QTM 347 Machine Learning

Lecture 20: Clustering Methods and Neural Networks

Ruoxuan Xiong

Suggested reading: ISL Chapter 10 and 12

Final project presentation

- 5 groups on 4/23 and 6 groups on 4/28
- Sign-up sheet released on 4/14
- 12 minutes presentation and 2~3 minutes Q&A (count toward class participation)
- Send me the slides before 11:59 PM on the day before the presentation

Structure of presentation

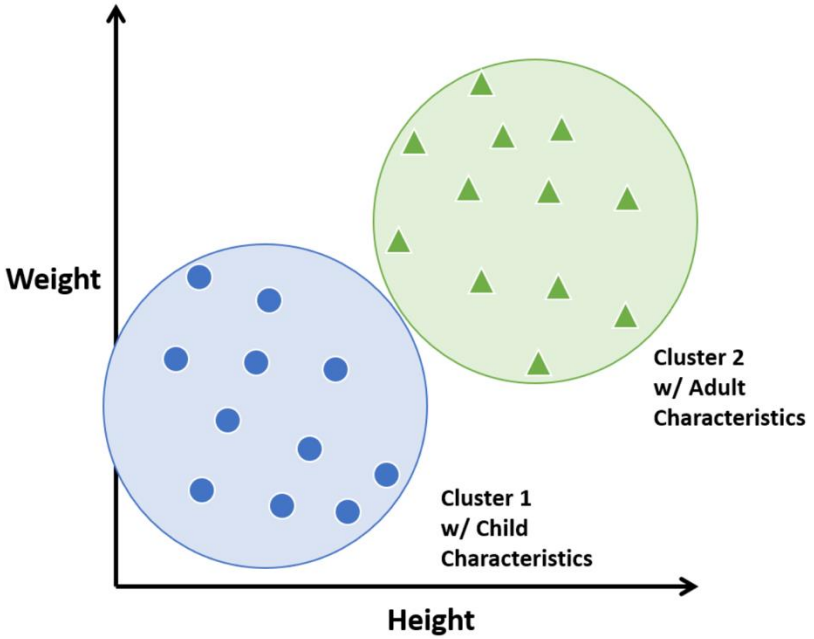
- Introduction
  - What is the problem
  - Why is the problem interesting and important
- Setup
  - Describe the dataset
  - Experiment setup
- Results
  - Main results and findings
  - Supplementary results including model and parameter choices
- Discussion and conclusion

Lecture plan

- K-means clustering
- Neural networks

Clustering methods

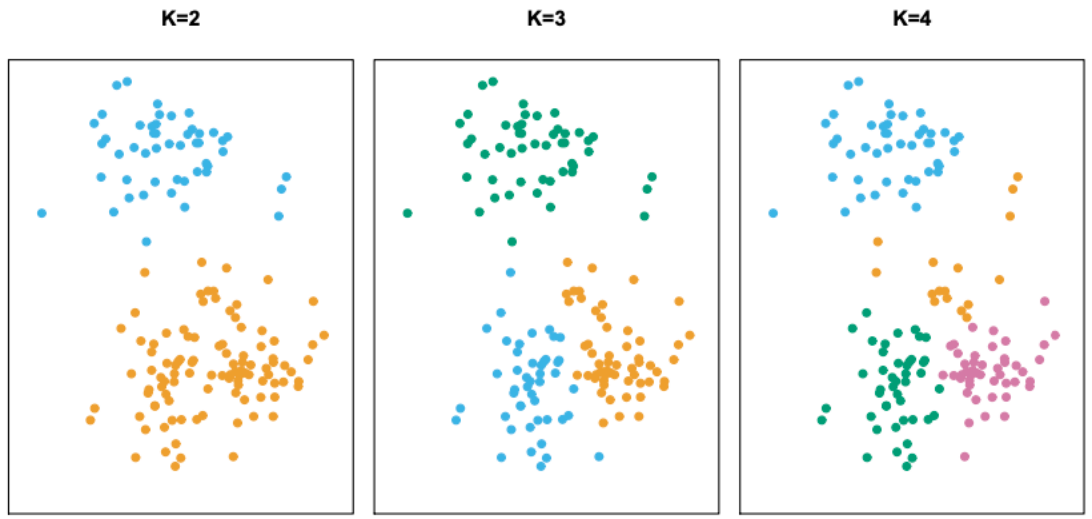
- PCA looks to find a low-dimensional representation of observations
- Clustering looks to find homogeneous subgroups among observations
- A marketing example
  - Perform market segmentation by identifying subgroups who are more likely to purchase a particular product
  - Use features, e.g., median household income, occupation, distance form nearest urban area



EMORY

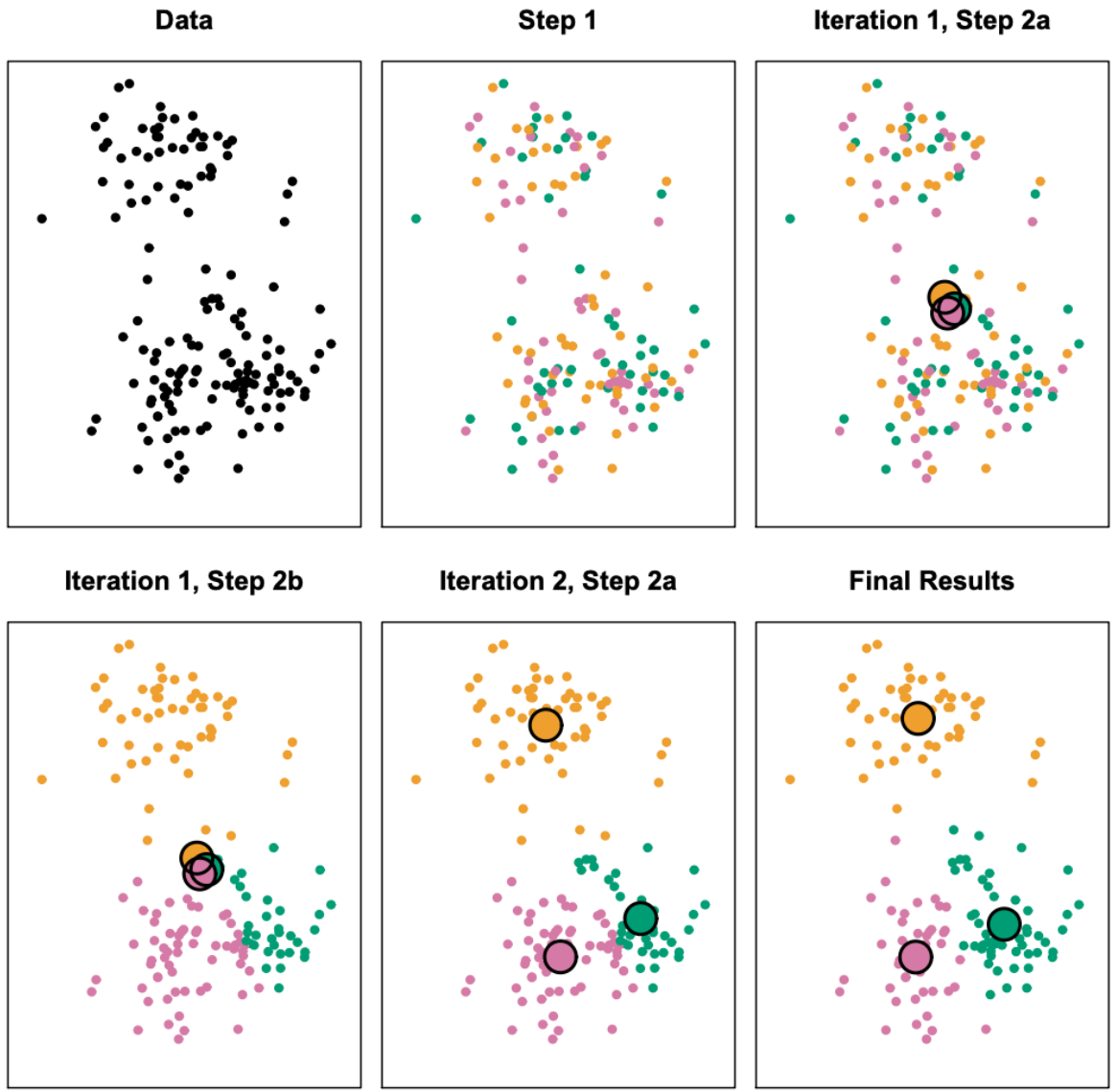
K-means clustering

- A *good clustering* is one for which the within-cluster variation is as small as possible
- **K-means clustering** is a simple and elegant approach for partitioning a data set into *K distinct, non-overlapping clusters*
  - Let $C_1, \cdots, C_K$ denote set of indices of observations in each cluster
    - $C_1 \cup \cdots \cup C_K = \{1, \cdots, n\}$: each observation belongs to at least one cluster
    - $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$: no observation belongs to more than one cluster



K-means clustering procedure

- *Randomly assign* a number, from 1 to K, to each observation, i.e., initial cluster assignment
- *Iterate* until the cluster assignments stop changing
  - For each cluster, *compute the cluster centroid*, i.e., vector of the p feature means for observations in the kth cluster
  - *Assign each observation* to the *cluster* whose *centroid is closest*



EMORY

Initialization of K-means clustering

- The *results* of K-means clustering will *depend on the initial cluster assignment*
- Solution: *run* the algorithm *multiple times* from different random initial configurations. Then one *selects the best solution*

K-means clustering performed six times with $K = 3$ and each time with a different initial cluster. Those labeled in red are optimal, with an objective value of 235.8.

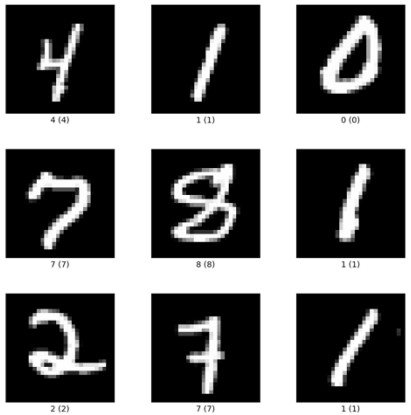
Lecture plan

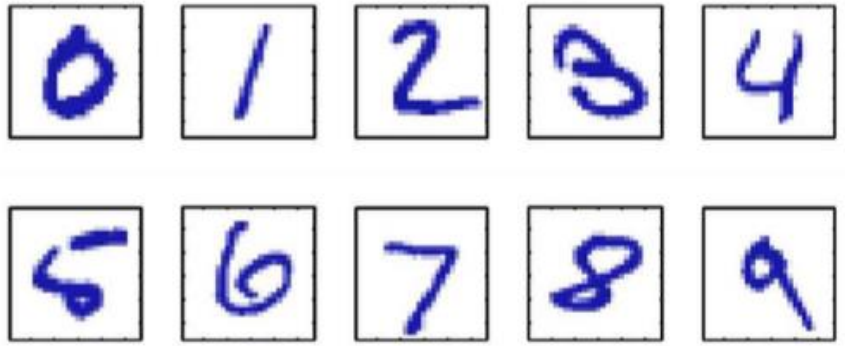
- K-means clustering

- Neural networks

Recognizing handwritten digits

- Input is a collection of handwritten digits from 0 to 9 in black and white



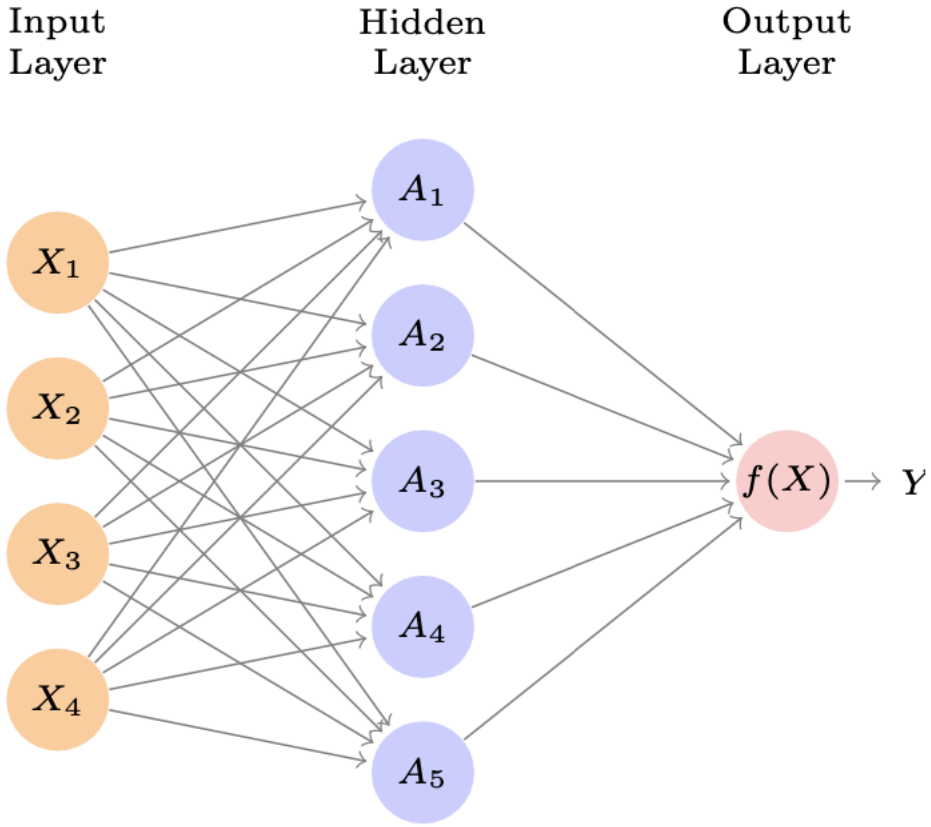


- **MNIST data set**
  - 50,000 handwritten digits for training
  - 5,000 for validation
  - 5,000 for testing

Link to dataset http://yann.lecun.com/exdb/mnist/

Single-layer neural network

- A neural network takes an input vector $X = (X_1, X_2, \cdots, X_p)$ and builds a nonlinear function $f(X)$ to predict Y



- This simple approach works well on MNIST (over 90% test accuracy)

Single-layer neural network

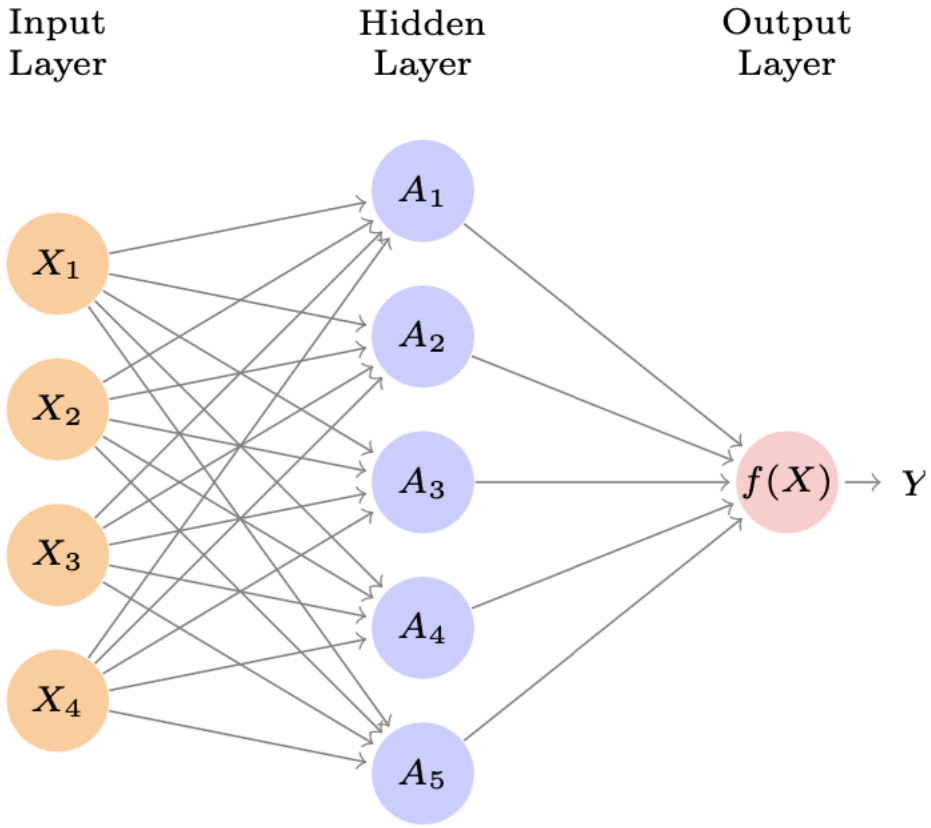
- The neural network model has the form

$$f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k h_k(X)$$

Here $h_k(X)$ is a neuron / hidden unit (analogous to neurons in the brain)

$$A_k = h_k(X) = g\left(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j\right)$$

g is a nonlinear activation function



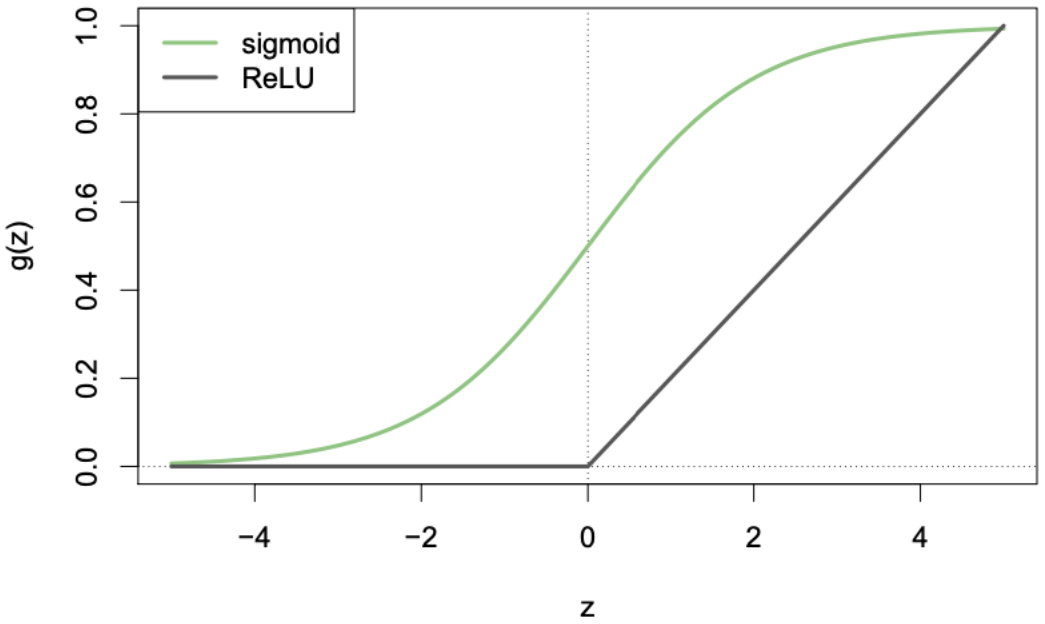
Activation function

- In the early instances of neural networks, the *sigmoid activation function* is preferred

$$g(z) = \frac{1}{1+e^{-z}}$$

- In modern neural networks, *ReLU (rectified linear unit)* is the preferred choice

$$g(z) = \begin{cases} 0 & z < 0 \\ z & z \geq 0 \end{cases}$$



A toy example

- Two features $X = (X_1, X_2)$ and two hidden units $h_1(X)$ and $h_2(X)$ with $g(z) = z^2$

- We specify parameters as (9 parameters in total)

$$\beta_0 = 0 \quad \beta_1 = \frac{1}{4} \quad \beta_2 = -\frac{1}{4}$$
$$w_{10} = 0 \quad w_{11} = 1 \quad w_{12} = 1$$
$$w_{20} = 0 \quad w_{21} = 1 \quad w_{22} = -1$$

$$h_1(X) = (w_{10} + w_{11}X_1 + w_{12}X_2)^2 = (0 + X_1 + X_2)^2$$

$$h_2(X) = (w_{20} + w_{21}X_1 + w_{22}X_2)^2 = (0 + X_1 - X_2)^2$$

$$f(X) = \beta_0 + \beta_1 h_1(X) + \beta_2 h_2(X) = \frac{1}{4}(0 + X_1 + X_2)^2 - \frac{1}{4}(0 + X_1 - X_2)^2 = X_1X_2$$

Fitting a neural network

- Fitting a neural network requires estimating the unknown parameters (β_k and w_{kj}) typically *using stochastic gradient descent and backpropagation*.
- For a regression problem, typically *squared-error* is used

$$\sum_{i=1}^{n}\left(y_i - f(x_i)\right)^2$$

- For a classification problem, typically *cross-entropy (negative multinomial log-likelihood)* is used

$$-\sum_{i=1}^{n}\sum_{m} y_{im}\log(f_m(x_i))$$

$Y = (Y_0, Y_1, \cdots, Y_M)$ with *a one* in the position corresponding to label and zeros elsewhere ($M+1$ classes)

$f_m(X) = \Pr(Y = m \mid X)$ the *probability* in class m

Multilayer neural networks

- Modern neural networks typically have more than one hidden layer and often many units per layer

Method	Test Error
Neural Network + Ridge Regularization	2.3%
Neural Network + Dropout Regularization	1.8%
Multinomial Logistic Regression	7.2%
Linear Discriminant Analysis	12.7%

Test error rate on the MNIST data