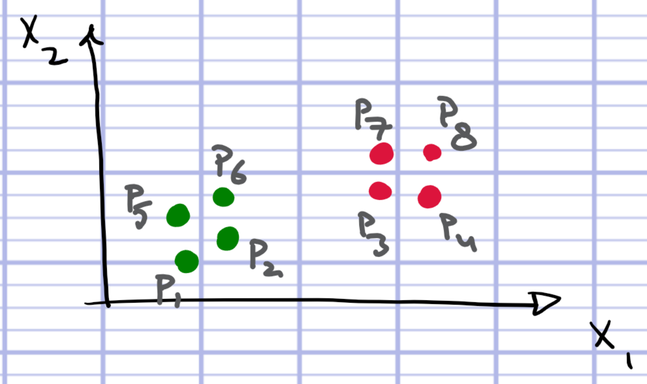
Clustering



The problem is to find a Partitioning of the points (samples) that is ?

Input $X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n_s} \end{bmatrix}$ → Sample 1, 2, ...

& implicitly we assume a distance d

$$d(x_1, x_2)$$

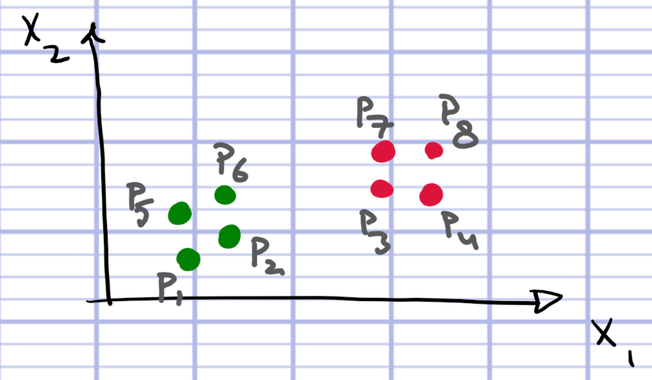
* Sometimes it is ok if it does not satisfy $d(x,y) \leq d(x,z) + d(z,y)$

Output Partioning P such that

if x_i & x_j are in the same cluster, ?

then $P(x_i) = P(x_j)$.

Example



① $P(X_i) = 1$ → All in the same partition.

② $P(X_i) = i$ → Each partitioned individually.

③ $P(X_1) = P(X_2) = P(X_5) = P(X_6) = 1$

$P(X_3) = P(X_4) = P(X_7) = P(X_8) = 2$.

This problem does not have a unique answer.

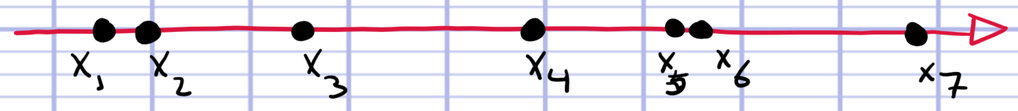
We need to think about what our expectations are.

We will go through some of the algorithms.

Single Linkage

For simplicity take a 1-D example

e.g. X_i is a number :



* SL (m) m is an input.

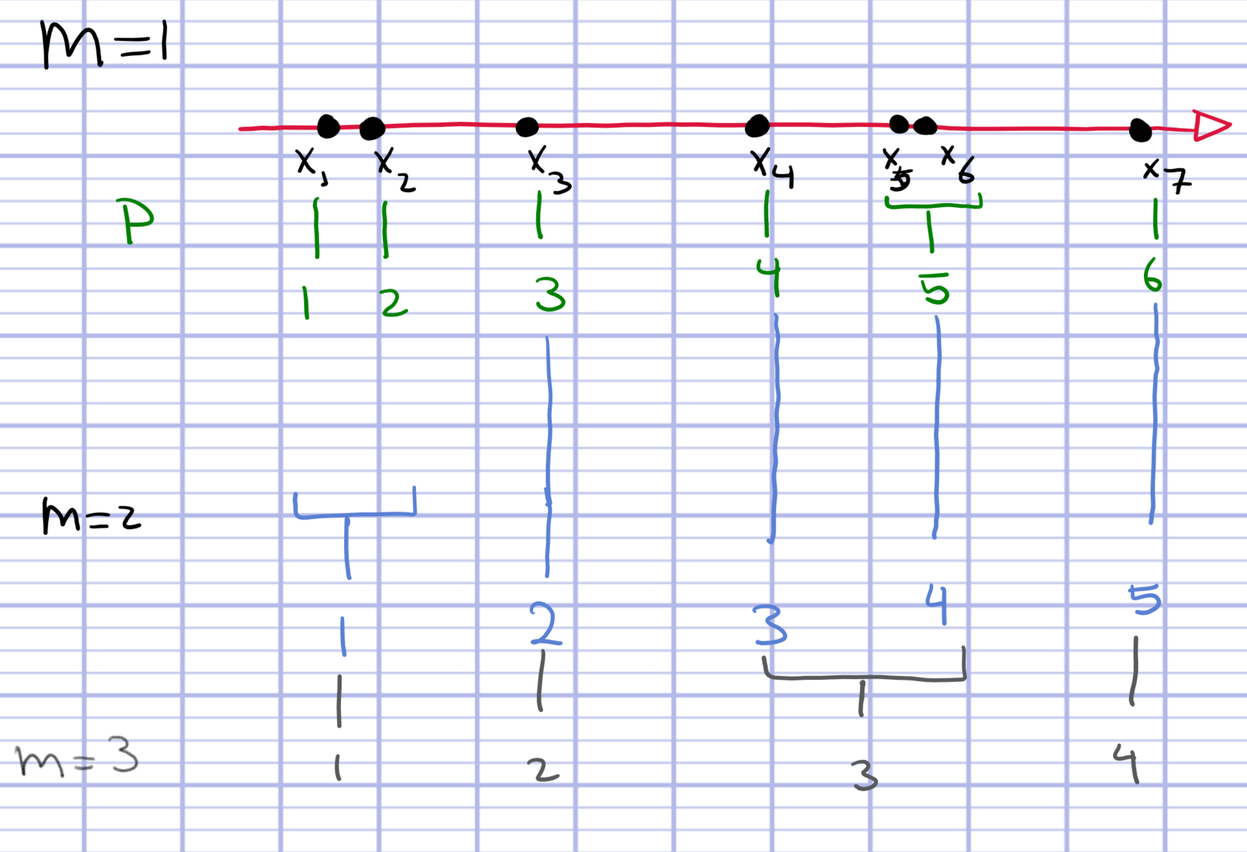
Algorithm:

① $P(X_i) = i$

② Merge the two closest samples into one and $m = m-1$

③ Repeat 2 until $m = 0$

* We need the distance
* For the distance of two clusters, take the closest two points.



⋮

Some remarks

① $m < n_s$

② $m=0 \rightarrow P(x_i) = i$

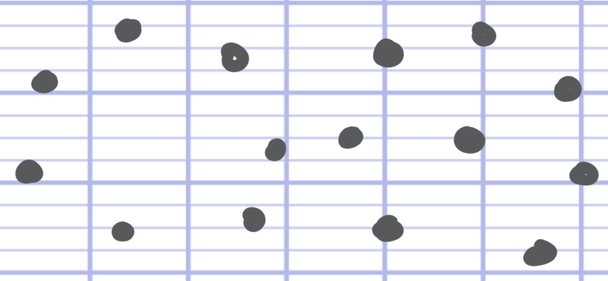
③ $m = n_s - 1 \Rightarrow P(x_i) = 1$

④ # Clusters $= n_s - m$

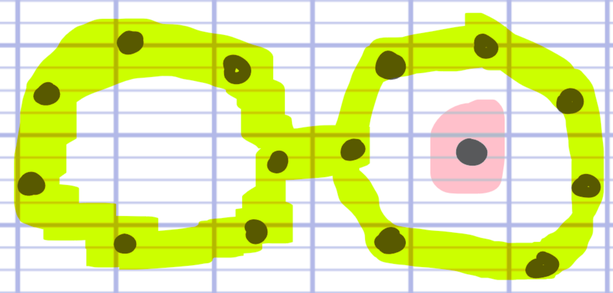
Assignment: What's the scaling of the complexity of the SL?

(Hint: How many times do we need to calculate $d(x_i, x_j)$?)

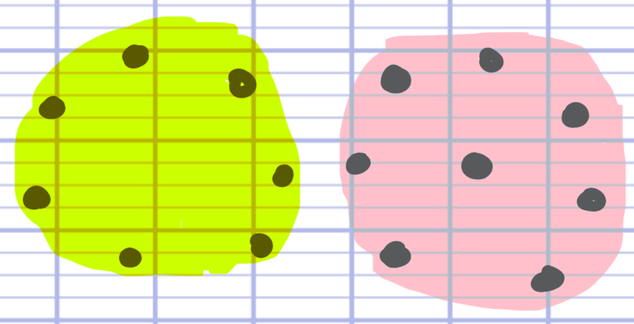
Example 2



Do this for $m = n-2$ ⟶ Want two clusters



whereas it may be more sensible (for some applications) to do it like this



What's the limit of SL?

It does not recognize "centrality".

It only takes into account the distance.

K-Means

Input: $\{x_i\}$ & k → # clusters

Algorithm:

* Pick k points @ random & set the centers

$C_i = x_{\tilde{i}}$ $\quad$ $\tilde{i}$ is the random index.

- ☐ Set clustering as $P(x_i) = \underset{j}{\text{argmin}}\, d(c_j, x_i)$
- ☐ Set the centers as

$$c_i = \frac{1}{n_i} \sum_{P(x_j)=i} x_j$$ → Average of the points in that cluster i.e. center of mass.

n_i = Points in i^{th} cluster.

Repeat until converges.

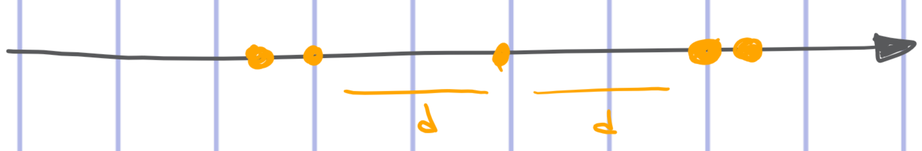
Question: Does it always converge?

Question: Are there local minimum for the optimization?

(hint: Consider )

Question: Complexity ?

Question: What if there eq. distances?



Sometimes we have points that don't fit any cluster or multiple clusters.

Some algorithms flag these as anamoly.

Some algorithms assign a probability to each sample for being in any of the clusters.

↓

"Soft clustering"

[Add DBSCAN] → Anamoly

[Add Gaussian Mixture] → Soft

Assignment

Find a measure that quantifies how well a clustering algorithm is doing?

(Hint: First characterize our expectations from a good clustering)