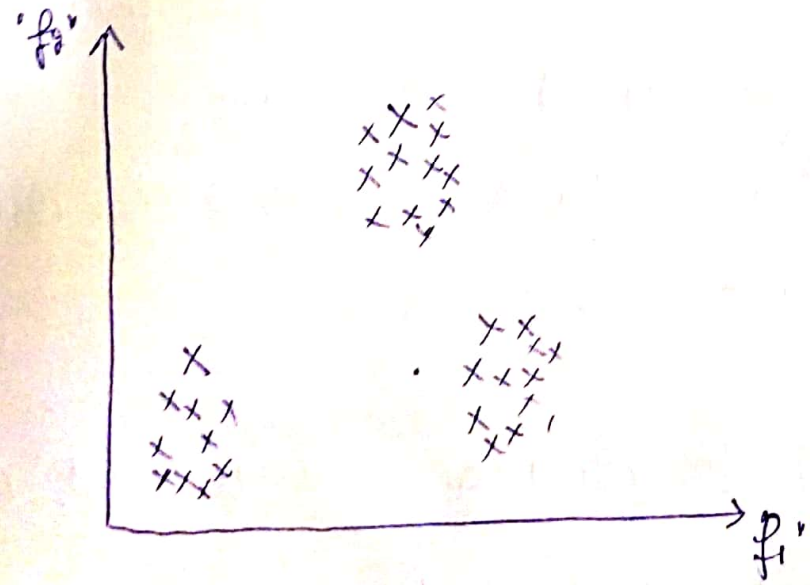
In all the techniques like "logistic regression", "linear regression", "decision trees", "Naive Bayes", "random forest" etc, we are basically trying to find out this function $\boxed{y = f(x)}$. which maps "xi's" with "yi's".

In clustering, we don't have any such "yi's"; we are just given xi's & our task is to group or cluster similar datapoints.

Task in clustering :⇒ To group or cluster "similar datapoints".

Let's try to understand it geometrically :-

Suppose we have two features f_1 & f_2 & we have a bunch of points as shown below —

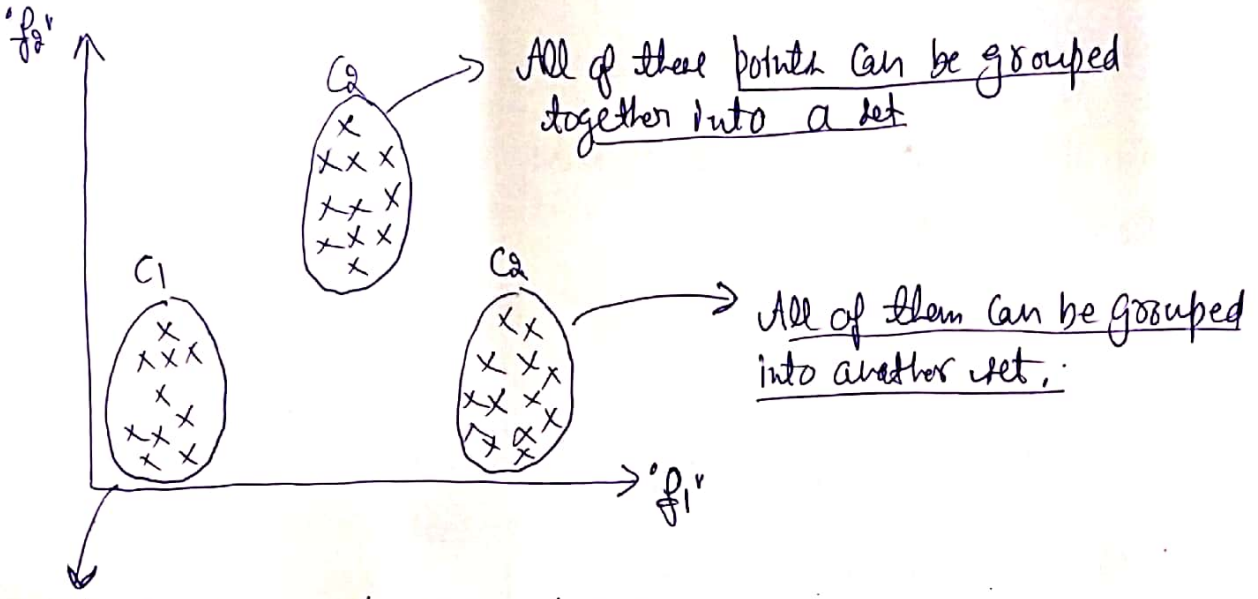


Here we don't have yi's anymore, we are simply given a bunch of points.

Scanned with CamScanner

The task of clustering is to group "similar" points.

Geometrically speaking, if I say group similar points given this visual data. then the groups would look something like this.



In this case, we can say that these points can be grouped together into three clusters {C1, C2 & C3} & the definition of similarity here is, points are closed together (within a cluster) & they are farther away from other points: (from other clusters).

The two things we are looking for are :>

a) Points in a cluster are closed together.

b) Points in different clusters are far away from each other.

Note:> The word "similar" is very very problem specific..

The task of clustering from a geometrical perspective is to group "similar points".

Scanned with CamScanner

In clustering, from a mathematical stand point, all we are given is $D = \{x_i\}$ & no y_i's are available.

Now, the big question here is, how do we measure, how well clustering algorithm performs.

In case of classification & regression, we have a bunch of metrics (performance metrics)

- ↳ AUC (Area under the curve)
- ↳ Precision
- ↳ Recall etc.

(All these metrics require y_i's)

but in case of clustering, we donot have y_i's, so how do we measure the performance of a clustering algorithm. we will see it later.

We will group the points in a cluster. on the basis of two things & this is the basis of clustering for most of the algos :->

a) Points in a cluster are close to each other.
b) Points in different clusters are far away from each other.

↳ This is the intuition behind clustering algorithms

There are three basic techniques that we will see :

1) K-Means & its variations
2) Hierarchical clustering.
3) DBSCAN

} Different clustering algorithms

Scanned with CamScanner

→ "Unsupervised Learning" :⇒

Clustering is often refered to as unsupervised learning.

Both classification & regression are called supervised learning algorithms or schemes. because in both these cases, we are given {xi's & yi's} & using the {xi's & yi's} i.e the training data, we are trying to find out a function $f(x) = y$. So, here we have y_i, which is helping or supervising us to find the function gracefully, whereas in case of clustering. we don't have y_i's available, so we don't have any y_i's to supervise our learning & hence it is often referred to as "Unsupervised Learning"

Apart from supervised & Unsupervised learning,

we have an area in ML called 'Semi-Supervised Learning"

In this we have a big dataset 'D' which is basically a union of D_1 & D_2

$$"D = D_1 \cup D_2"$$

such that for 'D_1', we have both {x_i's & y_i's}

& for 'D_2', we only have {x_i's}

& typically $\|D_1\| << \|D_2\|$ {The size of 'D_1' is much smaller than the size of 'D_2'}, which basically means that, we have a small portion of data with labels & a large portion of data without labels. This happens when the cost of obtaining labels, i.e. 'y_i' is very expensive.

PTO

Scanned with CamScanner

It is called semi-supervised because on a small subset of data, we have y_i's to supervise our learning, but there is also some data where we don't have it. So, it is between "supervised" & "unsupervised" & hence it is referred to as "semi-supervised".

→ "Metrics of Clustering" :-

The dataset given for clustering comprises of just ["x_i's"] & no ["y_i's"].

Now the question here is, how do we measure, how good is clustering, or what are the performance metrics, which measures how good clustering is. It is a very very important problem, because for classification & regression, we have seen a bunch of metrics & all of them use "y_i's".

y_i's → "Class-labels" in case of Classification

→ "Regression values" in case of Regression.

↓

These are often referred to as ground truths

because 'y_i' is the truth, that we are already given & our job is, given an "x", we need to determine "y"

$$f(x) = y$$

↙

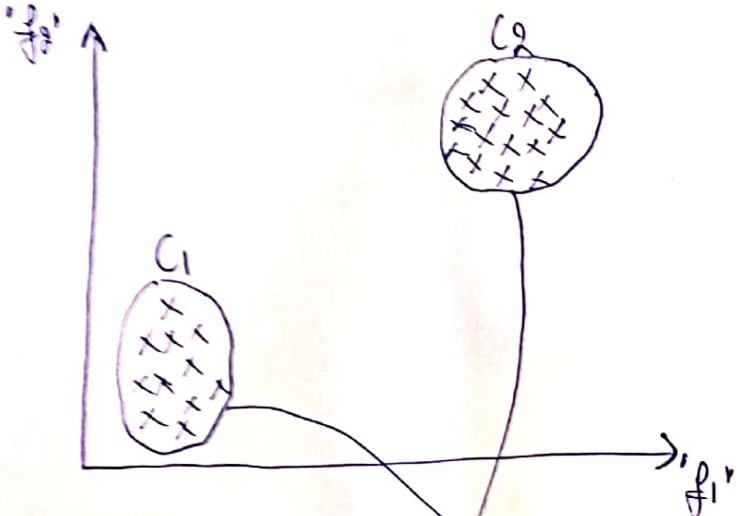
This is what whole "classification" & "Regression" is all about.

When we come to clustering, all we are given is x_i's & no ground truths.

Scanned with CamScanner

Now, before going into the measures, let's first understand the geometry. to get an intuition about, what is a good clustering result.?

Let's assume we have a cluster of points as shown, & remember we do not have any class labels here :>



Suppose, we have this data set $D = \{x_i\}$ where each 'x_i' belongs to $\mathbb{R}^2$. $[x_i \in \mathbb{R}^2]$ & when we do a scatter plot this is what we get. as shown above.

Now if I say a 5 year old kid to group these points into two clusters. He will say that all these points are one cluster & all these points belong to another cluster.

Let's call first group as cluster 'C_1' & second group as cluster 'C_2'

There are two terms here :-

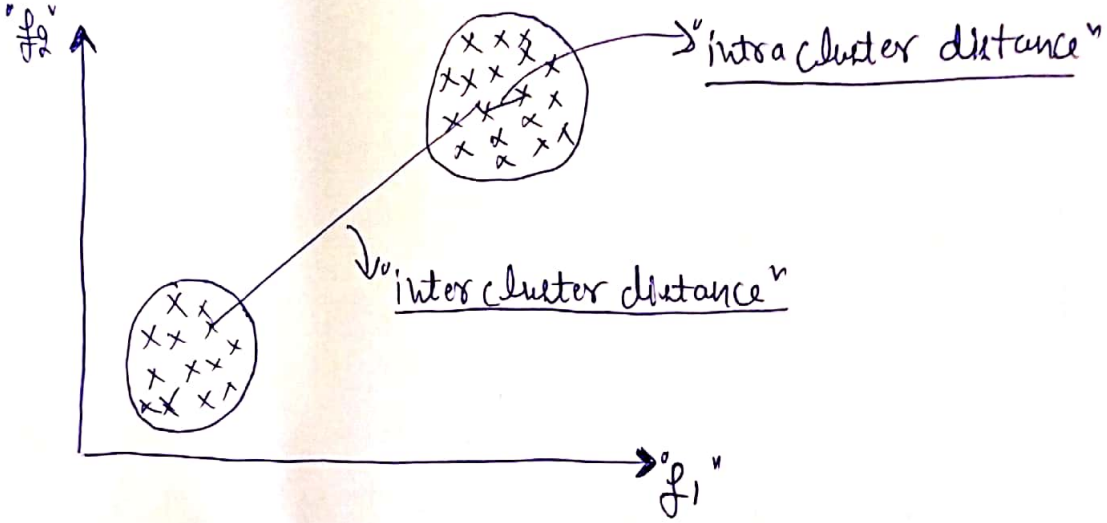
1) 'Intra-cluster' 2) 'Inter-cluster'

↓ ↓

[It means within a cluster.] [It means across, or between clusters.]

Scanned with CamScanner

So, one thing that has been done is, if we want to group the points into clusters. then the "intra-cluster" distance is kept small, which means if we take any two points within a cluster & if we look at the distance b/w those points, the intra-cluster distance is kept small. while the inter cluster distance (the distance between points belonging to two diff. clusters) is kept large.



This whole thing about "intra-cluster" being small and "inter-cluster distance" being large is the basis of how we measure the clustering effectiveness.

In an ideal world, we want our "inter-cluster distances" to be very high & "intra-cluster distances" to be very low.

There is one metric called the "Dunn" index.

It is often referred to as:

"D"

Suppose, we have "K-clusters", {C1, C2, C3, ---Ci, Cj, Ck}

Then Dunn index is defined as :-

Scanned with CamScanner

$$D = \frac{\max_{i,j} d(i,j)}{\max_{k} d'(k)}$$

Note :→ "d" is different from "d'".

$d(i,j)$ is the distance between cluster["Ci"] & ["Cj"].

[Numerator is basically the maximum 'inter cluster distance']

$d'(k)$ is the intra-cluster distance. It says loop at each of the clusters $\{C_1, C_2, \ldots\}$ & find the cluster 'K' which has the maximum "intra-cluster distance"

for Dunn index to be high, the 'inter-cluster distance" should be high & 'intra-cluster distance" should be low.

Note :→ If Dunn index is high, it implies very good clustering.

Let's now understand how to measure these distances.

Let's assume we have a '2d data'. Let's keep fewer points, so that it's easy for us to understand.

Here we have only two clusters.

Scanned with CamScanner

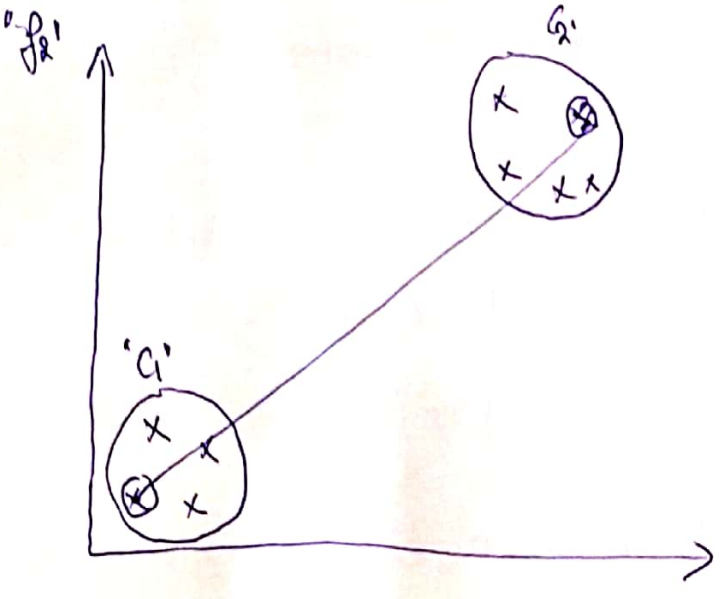
In Dunn index

$d(i,j)$ = distance between Ci's & Cj's farthest points

Here we basically take every point in cluster ["Ci"] & measure the distance of that from every other point in cluster ["Cj"]. We keep doing this for every pair of points. where one point is from cluster ["Ci"] & other point is from cluster ["Cj"].

$d_1, d_2, d_3, \ldots\ldots\ldots$

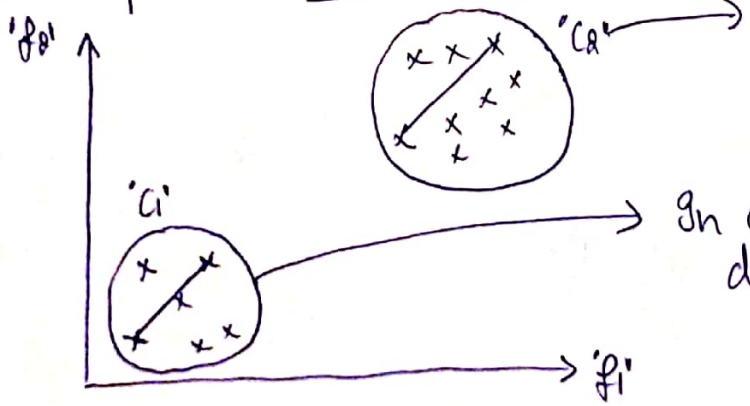
& from these distances take the maximum one.



So, d(i,j) in the numerator is the distance b/w clusters C1 & C2 such that, it is the distance b/w two farthest points, such that the first point lies in 'C1' & second point lies in 'C2'.

This is about Numerator.

Let's talk about denominator:-



This is the farthest distance in "C2"

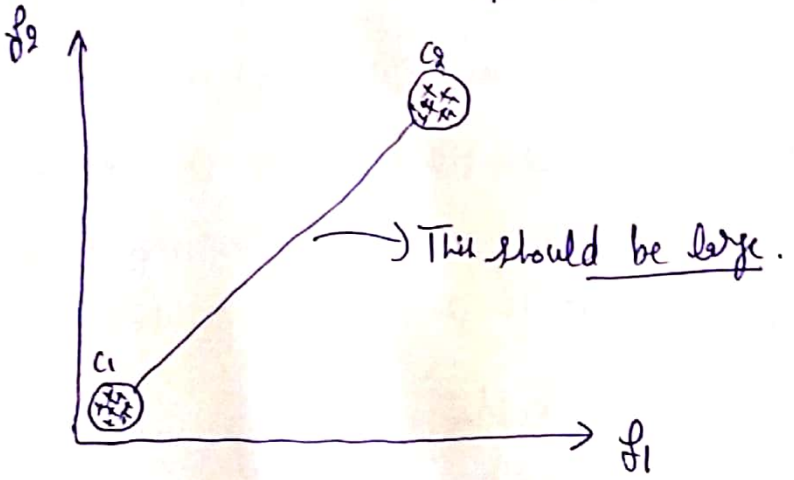
In cluster "C1" this is the farthest distance.

Scanned with CamScanner

So, denominator says, for every cluster, take the farthest distance & take the maximum among them. Let's call the farthest distance in cluster 'C1' as d'_1 & farthest distance in cluster 'C2' as d'_2 then take max $\{d'_1, d'_2\}$. That's what the denominator is.

This is one metric of clustering. So Ideally we want "Intra-cluster distance" to be as small as possible and "Inter-cluster distance" to be as large as possible, for Dunn Index to be large.

So, Ideal clusters will look like this :-



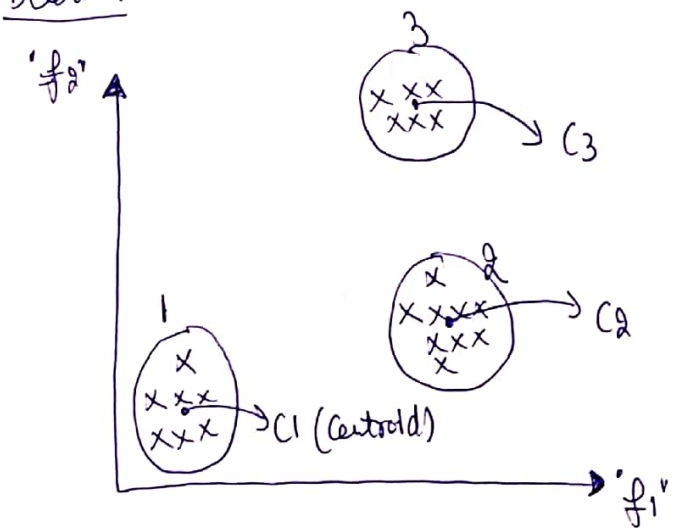
→ "K-Means Clustering"

It is one of the most popular & also very very simple clustering algorithm. Before we understand it, let's look at the geometric intuition behind 'K-Means'

There are various variants of K-Means like K-medoids. Kmeans++ etc. But first let's understand, what does K-mean actually mean from a geometric perspective.

Scanned with CamScanner

Let's say we have a bunch of points as shown below :-



['K' in "K-means" means # of clusters.]

Given this dataset, suppose we want to do 'K-means' clustering with 'K=3". (K is # of clusters) it is a hyperparameter here, & it can be determined using an idea called 'cross-validation".

So, what K-means with "K=3" effectively does is, it groups these points into three clusters. For every cluster, this algorithm assigns something called a centroid, which is the central point.

C_1, C_2 & C_3 are centroids

& for each of these three clusters, we have a set

S_1, S_2 & S_3

Note :- Set of points which are present in a cluster is called a set basically.

C_1, C_2, C_3 : centroids

S_1, S_2, S_3 : sets.

such that

$$S_1 \cup S_2 \cup S_3 = D$$

& no point should belong to two different sets.

i.e $S_1 \cap S_2 = \phi$, $S_2 \cap S_3 = \phi$ & $S_1 \cap S_3 = \phi$.

Scanned with CamScanner

This means, there should be no point which belongs to more than one set. & every point should belong to one of these clusters.

When you say, you want ["K-clusters"] what you get in K-means is basically ["K-centroids"] & they are referred to as ["C1, C2 - - Ck"] & ["K-sets"] of points referred to ["S1, S2, --- Sk"]

K-clusters mean

K-Centroids : [C1, C2, C3 - - - - Ck]

K-Sets : [S1, S2, S3 - - - - - Sk]

① ② ③ - - - (k) → Clusters

Typically the centroid of any cluster, is the mean of all the points & it can be represented as:-

$$\left[C_i = \frac{1}{n} \sum_{x_j \in S_i} x_j \right] \longrightarrow \text{Mean or central point in } S_i$$

Centroid is basically "geometric mean"

"K-means" is basically centroid based clustering scheme., because here we define each cluster using its centroid.

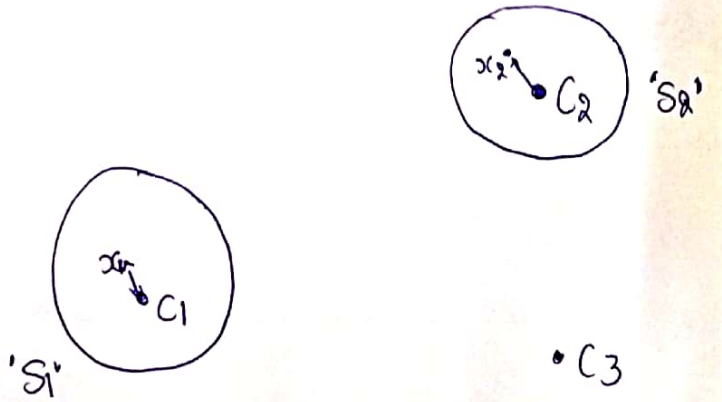
Every point is assigned to a cluster corresponding to the nearest centroid.

This is the idea behind clustering & the big challenge in K-means clustering is, how to find the "K-centroids" ; beoz when we find the "K-centroids", we can easily find the sets.

Scanned with CamScanner

So, once we get the ["K-centroids"], we can easily get ["K-sets"], by just using nearest centroid idea.

Suppose, we have a centroid "C_1", "C_2" & "C_3" as shown below & we have a datapoint "x" Now since "x" is closest to "C_1", we will assign it to set 'S_1'.



similarly if we have a point "x_2", which is closer to "C_2". so we will assign it to set 'S_2'. & so-on.

.——.

→ "K-Means Mathematical formulation & objective function"

Let's now look at the mathematical formulation of K-Means clustering :→

We are given a dataset D comprising of n points

$$D = \{x_1, x_2, x_3 \text{ - - - - } x_n\}$$

our task is to find "K" centroids $\{'C_1, C_2 \text{ - - - } C_K'\}$ & their corresponding sets of points $\{'S_1, S_2, \text{ - - - } S_K'\}$ such that each point in a set like ['S_1'] has the nearest centroid ['C_1'].

P.T.O

such that each point 'x_i' belongs to atleast one cluster (15)
set 'S_j'

$$\forall i \quad x_i \in S_j$$

& given two clusters, their intersection is a null set

$$\forall_{i,j} \quad S_i \cap S_j = \phi$$

Mathematically speaking these are the constraints

$$\begin{cases} \forall i \; x_i \in S_j \\ \forall_{i,j} \; S_i \cap S_j = \phi \end{cases}$$

It says every point should belong to atleast one cluster & between two cluster 'i' & 'j' there should be no common points

Let's write the objective function → squared distance of x from centroid

$$\underset{\substack{c_1, c_2, \ldots c_k \\ (S_1, S_2 \ldots S_k)}}{\arg\min} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - c_i\|^2$$

$i = 1$ ↓ across all the cluster.

s.t $\begin{cases} x \in S_i \\ S_i \cap S_j = \phi \end{cases}$ & Constraints

We need to find '$c_1, c_2 \ldots c_k$'. & once we find the centroids, it is easy to find the sets using proximity idea. i.e (Given a point, assign it to the nearest centroid.)

So, we want to find the 'K-centroids' which minimizes the sum over each cluster.. we want to minimize the distance of points from the centroid..

$\|x - c_i\|^2$ → squared distance of 'x' from centroid.

Scanned with CamScanner

So, what it is telling us is, find the centroids such that, each of the points is assigned to the nearest centroid so that the intra-cluster distance is minimized.

So, intuitively, it says to minimize the ["intra-cluster distance"] it is not saying anything about ["inter-cluster distance"]

$$\underset{c_1, c_2, \ldots c_K}{\arg\min} \sum_{i=1}^{K} \sum_{x \in S_i} \|x - c_i\|^2 \quad \text{s.t} \quad \{ x_i \in S_i, \; S_i \cap S_j = \phi \}$$

- $\sum_{x \in S_i} \|x - c_i\|^2$ → Sum of squared distance from centroid in cluster i (1)
- $\sum_{i=1}^{K}$ → summate over all clusters (2)

→ This mathematical problem is very hard to solve. It is an NP hard problem. Its time complexity is exponential $O(\partial^n)$.

So, we will solve this using approximation algorithms.

Scanned with CamScanner