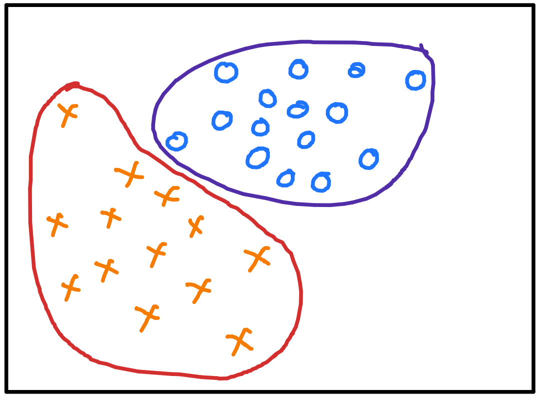
k-means Clustering



- Clustering is an unsupervised ML algorithm

- Idea in clustering
  - Samples within a cluster are similar to each other
  - Samples in different clusters are dissimilar

- We have learned about clustering with GMM using EM algorithm
  - GMM models the cluster probabilistically (soft assignments)

    i.e. $\underbrace{p(\underline{x}^{(i)} \mid y=k)} = \pi_k \mathcal{N}(\underline{x}^{(i)} \mid \underline{\mu}_k, \underline{\Sigma}_k)$

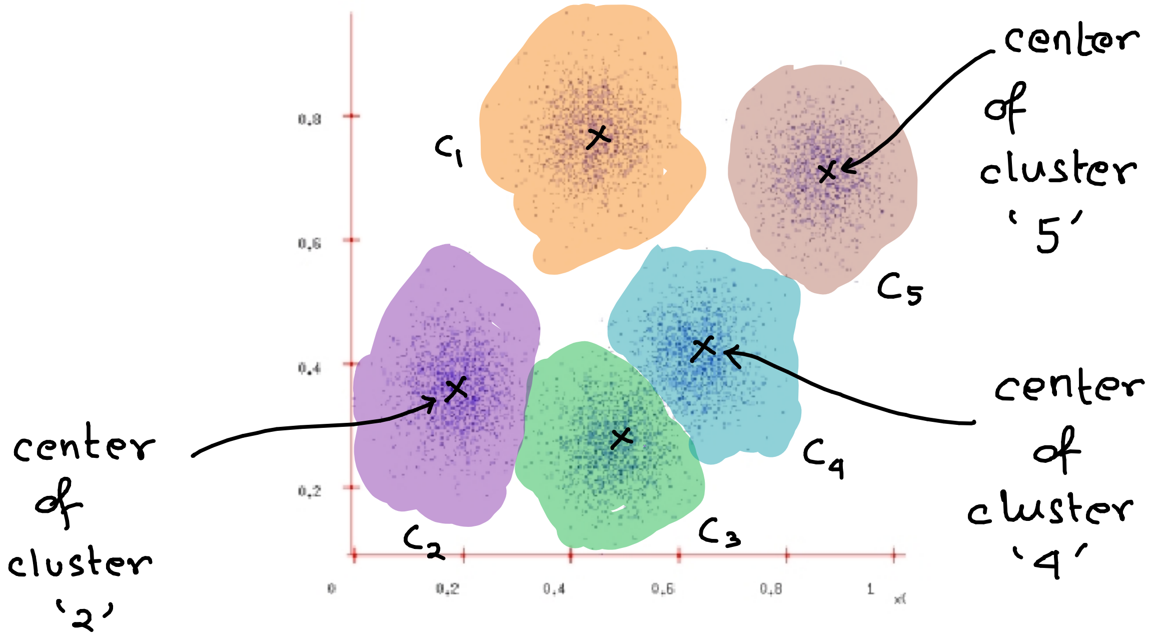
    probability of data point $\underline{x}^{(i)}$ belonging to the 'm' th cluster

- In this lecture, we introduce the k-means clustering algorithm
  - Unlike GMM, in k-means, we do 'hard' cluster assignments and there is no probabilistic model

Intuition of k-means

- k-means assumes that there are 'k' clusters, and each point is close to its cluster center or mean (the average of points in the cluster)
    - If we knew the cluster assignment, we could easily compute the centers
    - If we knew the centers, we could easily compute which points belong to which cluster
    - Chicken and egg problem!

- Heuristically speaking, one could start randomly and alternate between the two!



center of cluster '5'

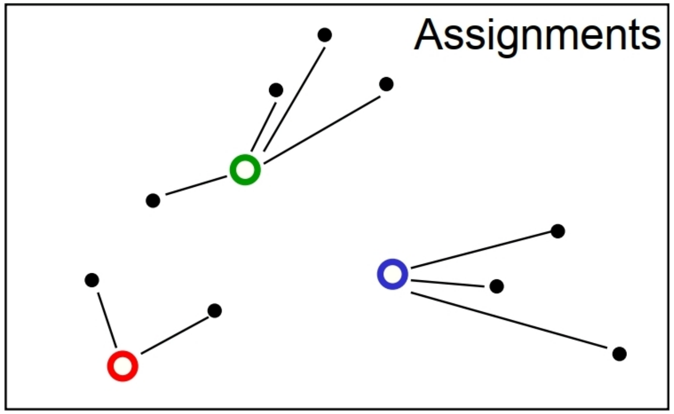
center of cluster '4'

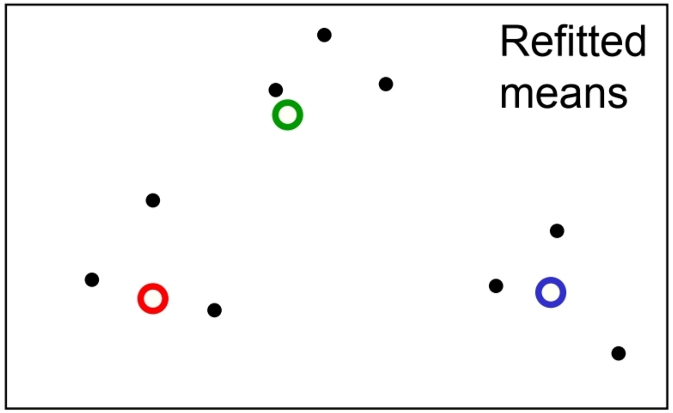
center of cluster '2'

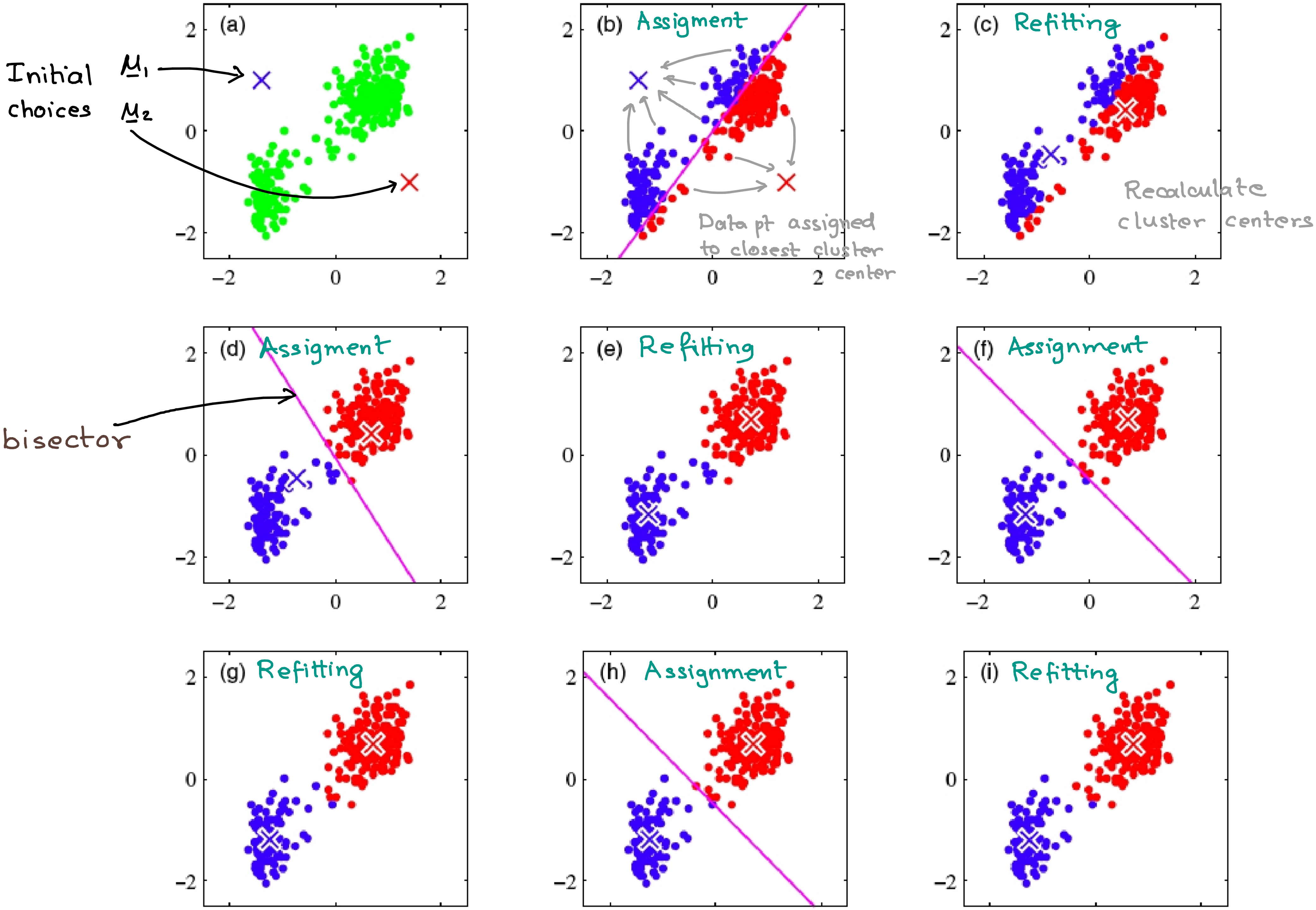
K-means

- Initialization: Randomly initialize cluster centers (or means)

- The algorithm iteratively alternates between two steps:
  - Assignment step: Assign each data point to the closest cluster
  - Refitting step: Move each cluster center to the center of gravity of the data assigned to it







Initial choices

(b) Assigment

Data pt assigned to closest cluster center

(c) Refitting

Recalculate cluster centers

(d) Assigment

bisector

(e) Refitting

(f) Assignment

(g) Refitting

(h) Assignment

(i) Refitting

K-means Objective

What is actually being optimized?

k-means clustering amounts to selecting the 'k' clusters such that the distances of the points to the cluster centers, summed over all data points, is minimized:

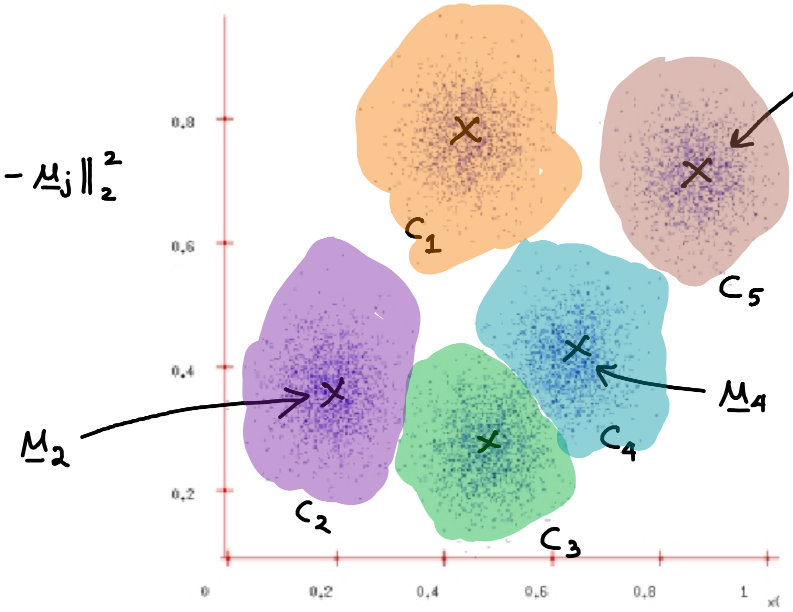
$$\{r_k^{(i)}, \underline{\mu}_k\} = \underset{\{r_k^{(i)}, \underline{\mu}_k\}}{\arg\min} \sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} \| \underline{x}^{(i)} - \underline{\mu}_k \|_2^2$$

mean of all data pts $\underline{x}^{(i)} \in C_m$

center of cluster m

Indicator function

$$r_k^{(i)} = \begin{cases} 1 & \text{if } k = \underset{j}{\operatorname{argmin}} \| \underline{x}^{(j)} - \underline{\mu}_j \|_2^2 \\ 0 & \text{otherwise} \end{cases}$$



$$\underline{\mu}_5 = \frac{1}{|C_5|} \sum_{i=1}^{N} r_5^{(i)} \, \underline{x}^{(i)}$$

# of data pts that belong to cluster '5'

How to optimize?

Optimization problem:

$$\{r_k^{(i)}, \underline{\mu}_k\} = \underset{\{r_k^{(i)}, \underline{\mu}_k\}}{\arg\min} \sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} \| \underline{x}^{(i)} - \underline{\mu}_k \|_2^2$$

- This is a combinatorial optimization which is NP-hard to solve
- An alternating minimization strategy is used to solve the optimization:
  - If we fix the centers $\{\underline{\mu}_k\}$, then we can easily find the optimal assignments $r_k^{(i)}$ for each sample $\underline{x}^{(i)}$

$$\{r_k^{(i)}\} = \underset{\{r_k^{(i)}\}}{\arg\min} \sum_{k=1}^{K} r_k^{(i)} \| \underline{x}^{(i)} - \underline{\mu}_k \|_2^2$$

That is, assign each point to the cluster with the nearest center

e.g. if $\underline{x}^{(i)}$ is assigned to cluster k

$$r_1^{(i)} = 0, \quad r_2^{(i)} = 0, \quad \dots, \quad r_k^{(i)} = 1, \dots, \quad r_K^{(i)} = 0$$

How to optimize?

Optimization problem:

$$\min \sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} \| \underline{x}^{(i)} - \underline{\mu}_k \|_2^2$$

- An alternating minimization strategy is used to solve the optimization:

  - Similarly, if we fix the assignments $r_k^{(i)}$, then we can easily find optimal centers $\underline{\mu}_k$

$$\frac{\partial}{\partial \mu_\ell} \sum_{i=1}^{N} \sum_{k=1}^{K} r_k^{(i)} \| \underline{x}^{(i)} - \underline{\mu}_k \|_2^2 = 0$$

$$\Rightarrow \quad 2 \sum_{i=1}^{N} r_\ell^{(i)} (\underline{x}^{(i)} - \underline{\mu}_\ell) = 0$$

$$\Rightarrow \quad \boxed{\underline{\mu}_\ell = \frac{\sum_{i=1}^{N} r_\ell^{(i)} \underline{x}^{(i)}}{\sum_{i=1}^{N} r_\ell^{(i)}}}$$

K-means algorithm (Lloyd's algorithm)

Data : $\{ \underline{x}^{(i)} \}_{i=1}^{N}$, number of cluster K

Procedure:

- Initialization : Set K cluster means $\underline{\mu}_1, \ldots, \underline{\mu}_K$ to random values
- Repeat until convergence (until assignments do not change)
  - Assignment : Each data point $\underline{x}^{(i)}$ is assigned to nearest center

$$k^{(i)} = \arg\min_{j} \| \underline{x}^{(i)} - \underline{\mu}_j \|$$

and the responsibilities

$$r_k^{(i)} = \mathbb{I}[k^{(i)} = k] \quad \text{for } k = 1, \ldots, K$$

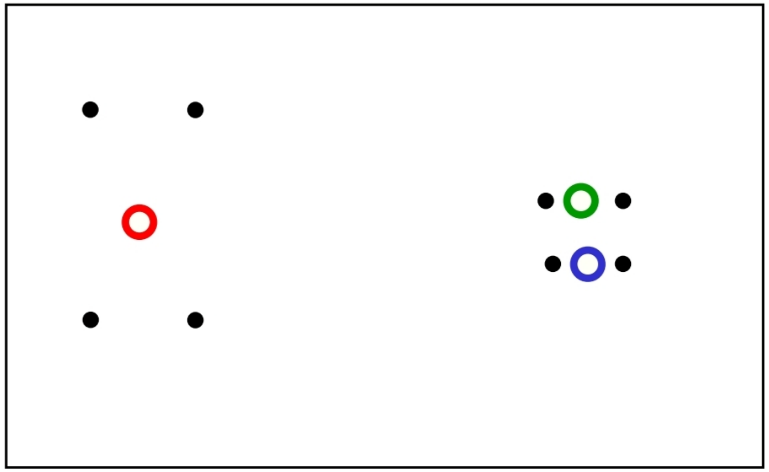
  - Refitting : Each center is set to mean of data assigned to it

$$\underline{\mu}_k = \frac{\sum_i r_k^{(i)} \underline{x}^{(i)}}{\sum_i r_k^{(i)}}$$

Convergence of k-means algorithm

- Similar to the EM algorithm, Lloyd's algorithm converges to a stationary point of the objective function, but is not guaranteed to find the global optimum

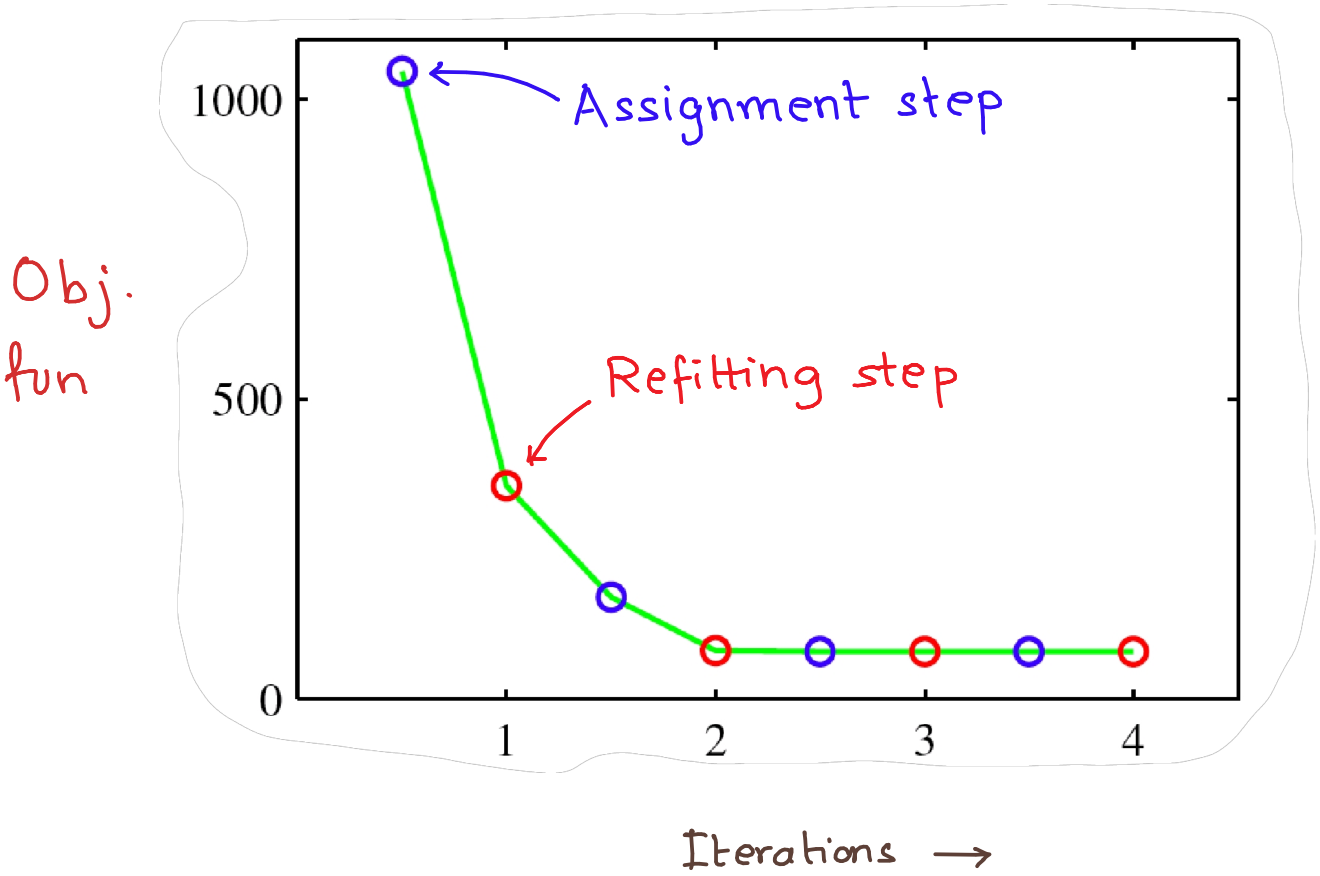
A bad local optimum



- In practice, run it multiple times, each time with a different initialization and pick the result of the run with smallest objective function value

Convergence of k-means algorithm

- Test of convergence : If the assignments do not change in the assignment step, then converged (to at least a local minimum)



k-means should not confused with k-NN

- k-means and k-NN are different, though they have certain similarities

- Both k-means and k-NN use Euclidean distances to define similarities in input space

- Both are sensitive to the normalization of the input values

- However, kNN is a supervised learning method, while k-means is an unsupervised learning method

- The 'k' in the two methods have different meaning

Choosing the number of clusters

- The number of clusters K has to be chosen apriori for both GMM and k-means algorithm for clustering

- Increasing K will reduce training loss (or reduce the objective function)
  - If $K = N$, then each data point will have its own cluster

- Cross-validation techniques are needed to guide selection of K
  - But they need to be adapted to unsupervised setting (There is no new data error E_{new} for clustering)

- For GMM, one can use the likelihood of the validation data to find K

<u>Training set</u> $\{\underline{x}^{(i)}\}_{i=1}^{N}$ <u>Validation set</u> $\{\underline{\tilde{x}}^{(i)}\}_{i=1}^{N_v}$

$K = 1 \longrightarrow M^{(1)}, \hat{\underline{\theta}}^{(1)}$ $\quad P\left(\{\underline{\tilde{x}}^{(i)}\}_{i=1}^{N_v} \mid \hat{\underline{\theta}}^{(1)}, M^{(1)}\right) = 0.2$

$K = 2 \longrightarrow M^{(2)}, \hat{\underline{\theta}}^{(2)}$ $\quad P\left(\{\underline{\tilde{x}}^{(i)}\}_{i=1}^{N_v} \mid \hat{\underline{\theta}}^{(2)}, M^{(2)}\right) = 0.45$ ✓ → $M = 2$ optimal

$K = 3 \longrightarrow M^{(3)}, \hat{\underline{\theta}}^{(3)}$ $\quad P\left(\{\underline{\tilde{x}}^{(i)}\}_{i=1}^{N_v} \mid \hat{\underline{\theta}}^{(3)}, M^{(3)}\right) = 0.1$

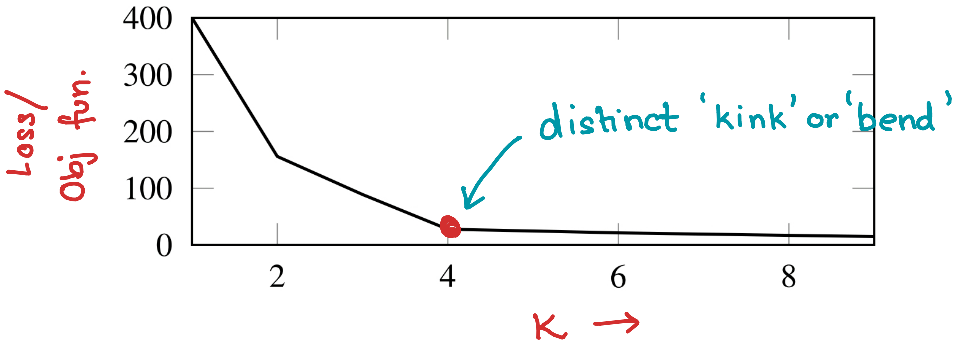
Choosing the number of clusters

- The validation methods should be handled with care

- In supervised learning, our goal is to obtain good predictions, so minimizing new data error makes sense

- In clustering, the goal is not necessarily to minimize "clustering loss" but to gain insights by finding a small number of clusters
  - So we may prefer a smaller number of clusters even if it gives not-so good validation loss

- The ELBOW method is often used for selecting K
  - plot of loss (either training, validation, or both)


Loss/
Obj fun.
K →
distinct 'kink' or 'bend'