ECE 295: Lecture 06 Unsupervised Learning

Spring 2018

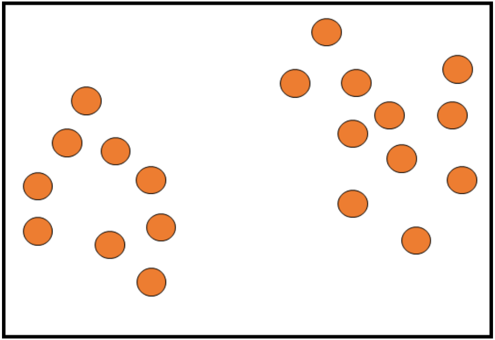
Prof Stanley Chan

School of Electrical and Computer Engineering
Purdue University

Unsupervised Learning

What is unsupervised learning?

- I give you a set of data points.
- They are not labeled.
- Your job is to learn structure from these data points.



Unsupervised Learning

We will learn two techniques.

Gaussian Mixture

- Requires a model.
- Uses the EM algorithm to estimate the parameters.
- Soft decision boundary.
- Sensitive to initial guesses.

K-Means

- Does not require a model.
- Uses clustering and mean shifting technique to estimate the parameters.
- Hard decision boundary.
- Also sensitive to initial guesses.

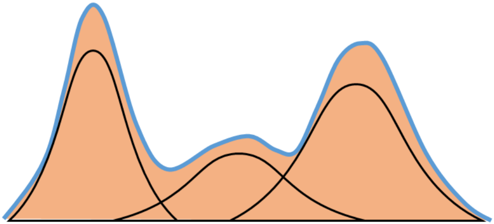
Gaussian mixture

Recall Gaussian:

$$\mathcal{N}(x \mid \mu_i, \sigma_i^2) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left\{ -\frac{(x-\mu_i^2)}{2\sigma_i^2} \right\}$$

A **Gaussian Mixture Model** (GMM) with K components is

$$p_X(x) = \sum_{i=1}^{K} \pi_i \mathcal{N}(x \mid \mu_i, \sigma_i^2)$$



Gaussian mixture

Ingredients of a GMM:

$$p_X(x) = \sum_{i=1}^{K} \pi_i \mathcal{N}(x \mid \mu_i, \sigma_i^2)$$

- μ_i = mean of the i-th Gaussian
- σ_i^2 = variance of the i-th Gaussian
- π_i = weight of the i-th Gaussian
- K = number of mixture components

Remark: We need $\sum_{i=1}^{K} \pi_i = 1$.

Question: Given data points $x_1, \ldots, x_j, \ldots, x_N$, how to estimate $(\pi_1, \mu_1, \sigma_1), \ldots, (\pi_K, \mu_K, \sigma_K)$?

Expectation Maximization

Expectation Maximization (EM) algorithm is a method to estimate the parameters.

- EM is iterative, so you need to do the steps multiple times.
- There are two steps. (1) Expectation, (2) Maximization.
- The iteration number is $(\cdot)^{(t)}$

Remark: Most computational packages has EM library.

Expectation Maximization for GMM

Step 1. Expectation. We define this quantity:

$$\gamma_{ij} = \frac{\pi_i^{(t)} \mathcal{N}(x_j \mid \mu_i^{(t)}, \sigma_i^{2(t)})}{\sum_{i=1}^{K} \pi_i^{(t)} \mathcal{N}(x_j \mid \mu_i^{(t)}, \sigma_i^{2(t)})}$$

Step 2. Maximization. Do the updates

$$\pi_i^{(t+1)} = \frac{1}{N} \sum_{j=1}^{N} \gamma_{ij}$$

$$\mu_i^{(t+1)} = \frac{\sum_{j=1}^{N} \gamma_{ij} x_j}{\sum_{j=1}^{N} \gamma_{ij}}$$

$$\sigma_i^{2(t+1)} = \frac{\sum_{j=1}^{N} \gamma_{ij} (x_j - \mu_i^{(t+1)})^2}{\sum_{j=1}^{N} \gamma_{ij}}$$

E-Step

Step 1. Expectation. We define this quantity:

$$\gamma_{ij} = \frac{\pi_i^{(t)} \mathcal{N}(x_j \mid \mu_i^{(t)}, \sigma_i^{2(t)})}{\sum_{i=1}^{K} \pi_i^{(t)} \mathcal{N}(x_j \mid \mu_i^{(t)}, \sigma_i^{2(t)})}$$

- Assume you are originally at $t-1$.
- You have parameters $\pi_i^{(t)}$, $\mu_i^{(t)}$ and $\sigma_i^{2(t)}$.
- Now you want to compute the parameters at t.
- In the E-step you compute γ_{ij}.
- γ_{ij} is a weighted average of the individual Gaussian at t.
- If x_j fits the current parameter, then γ_{ij} will be large.
- If x_j does not fit well, then γ_{ij} will adjust its weight.

M-Step

Step 2. Maximization. Do the updates

$$\pi_i^{(t+1)} = \frac{1}{N} \sum_{j=1}^{N} \gamma_{ij}$$

$$\mu_i^{(t+1)} = \frac{\sum_{j=1}^{N} \gamma_{ij} x_j}{\sum_{j=1}^{N} \gamma_{ij}}$$

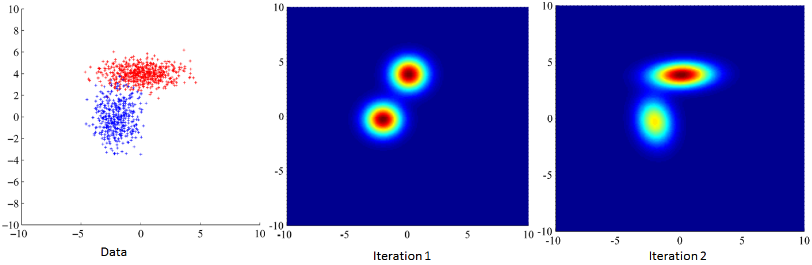
$$\sigma_i^{2(t+1)} = \frac{\sum_{j=1}^{N} \gamma_{ij} (x_j - \mu_i^{(t+1)})^2}{\sum_{j=1}^{N} \gamma_{ij}}$$

- They are all weighted averages of something
- The weights are the γ_{ij}
- You repeat Step 1 and Step 2 until "convergence"

Iterations of the EM

Here is a 2D example.

- Start with 1000 data points
- The initial GMM is arbitrary
- Looks reasonable in 2 iterations



M. R. Gupta, and Y. Chen, "Theory and Use of the EM Algorithm", Froundations and Trends in Signal Processing, vol. 4, no. 3, pp.223-296, 2010.

Summary of Mixture Model

- You need a model, typically a Gaussian
- Can use other types of models, e.g., mixtures of exponentials
- Need to determine K, which could be hard
- For high-dimensional Gaussian, we can change their shapes
- Mixture model is a type of "soft"-decision
- Most computing libraries have EM built-in for Gaussian mixture

K-means

An alternative method to do un-supervised learning.

Two steps.

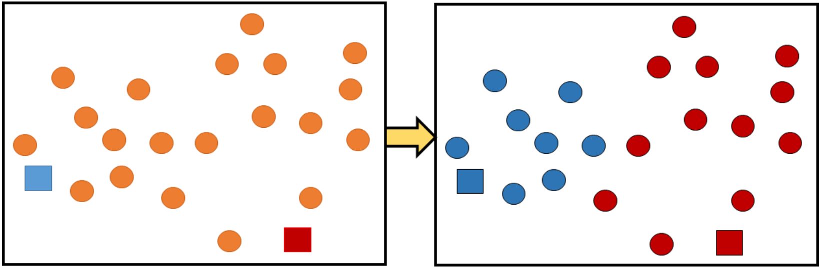
- Cluster Assignment
- Update Centroid

K-means

Example: $K = 2$.

Iteration 1a. Cluster Assignment.

- Start with two centroid $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$
- For every data point $\boldsymbol{x}_j$, find its *nearest* centroid
- Then label them according to the class



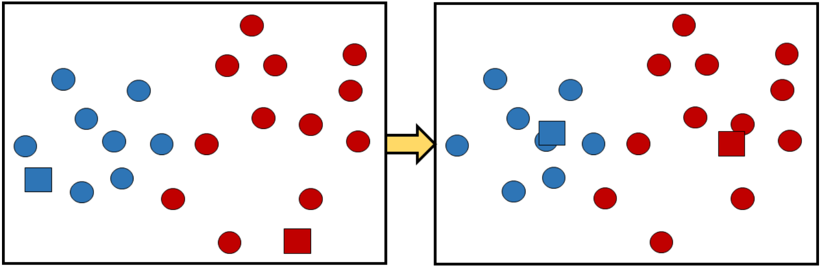
K-means

Iteration 1b. Centroid Update.

- Recompute the centroid

$$\boldsymbol{c}_i = \frac{1}{|\mathcal{C}_i|} \sum_{j \in \mathcal{C}_i} \boldsymbol{x}_j$$

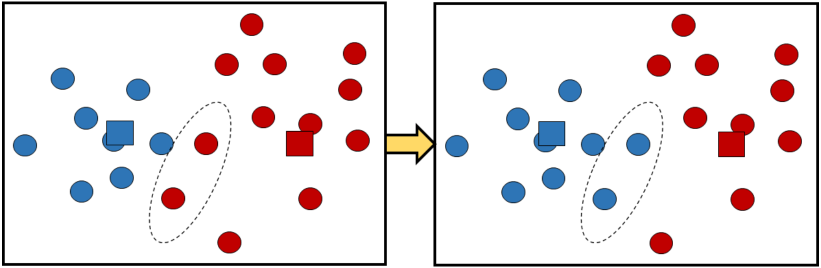
- Here $\mathcal{C}_i$ is the index set containing all data points in class i.



K-means

Iteration 2a. Cluster Assignment.

- Use the new centroid $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$
- For every data point $\boldsymbol{x}_j$, find its *nearest* centroid
- Then label them according to the class



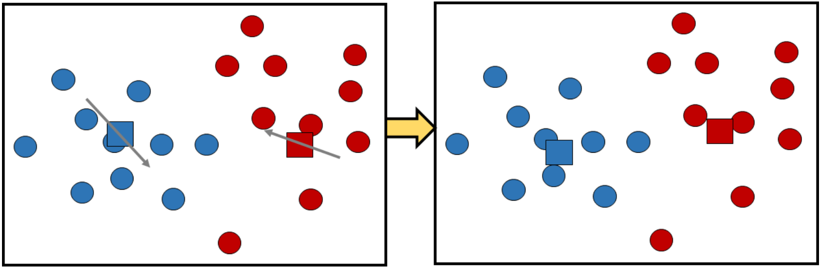
K-means

Iteration 2b. Centroid Update.

- Recompute the centroid

$$\boldsymbol{c}_i = \frac{1}{|\mathcal{C}_i|} \sum_{j \in \mathcal{C}_i} \boldsymbol{x}_j$$

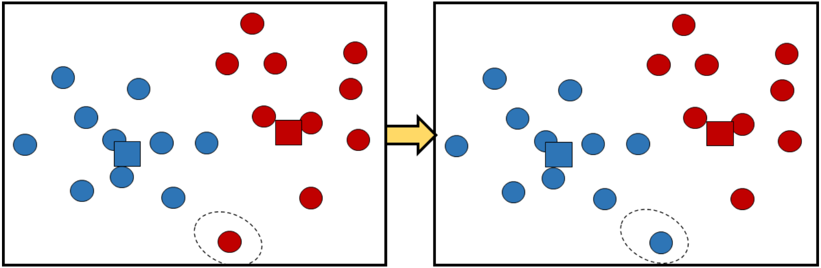
- Here $\mathcal{C}_k$ is the index set containing all data points in class k.



K-means

Iteration 3a. Cluster Assignment.

- Use the new centroid $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$
- For every data point $\boldsymbol{x}_j$, find its *nearest* centroid
- Then label them according to the class



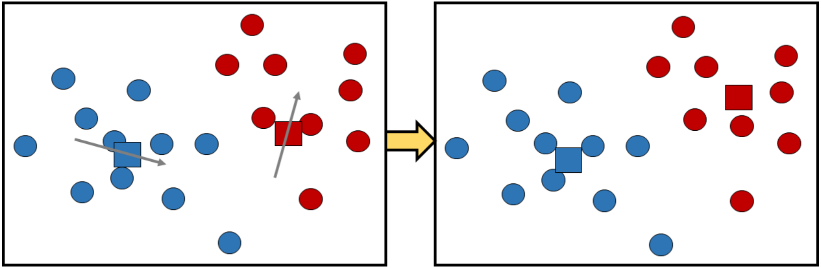
K-means

Iteration 3b. Centroid Update.

- Recompute the centroid

$$\boldsymbol{c}_i = \frac{1}{|\mathcal{C}_i|} \sum_{j \in \mathcal{C}_i} \boldsymbol{x}_j$$

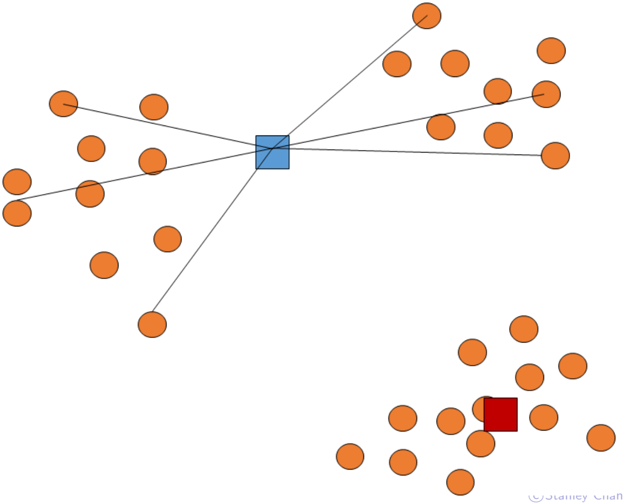
- Here $\mathcal{C}_k$ is the index set containing all data points in class k.



Stop if no more changes in clustering.

How to Select K?

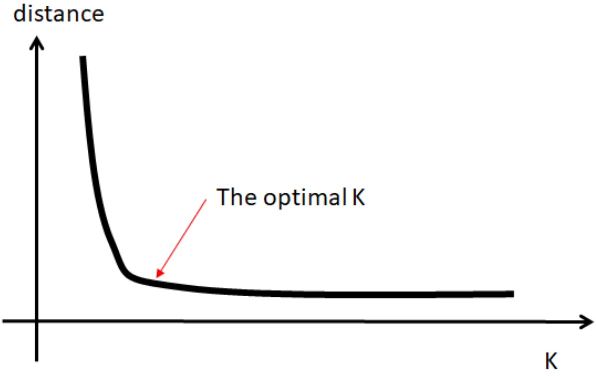
- If K is too small, then distance between $\boldsymbol{c}_i$ and and $\boldsymbol{x}_j$ will be large
- If K is too large, then distance will not drop much further



How to Select K?

Solution:

- Try a few K's.
- Find the one that starts to cause no more reduction



Summary of K-Means

- Less complicated than EM
- Often assume spherical geometry; May not work well for complex geometry
- "hard" decision

Summary

- Unsupervised learning
- GMM and K-means
- Model VS no model
- Both are high cost
- Both are sensitive to initialization
- Typical strategy is to randomize initialization