

# CS540 Introduction to Artificial Intelligence

## Lecture 15

Young Wu

Based on lecture slides by Jerry Zhu, Yingyu Liang, and Charles Dyer

July 27, 2022

# Supervised Learning

## Motivation

- Given training data and label.
- Discriminative: estimate  $\hat{\mathbb{P}}\{Y = y|X = x\}$  to classify.
- Generative: estimate  $\hat{\mathbb{P}}\{X = x|Y = y\}$  and Bayes rule to classify.

# Naive Bayes

## Motivation

- Naive Bayes:  $X_j \leftarrow Y$ .

$$\mathbb{P}\{Y = 1 | X_1 = x_1, \dots, X_m = x_m\}$$

$$\mathbb{P}\{Y = 1\} \prod_{j=1}^m \mathbb{P}\{X_j = x_j | Y = 1\}$$

$$= \frac{\mathbb{P}\{Y = 1\} \prod_{j=1}^m \mathbb{P}\{X_j = x_j | Y = 1\}}{\mathbb{P}\{X_1 = x_1, \dots, X_m = x_m\}}$$

$$= \frac{1}{1 + \exp\left(-\log\left(\frac{\mathbb{P}\{Y = 1\}}{\mathbb{P}\{Y = 0\}}\right) - \sum_{j=1}^m \log\left(\frac{\mathbb{P}\{X_j = x_j | Y = 1\}}{\mathbb{P}\{X_j = x_j | Y = 0\}}\right)\right)}$$

# Logistic Regression

## Motivation

$$\frac{1}{1 + \exp \left( -\log \left( \frac{\mathbb{P} \{Y = 1\}}{\mathbb{P} \{Y = 0\}} \right) - \sum_{j=1}^m \log \left( \frac{\mathbb{P} \{X_j = x_j | Y = 1\}}{\mathbb{P} \{X_j = x_j | Y = 0\}} \right) \right)}$$

- Logistic Regression:  $X_j \rightarrow Y$ .

$$\begin{aligned} & \tilde{\mathbb{P}} \{Y = 1 | X_1 = x_1, \dots, X_m = x_m\} \\ &= \frac{1}{1 + \exp \left( - \left( b + \sum_{j=1}^m w_j x_j \right) \right)} \end{aligned}$$

# Generative Adversarial Network

## Motivation

- Generative Adversarial Network (GAN): two competitive neural networks.
- ① Generative network input random noise and output fake images.
- ② Discriminative network input real and fake images and output label real or fake.

# Generative Adversarial Network Diagram

## Motivation

# Unsupervised Learning

## Motivation

- Supervised learning:  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  .
  - Unsupervised learning:  $x_1, x_2, \dots, x_n$  .
  - There are a few common tasks without labels.
- 1 Clustering: separate instances into groups.
  - 2 Novelty (outlier) detection: find instances that are different.
  - 3 Dimensionality reduction: represent each instance with a lower dimensional feature vector while maintaining key characteristics.

# Unsupervised Learning Applications

## Motivation

- 1 Google News
- 2 Google Photo
- 3 Image Segmentation
- 4 Text Processing



# Hierarchical Clustering

## Description

- Start with each instance as a cluster.
- Merge clusters that are closest to each other.
- Result in a binary tree with close clusters as children.

# Hierarchical Clustering Diagram

## Description

# Clusters

## Definition

- A cluster is a set of instances.

$$C_k \subseteq \{x_i\}_{i=1}^n$$

- A clustering is a partition of the set of instances into clusters.

$$C = C_1, C_2, \dots, C_K$$

$$C_k \cap C_{k'} = \emptyset \text{ for } k' \neq k, \bigcup_{k=1}^K C_k = \{x_i\}_{i=1}^n$$

# Distance between Points

## Definition

- Usually, the distance between two instances is measured by the Euclidean distance or  $L_2$  distance.

$$d(x_i, x_{i'}) = \|x_i - x_{i'}\|_2 = \sqrt{\sum_{j=1}^m (x_{ij} - x_{i'j})^2}$$

- Other examples include:  $L_1$  distance and  $L_\infty$  distance.

$$d_1(x_i, x_{i'}) = \|x_i - x_{i'}\|_1 = \sum_{j=1}^m |x_{ij} - x_{i'j}|$$

$$d_\infty(x_i, x_{i'}) = \|x_i - x_{i'}\|_\infty = \max_{j=1,2,\dots,m} \{|x_{ij} - x_{i'j}|\}$$

# Single Linkage Distance

## Definition

- Usually, the distance between two clusters is measured by the single-linkage distance.

$$d(C_k, C_{k'}) = \min \{d(x_i, x_{i'}) : x_i \in C_k, x_{i'} \in C_{k'}\}$$

- It is the shortest distance from any instance in one cluster to any instance in the other cluster.

# Complete Linkage Distance

## Definition

- Another measure is complete-linkage distance,

$$d(C_k, C_{k'}) = \max \{d(x_i, x_{i'}) : x_i \in C_k, x_{i'} \in C_{k'}\}$$

- It is the longest distance from any instance in one cluster to any instance in the other cluster.

# Average Linkage Distance Diagram

## Definition

- Another measure is average-linkage distance.

$$d(C_k, C_{k'}) = \frac{1}{|C_k| |C_{k'}|} \sum_{x_i \in C_k, x_{i'} \in C_{k'}} d(x_i, x_{i'})$$

- It is the average distance from any instance in one cluster to any instance in the other cluster.

# Hierarchical Clustering

## Algorithm

- Input: instances:  $\{x_i\}_{i=1}^n$ , the number of clusters  $K$ , and a distance function  $d$ .
- Output: a list of clusters  $C = C_1, C_2, \dots, C_K$ .
- Initialize for  $t = 0$ .

$$C^{(0)} = C_1^{(0)}, \dots, C_n^{(0)}, \text{ where } C_k^{(0)} = \{x_k\}, k = 1, 2, \dots, n$$

- Loop for  $t = 1, 2, \dots, n - k + 1$ .

$$(k_1^*, k_2^*) = \underset{k_1, k_2}{\operatorname{argmin}} d \left( C_{k_1}^{(t-1)}, C_{k_2}^{(t-1)} \right)$$

$$C^{(t)} = \left( C_{k_1^*}^{(t-1)} \cup C_{k_2^*}^{(t-1)} \right), C_1^{(t-1)}, \dots, \text{no } k_1^*, k_2^*, \dots, C_n^{(t-1)}$$



# Number of Clusters

## Discussion

- $K$  can be chosen using prior knowledge about  $X$ .
- The algorithm can stop merging as soon as all the between-cluster distances are larger than some fixed  $R$ .
- The binary tree generated in the process is often called dendrogram, or taxonomy, or a hierarchy of data points.
- An example of a dendrogram is the tree of life in biology.

# *K* Means Clustering

## Description

- This is not *K* Nearest Neighbor.
- Start with random cluster centers.
- Assign each point to its closest center.
- Update all cluster centers as the center of its points.

# Center

## Definition

- The center is the average of the instances in the cluster,

$$c_k = \frac{1}{|C_k|} \sum_{x \in C_k} x$$

# Distortion

## Distortion

- Distortion for a point is the distance from the point to its cluster center.
- Total distortion is the sum of distortion for all points.

$$D_K = \sum_{i=1}^n d(x_i, c_{k^*(x_i)}(x_i))$$

$$k^*(x) = \operatorname{argmin}_{k=1,2,\dots,K} d(x, c_k)$$

# Objective Function

## Definition

- When using Euclidean distance, sometimes total distortion is defined as sum of squared distances.

$$D_K = \sum_{i=1}^n d_2(x_i, c_{k^*(x_i)}(x_i))^2$$

- This algorithm stop in finite steps.
- This algorithm is trying to minimize the total distortion but fails.

# Gradient Descent

## Definition

- When  $d$  is the Euclidean distance.  $K$  Means algorithm is the gradient descent when distortion is the objective (cost) function.

$$\frac{\partial}{\partial c_k} \sum_{k=1}^K \sum_{x \in C_k} \|x - c_k\|_2^2 = 0$$

$$\Rightarrow -2 \sum_{x \in C_k} (x - c_k) = 0$$

$$\Rightarrow c_k = \frac{1}{|C_k|} \sum_{x \in C_k} x$$

# K Means Clustering

## Algorithm

- Input: instances:  $\{x_i\}_{i=1}^n$ , the number of clusters  $K$ , and a distance function  $d$ .
- Output: a list of clusters  $C = C_1, C_2, \dots, C_K$ .
- Initialize  $t = 0$ .

$$c_k^{(0)} = K \text{ random points}$$

- Loop until  $c^{(t)} = c^{(t-1)}$ .

$$C_k^{(t-1)} = \left\{ x : k = \underset{k' \in \{1, 2, \dots, K\}}{\operatorname{argmin}} d(x, c_{k'}^{(t-1)}) \right\}$$

$$c_k^{(t)} = \frac{1}{|C_k^{(t-1)}|} \sum_{x \in C_k^{(t-1)}} x$$

# Number of Clusters

## Discussion

- There are a few ways to pick the number of clusters  $K$ .
- ①  $K$  can be chosen using prior knowledge about  $X$ .
- ②  $K$  can be the one that minimizes distortion? No, when  $K = n$ , distortion = 0.
- ③  $K$  can be the one that minimizes distortion + regularizer.

$$K^* = \operatorname{argmin}_k (D_k + \lambda \cdot m \cdot k \cdot \log n)$$

- $\lambda$  is a fixed constant chosen arbitrarily.



# Initial Clusters

## Discussion

- There are a few ways to initialize the clusters.
- ①  $K$  uniform random points in  $\{x_i\}_{i=1}^n$ .
- ② 1 uniform random point in  $\{x_i\}_{i=1}^n$  as  $c_1^{(0)}$ , then find the farthest point in  $\{x_i\}_{i=1}^n$  from  $c_1^{(0)}$  as  $c_2^{(0)}$ , and find the farthest point in  $\{x_i\}_{i=1}^n$  from the closer of  $c_1^{(0)}$  and  $c_2^{(0)}$  as  $c_3^{(0)}$ , and repeat this  $K$  times.

# Gaussian Mixture Model

## Discussion

- In *K* means, each instance belong to one cluster with certainty.
- One continuous version is called the Gaussian mixture model: each instance belongs to one of the clusters with a positive probability.
- The model can be trained using Expectation Maximization Algorithm (EM Algorithm).

# EM Algorithm, Part I

## Discussion

- The means  $\mu_k$  and variances  $\sigma_k^2$  for each cluster need to be trained. The mixing probability  $\pi_k$  also needs to be trained.

$$(\mu_1, \sigma_1^2, \pi_1), (\mu_2, \sigma_2^2, \pi_2), \dots, (\mu_K, \sigma_K^2, \pi_K)$$

- Initialize by random guesses of clusters means and variances.

# EM Algorithm, Part II

## Discussion

- Expectation Step. Compute responsibilities for  $i = 1, 2, \dots, n$  and  $k = 1, 2, \dots, K$ .

$$\hat{\gamma}_{i,k} = \frac{\hat{\pi}_k \varphi_k(x_i)}{\sum_{k'=1,2,\dots,K} \hat{\pi}_{k'} \varphi_{k'}(x_i)}$$

$$\varphi_k(x) = \frac{1}{\sqrt{2\pi\hat{\sigma}_k}} \exp\left(-\frac{(x - \hat{\mu}_k)^2}{2\hat{\sigma}_k^2}\right)$$

# EM Algorithm, Part III

## Discussion

- Maximization Step. Compute means and variances for each  $k = 1, 2, \dots, K$ .

$$\hat{\mu}_k = \frac{\sum_{i=1}^n \hat{\gamma}_{i,k} x_i}{\sum_{i=1}^n \hat{\gamma}_i}, \text{ and } \hat{\sigma}_k^2 = \frac{\sum_{i=1}^n \hat{\gamma}_{i,k} (x_i - \hat{\mu}_k)^2}{\sum_{i=1}^n \hat{\gamma}_i}$$

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^n \hat{\gamma}_{i,k}$$

- Repeat until convergent.