CSE 575
Statistical Machine Learning

Lecture 17

YooJung Choi

Fall 2022

Clustering

- Given a large collection of objects $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$, can we group similar objects together?
- Central to cluster analysis are:
  - Notion of the degree of similarity / dissimilarity
  - Efficient clustering algorithms

Market segmentation

Image segmentation

Document analysis

GMM for clustering

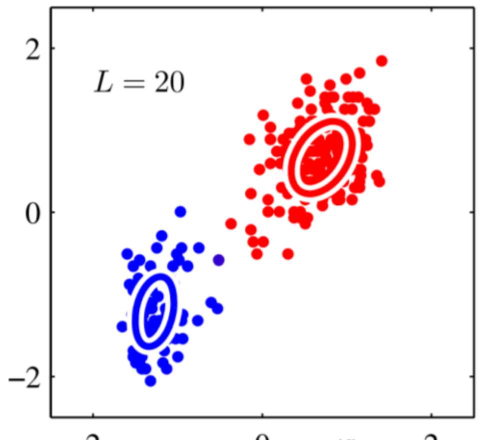
- Recall: latent variable interpretation of Gaussian mixture models

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_{k=1}^{K} p(z = k) \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- We can interpret the latent variable z as the *cluster*

- Soft clustering: GMM assigns a probability that a point $\mathbf{x}$ belongs to cluster $z = k$:

$$p(z = k \mid \mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$



(f)

- For hard clustering: assign $\mathbf{x}$ to the most likely cluster

$$\text{argmax}_k \, p(z = k \mid \mathbf{x})$$

K-means clustering

- Define "similarity" in terms of squared Euclidean (L2) distance

$$d(\mathbf{x}_n, \mathbf{x}_m) = \|\mathbf{x}_n - \mathbf{x}_m\|^2 = \sum_{i=1}^{D} (x_{ni} - x_{mi})^2$$

- Clustering: finding a mapping from each object $\mathbf{x}_n$ to cluster C_n
- Centroid-based clustering: represent each cluster by a centroid (a representative prototype) $\boldsymbol{\mu}_k$
- Objective: group objects to minimize the within-cluster sum of squared distances:

$$\operatorname{argmin}_{C,\boldsymbol{\mu}} \sum_{k=1}^{K} \sum_{n:C_n=k} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

Non-convex optimization

K-means algorithm

- Iteratively optimize the following, a la expectation maximization

$$\operatorname{argmin}_{C,\boldsymbol{\mu}} \sum_{k=1}^{K} \sum_{n:C_n=k} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

- Recall: EM for GMMs (informally)
  - E-step: guess the values of latent variable z_n for each $\mathbf{x}_n$
  - M-step: update the parameters π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ based on the guesses from the E-step
- K-means algorithm (informally): iteratively,
  - Guess the cluster C_n for each $\mathbf{x}_n$
  - Update $\boldsymbol{\mu}_k$ based on the assigned clusters

K-means algorithm

- Iteratively optimize the following, a la expectation maximization

$$\text{argmin}_{C,\boldsymbol{\mu}} \sum_{k=1}^{K} \sum_{n:C_n=k} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

1. Guess the cluster C_n for each $\mathbf{x}_n$
   - Fix $\boldsymbol{\mu}_k$ and minimize the following w.r.t C

Indicator function

$$\sum_{k=1}^{K} \sum_{n:C_n=k} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{I}[C_n = k]\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

   - Therefore, $C_n = \text{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$

Assign each point to the closest cluster

K-means algorithm

Exercise: $\sum_{n:C_n=k}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$ equivalent to $\frac{1}{2\mathrm{N}_k}\sum_{n,m:C_n=C_m=k}\|\mathbf{x}_n - \mathbf{x}_m\|^2$

K-means tries to minimize pairwise squared distances of points in the same cluster

- Iteratively optimize the following, a la expectation maximization

$$\mathrm{argmin}_{C,\boldsymbol{\mu}} \sum_{k=1}^{K}\sum_{n:C_n=k}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

2. Update $\boldsymbol{\mu}_k$ based on the assigned clusters
    - Fix C and minimize the following w.r.t $\boldsymbol{\mu}$

$$\sum_{k=1}^{K}\sum_{n:C_n=k}\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 = \sum_{k=1}^{K}\sum_{n:C_n=k}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

    - Take the partial derivative w.r.t. $\boldsymbol{\mu}_k$ and set it to zero

$$\frac{\partial}{\partial \boldsymbol{\mu}_k}\sum_{n:C_n=k}(\mathbf{x}_n - \boldsymbol{\mu}_k)^T(\mathbf{x}_n - \boldsymbol{\mu}_k) = \frac{\partial}{\partial \boldsymbol{\mu}_k}\sum_{n:C_n=k}(\mathbf{x}_n^T\mathbf{x}_n - 2\boldsymbol{\mu}_k^T\mathbf{x}_n + \boldsymbol{\mu}_k^T\boldsymbol{\mu}_k) = \sum_{n:C_n=k}(-2\mathbf{x}_n + 2\boldsymbol{\mu}_k) = 0$$

    - Therefore, $\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n:C_n=k}\mathbf{x}_n$ where $N_k = |\{n: C_n = k\}|$

Represent each cluster with the mean of all points in that cluster

K-means algorithm

Putting everything together

1. Initialize $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K$
2. Until convergence, repeat:
   1. For every n, set

$$C_n = \operatorname{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

   2. For every k, set

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n:C_n=k} \mathbf{x}_n \text{ where } N_k = |\{n: C_n = k\}|$$

K-means algorithm

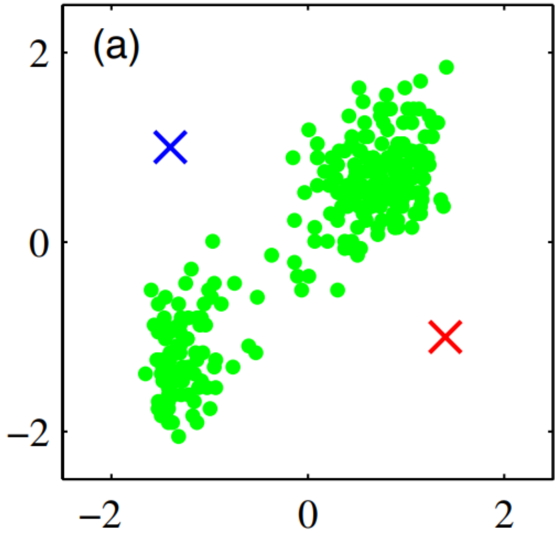
Putting everything together

1. Initialize $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K$
2. Until convergence, repeat:
    1. For every n, set

$$C_n = \mathrm{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

    2. For every k, set

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n:C_n=k} \mathbf{x}_n \text{ where } N_k = |\{n: C_n = k\}|$$



K-means algorithm

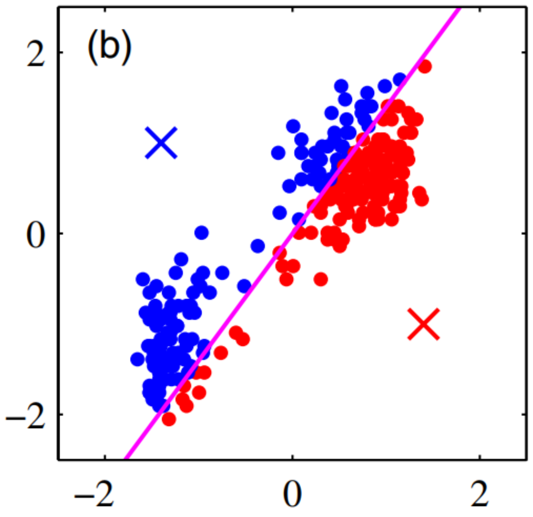
Putting everything together

1. Initialize $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K$
2. Until convergence, repeat:
    1. For every n, set

$$C_n = \operatorname{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

    2. For every k, set

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n:C_n=k} \mathbf{x}_n \text{ where } N_k = |\{n: C_n = k\}|$$



K-means algorithm

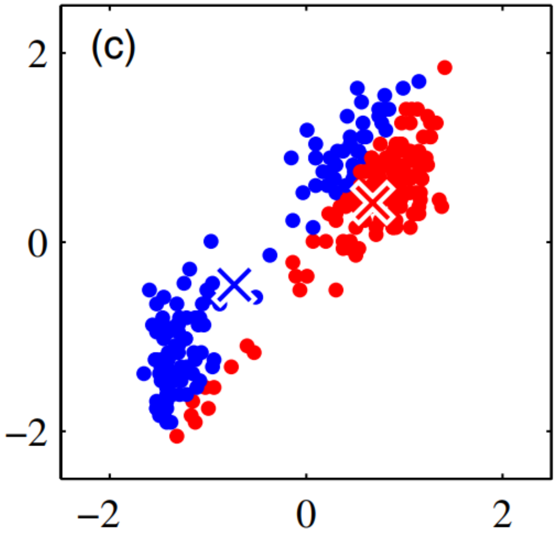
Putting everything together

1. Initialize $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
2. Until convergence, repeat:
    1. For every n, set

$$C_n = \operatorname{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

    2. For every k, set

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n: C_n = k} \mathbf{x}_n \text{ where } N_k = |\{n: C_n = k\}|$$



K-means algorithm

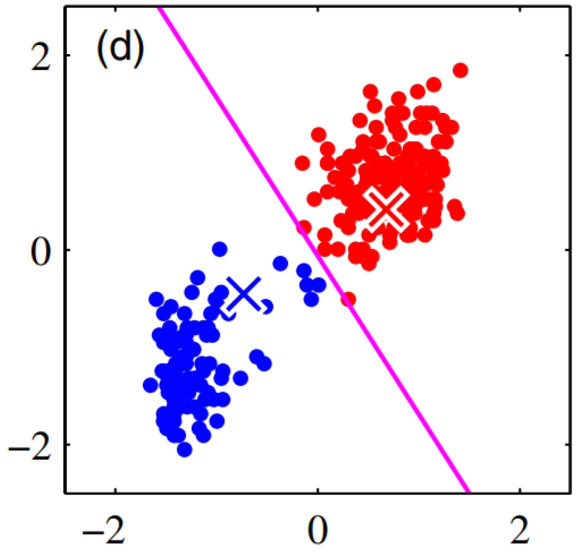
Putting everything together

1. Initialize $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
2. Until convergence, repeat:
   1. For every n, set

      $$C_n = \mathrm{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

   2. For every k, set

      $$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n: C_n = k} \mathbf{x}_n \text{ where } N_k = |\{n: C_n = k\}|$$



K-means algorithm

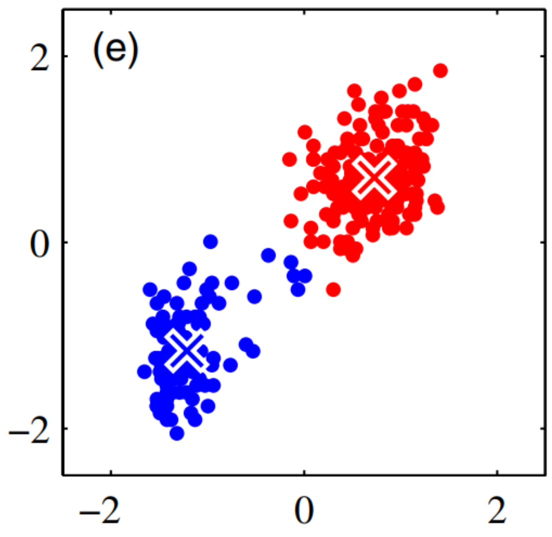
Putting everything together

1. Initialize $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
2. Until convergence, repeat:
    1. For every n, set

$$C_n = \operatorname{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

    2. For every k, set

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n:C_n=k} \mathbf{x}_n \text{ where } N_k = |\{n: C_n = k\}|$$



K-means algorithm

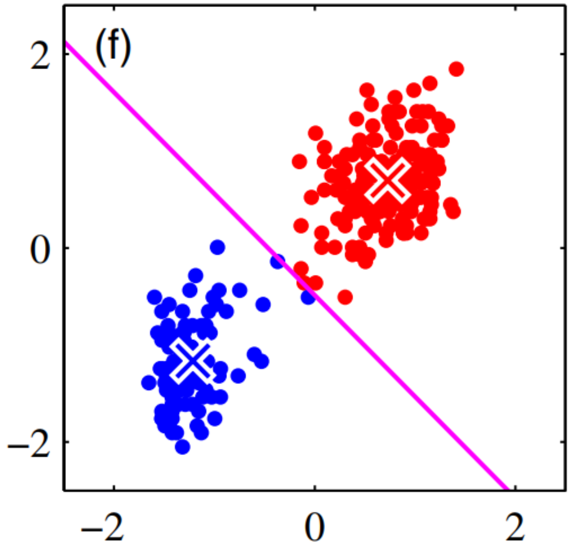
Putting everything together

1. Initialize $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K$
2. Until convergence, repeat:
    1. For every n, set

$$C_n = \mathrm{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

    2. For every k, set

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n:C_n=k} \mathbf{x}_n \text{ where } N_k = |\{n: C_n = k\}|$$



K-means algorithm

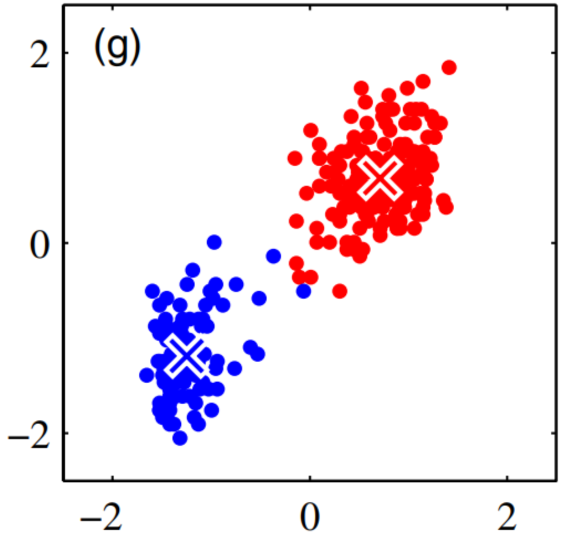
Putting everything together

1. Initialize $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
2. Until convergence, repeat:
    1. For every n, set

$$C_n = \operatorname{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

    2. For every k, set

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n: C_n = k} \mathbf{x}_n \text{ where } N_k = |\{n: C_n = k\}|$$



K-means algorithm

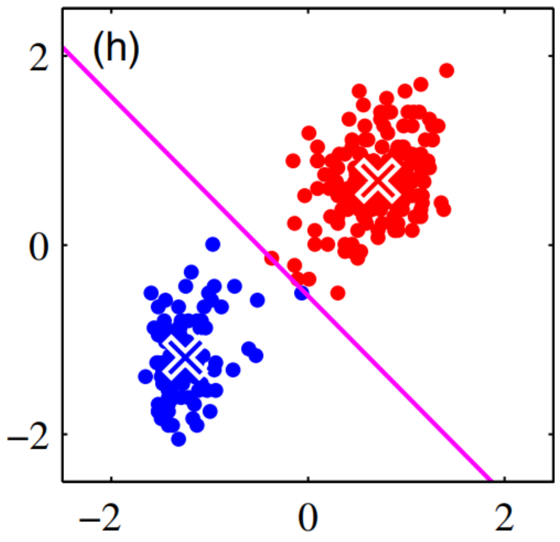
Putting everything together

1. Initialize $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$
2. Until convergence, repeat:
    1. For every n, set

$$C_n = \mathrm{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

    2. For every k, set

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n:C_n=k} \mathbf{x}_n \text{ where } N_k = |\{n: C_n = k\}|$$



K-means algorithm

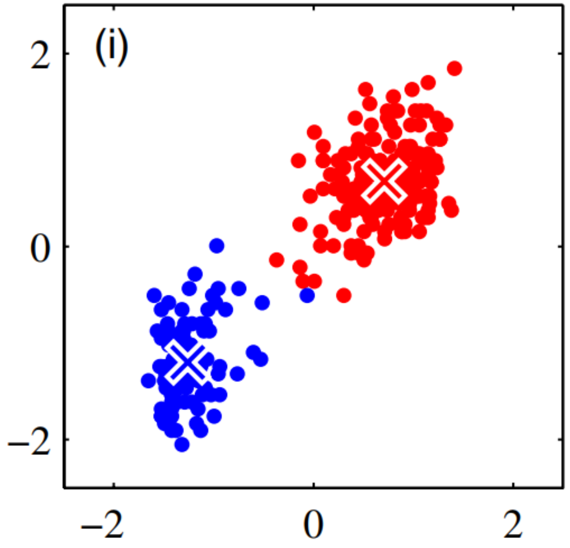
Putting everything together

1. Initialize $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K$
2. Until convergence, repeat:
    1. For every n, set

$$C_n = \operatorname{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

    2. For every k, set

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n: C_n = k} \mathbf{x}_n \text{ where } N_k = |\{n: C_n = k\}|$$

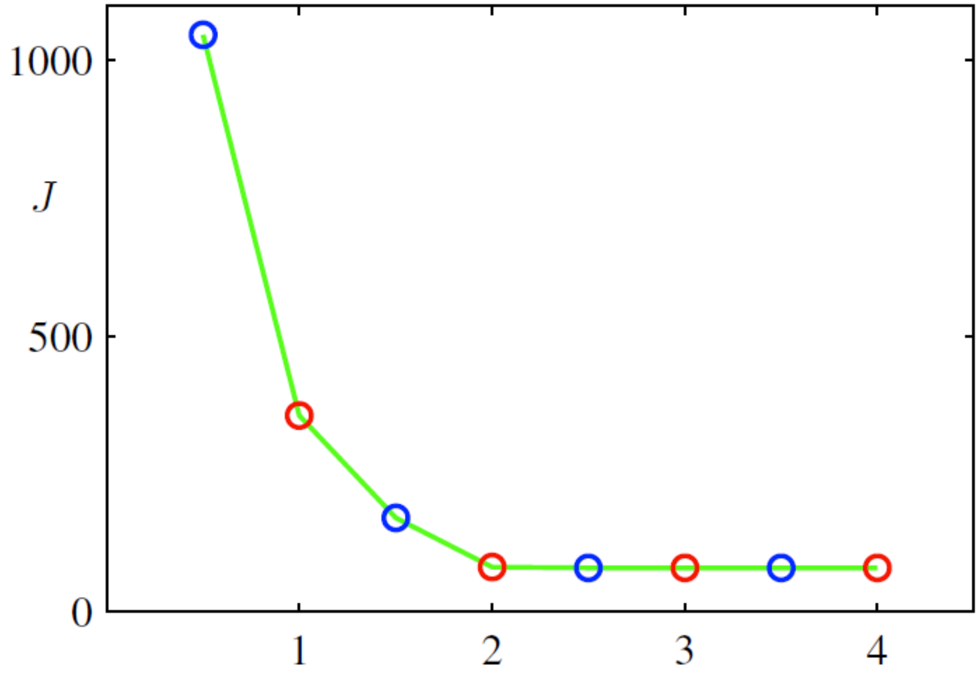


K-means: convergence

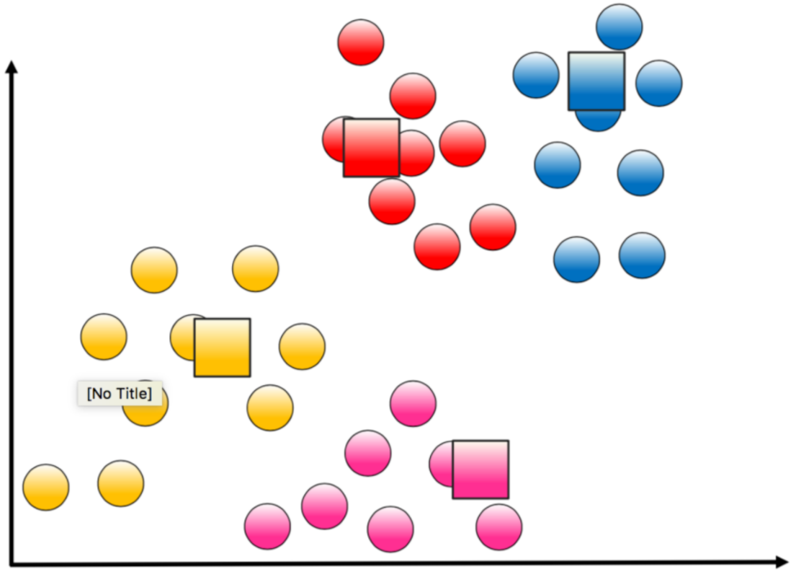
- Objective function value J is decreased in each E step & M step, in every iteration

$$J(C, \boldsymbol{\mu}) = \sum_{k=1}^{K} \sum_{n:C_n=k} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

- K-means always converges
- Algorithm is not guaranteed to converge to the global optimum
- Results depend on initialization



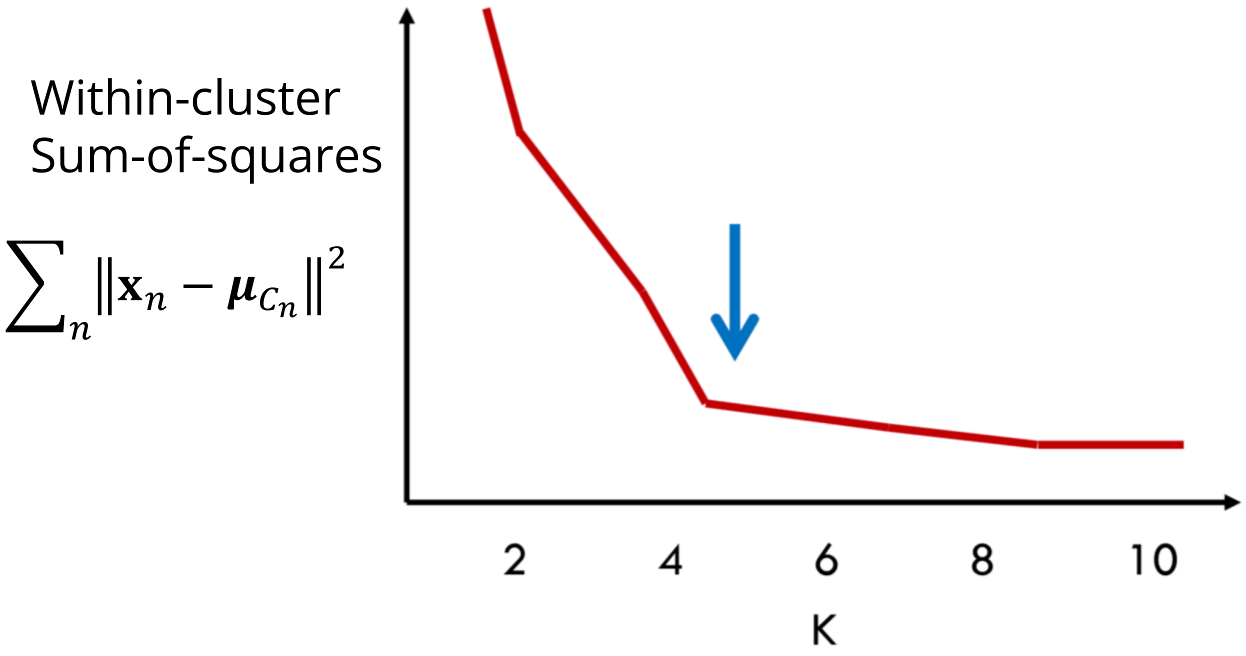
K-means++



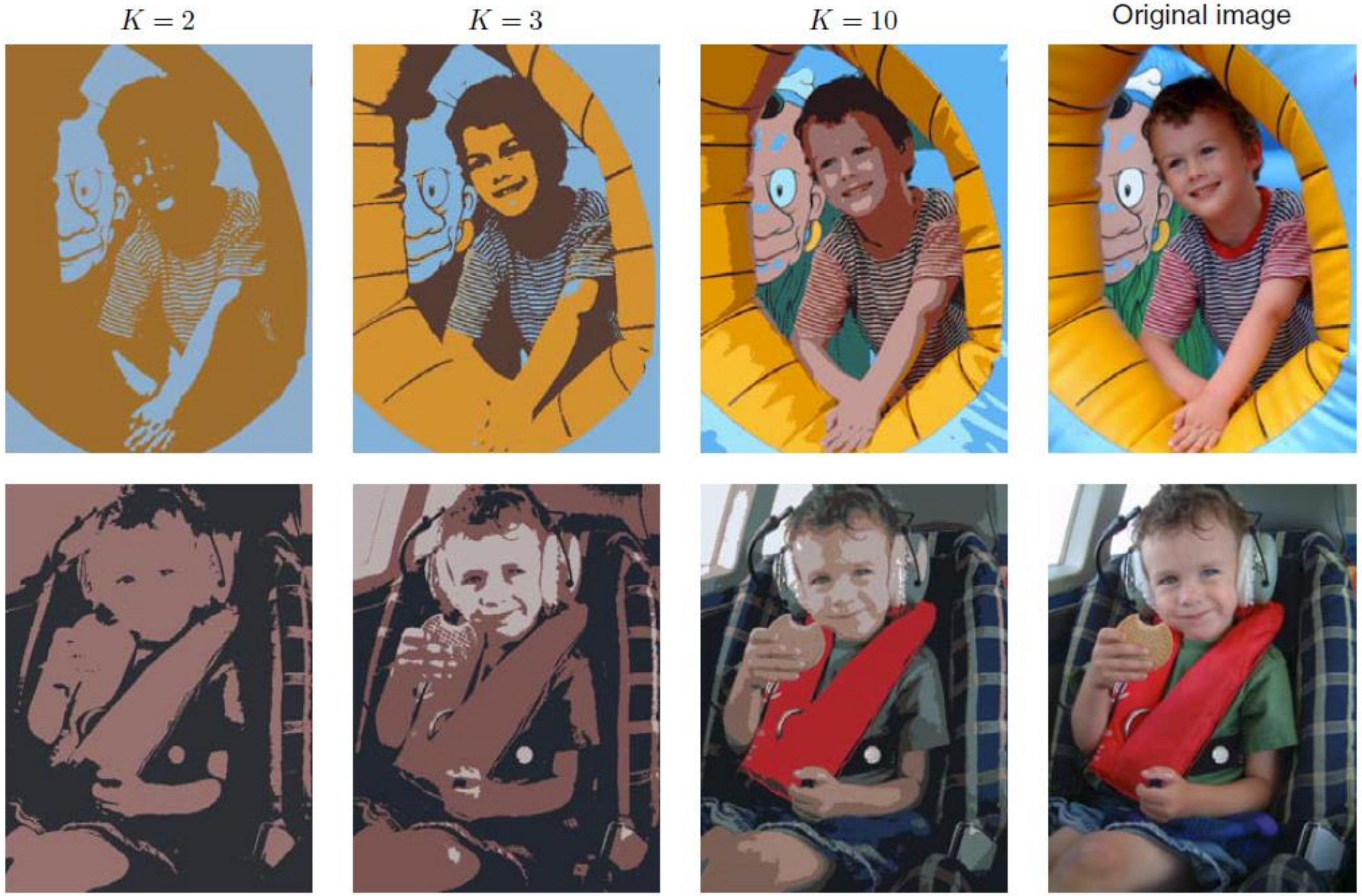
- Improved initialization for K-means
- Intuition: spread out the centroids
- Algorithm:
  1. Select an initial cluster center uniformly at random
  2. Compute $d(\mathbf{x}) = ||\mathbf{x} - \boldsymbol{\mu}_k||^2$ for each point, where k is the nearest center
  3. Sample the next centroid, with probability proportional to $d(\mathbf{x})$
  4. Repeat until K centroids have been chosen

How to choose K

- May be given as part of the problem
- May need to choose K: Elbow method (possibly with cross-validation)



Application: image segmentation

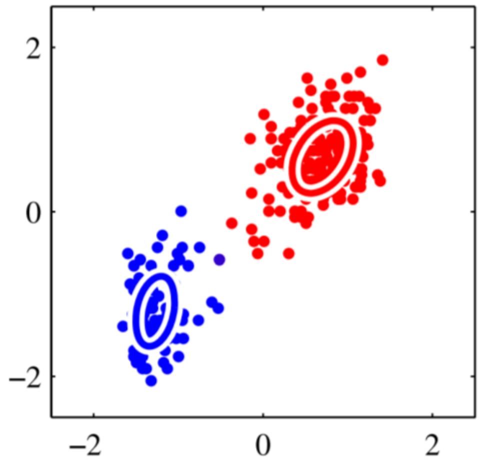


Relation to GMM

Gaussian mixture models

- Points that lie on this ellipse have the same contribution from the corresponding Gaussian component to their density

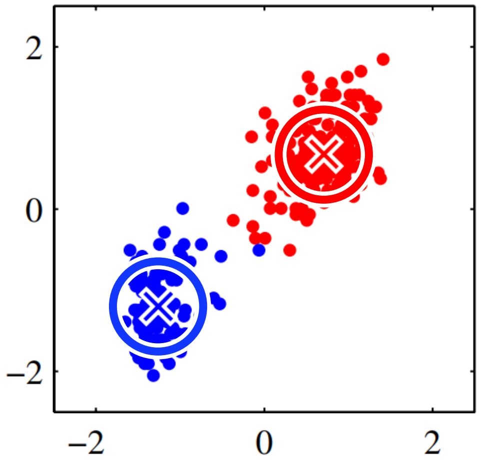
$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



K-means

- Points that lie on this circle have the same contribution from the corresponding centroid when assigning clusters

$$C_n = \mathrm{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$



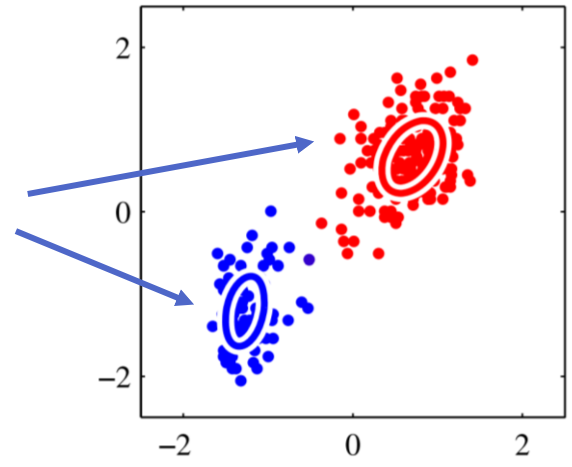
Analogous to a GMM with a spherical covariance matrix

i.e. $\boldsymbol{\Sigma}_k = \epsilon_k I$

Relation to GMM

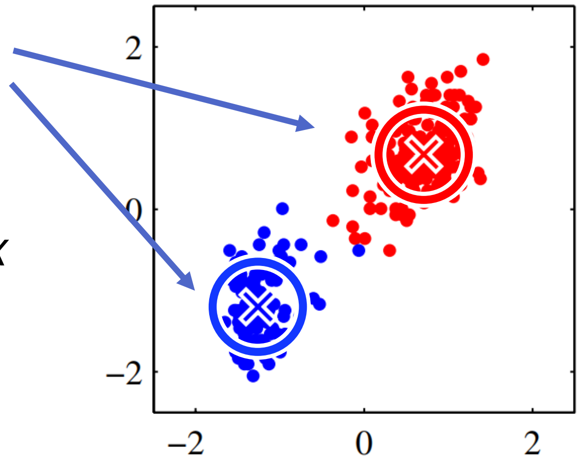
Gaussian mixture models

- The contours (ellipses) of equal contribution from the respective components can have different shapes and sizes



K-means

- The contours (circles/spheres) have the same shape and size across clusters



Analogous to a GMM with a shared covariance matrix

i.e. $\mathbf{\Sigma}_k = \mathbf{\Sigma} = \epsilon I$

Relation to GMM

- Hypothesis: K-means clustering is a special case of clustering given by a Gaussian mixture model with *a shared, spherical covariance matrix approaching zero* i.e. $\mathbf{\Sigma}_k = \epsilon I, \epsilon \to 0$

$$p(z = k \,|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}|\,\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)}{\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\,\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)} = \frac{\pi_k (2\pi)^{-\frac{D}{2}} |\mathbf{\Sigma}_k|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}}{\sum_{k=1}^K \pi_k (2\pi)^{-\frac{D}{2}} |\mathbf{\Sigma}_k|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}}$$

$$= \frac{\pi_k \cancel{(2\pi)^{-\frac{D}{2}} |\epsilon I|^{-\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \overset{\epsilon^{-1} I}{\cancel{(\epsilon I)^{-1}}}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}}{\sum_{k=1}^K \pi_k \cancel{(2\pi)^{-\frac{D}{2}} |\epsilon I|^{-\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \underset{\epsilon^{-1} I}{\cancel{(\epsilon I)^{-1}}}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}}$$

K-means cluster assignment!

$$= \frac{\pi_k \exp\left\{-\frac{1}{2\epsilon}\|\mathbf{x}-\boldsymbol{\mu}_k\|^2\right\}}{\sum_{k=1}^K \exp\left\{-\frac{1}{2\epsilon}\|\mathbf{x}-\boldsymbol{\mu}_k\|^2\right\}} \longrightarrow \begin{cases} 1 & \text{if } k = \text{argmin}_k \|\mathbf{x}-\boldsymbol{\mu}_k\|^2 \\ 0 & \text{otherwise} \end{cases} \text{ as } \epsilon \to 0$$

Relation to GMM

GMM

- Probabilistic
  - Finer grained, can express uncertainty
  - Can incorporate prior knowledge w/ Bayesian approach
- EM tends to take more iterations to converge
  - Initializing with K-means clusters works quite well
- More parameters: $O(K \cdot D^2)$
  - π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ for each k=1,…,K
- Elliptical/hyperbolic clusters

K-means

- Non-probabilistic
  - Directly solve for hard clustering
- Tends to converge faster
- Fewer parameters: $O(K \cdot D)$
  - $\boldsymbol{\mu}_k$ for each k=1,…,K
- Spherical clusters
  - Thus, a good idea to normalize data beforehand