ASICLAND
(주)에이직랜드

ASIC & AI

Stage 1: AI introduction & algorithms

Members: Jce, Ryan, Taki Lee, Sam Kim, Brian Choi

Created by: Tony Do

30.Sept.2019

Content

- AI Introduction
- Requirements
- Applications
- Linear regression
- K-nearest neighbors
- K-mean clustering
- Naïve Bayes classifier
- Overfitting

AI introduction

- Artificial Intelligence is an approach to make a computer, a robot, or a product to think how smart human think.
- Reasoning
- Learning
- Problem Solving
- Perception

AI application

- Education
- Transportation: autonomous vehicle, drone, robotics…
- Healthcare system
- AR, VR ….

Requirement

- Mathematics
- Matrix calculation
- Probabilities
- programming

AI projects

Deep Learning

- https://www.youtube.com/watch?v=B-0CkG2oqX0
- https://www.youtube.com/watch?v=CUfrgXUDSEE
- https://www.youtube.com/watch?v=GGuO04g8aLE&t=55s

Reinforcement Learning

- https://www.youtube.com/watch?v=d5zR5iQ3BrQ
- https://www.youtube.com/watch?v=01tC8wz9-TU
- https://www.youtube.com/watch?v=PfGbDEJPrXk

Formula of Machine learning problems

$$\hat{y} = f(x, W) \qquad (1)$$

How to find out f satisfy:

The predicted value $\hat{y}$ close to the true value y.

Steps of machine learning with dataset:

1. Dataset will be divided to 2 parts: training data, test data.
2. Use training data to find out f – function (feature function)
3. Test f – function by comparison the output $\hat{y}$ with y
4. The good feature function will be evaluated by how close of $\hat{y}$ and y

Linear regression

- **Linear regression** is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables).
- ***Given dataset***

$\{y_i, x_{i1}, \ldots., x_{ip}\}_{i=1}^{n}$, n statistical units

$y_i = w_0 + w_1 x_{i1} + w_2 x_{i2} + \ldots + w_p x_{ip} = x_i.\mathrm{W}$

⇒Y = x.W

$$\Rightarrow \mathrm{x} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix} \quad ; \qquad W = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{bmatrix}$$

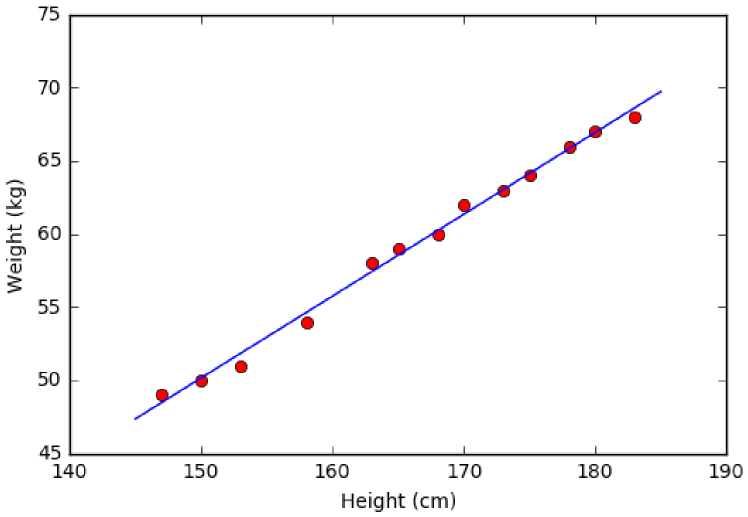
⇒We need to find the function f(x,W) subject to:

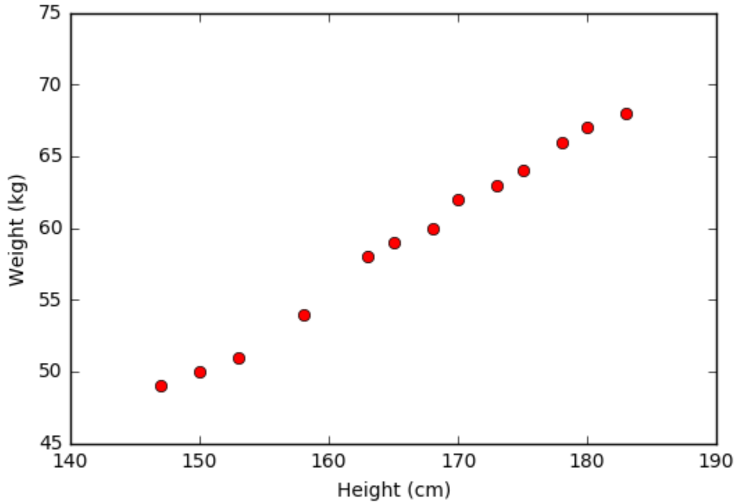
⇒ f(x, W) = y

⇒Loss function:

$$\Rightarrow L(w) = \frac{1}{2}\sum_{i=1}^{N}(y_i - x_i w)^2 = \frac{1}{2}\|y - X.w\|_2^2$$

⇒ find w to minimize L(w) by solving gradient equation of L(w).





Linear regression

- Ex: we have information of user related with person height and their weight.

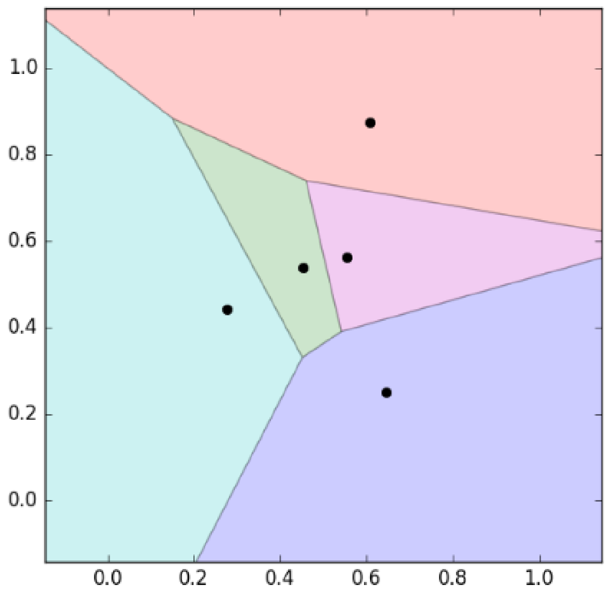
X_height = {147, 150, 153, 158, 163, 165, 170, 173, 175, 178, 180, 183}

Y_weight = {49, 50, 51, 54, 58, 59, 60, 62, 63, 64, 66, 67, 68}

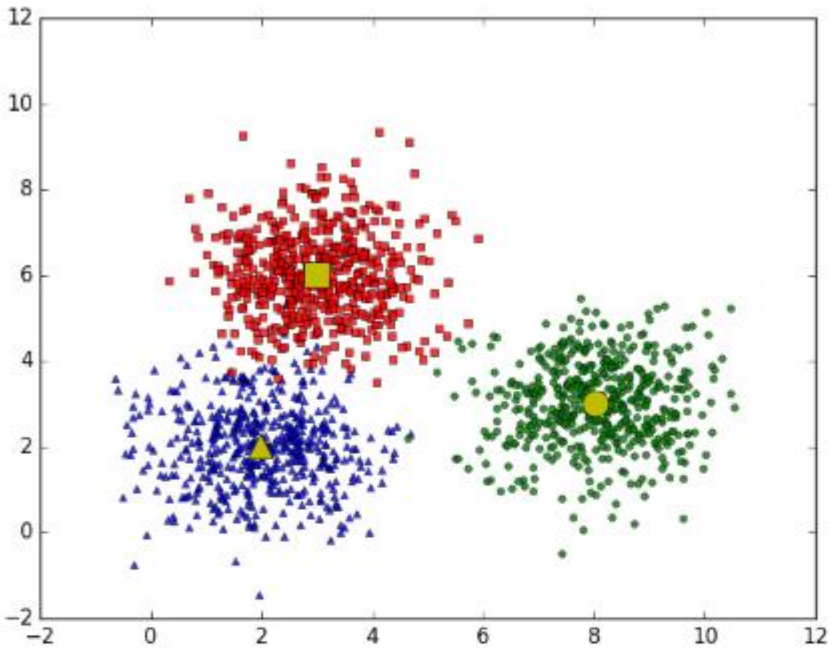
$\Rightarrow$Find out $\hat{Y}_{weight} \approx f(X_{height}, W)$

K-mean clustering

- The *‘means’* in the K-means refers to averaging of the data; that is, finding the centroid of cluster data.
- Unknown label of each data => how to divide data into clusters in the same categories



Separate regions with borders



Problem with 3 clusters

K-mean clustering

- We have N point data:

$$X = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{d\times N}$$

K < N, K is the number of clusters we desired to separated

$\Rightarrow$We need to find centers:

$$m_1, m_2, \dots, m_K \in \mathbb{R}^{d\times 1}$$

With each x_i we set $y_i = [y_{i1}, y_{i2}, \dots, y_{iK}]$ is label vector.

$\Rightarrow$If x_i in cluster k so $y_{ik} = 1$ and $y_{ij} = 0$, $\forall$ j $\neq k$

$\Rightarrow y_{ik} \in \{0,1\}$,

$\Rightarrow \sum_{k=1}^{K} y_{ik} = 1$

- Loss function

$$\|x_i - m_k\|_2^2 = y_{ik}\|x_i - m_k\|_2^2 = \sum_{j=1}^{K} y_{ij}\left\|x_i - m_j\right\|_2^2$$

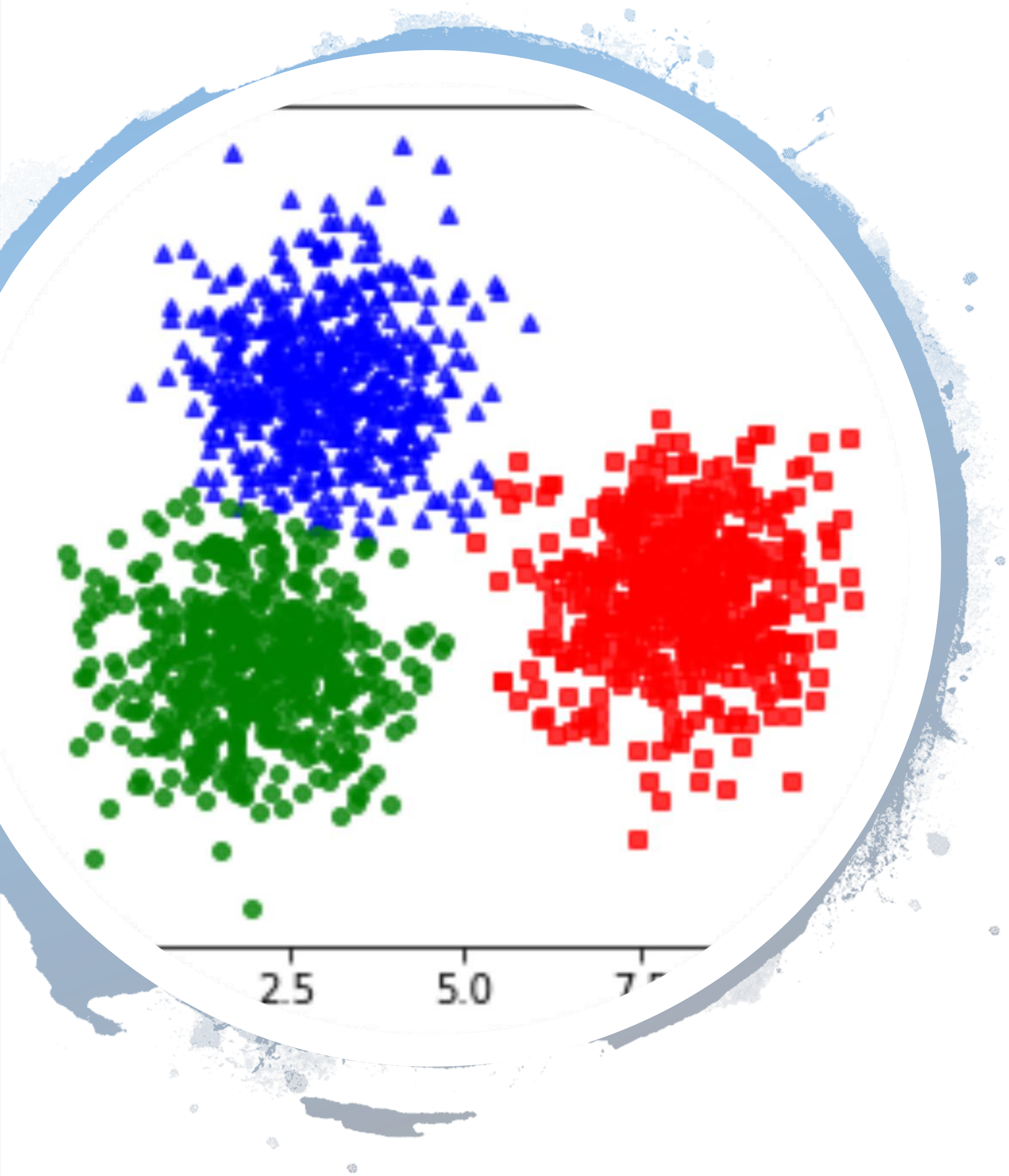
$\Rightarrow$ Loss for all data:

$$\mathrm{L}(Y, M) = \sum_{i=1}^{N} \sum_{j=1}^{K} y_{ij}\left\|x_i - m_j\right\|_2^2$$

$\Rightarrow$ Find **Y, M** subject to min loss function

K-mean clustering algorithms

- ***Input:*** data X and the number of clusters K
- ***Output:*** center M and labels vector in each data point Y
    1. Select random K for initialize center
    2. Divide each data point into nearest cluster
    3. If set data point in step 2 doesn't change with the previous iterations => stop algorithm
    4. Update center for each cluster by take the mean of all data points were set in cluster after step 2.
    5. Back to step 2



K-mean clustering

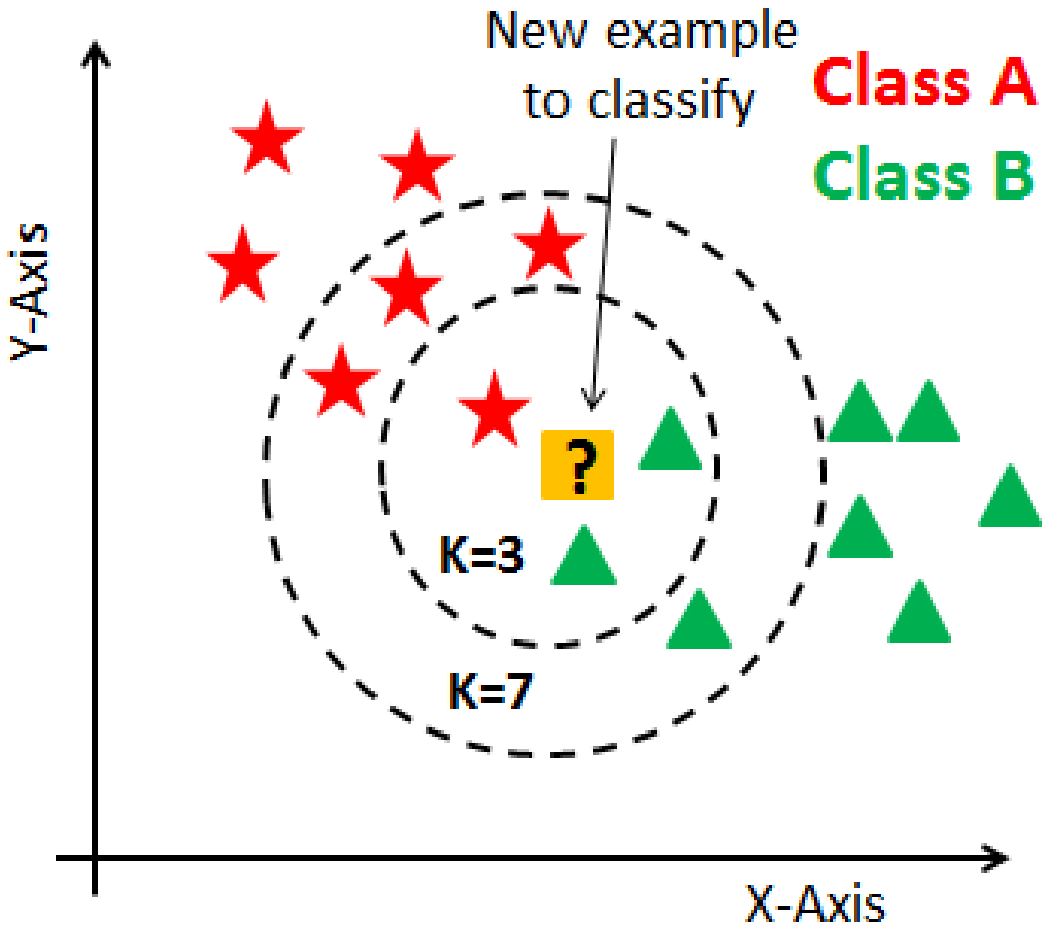
- Example (code)

K-mean clustering application

Cluster handwritten with MNIST dataset

Object segmentation

Image compression



K – nearest neighbor

- KNN – K nearest neighbors, is one of the simplest Supervised Machine learning algorithm mostly used for classification. It classifies a data point based on how its neighbors are classified.
- Given N training vectors, kNN algorithm identifies the k nearest neighbors of ‘c’, regardless of labels.

K-nearest neighbor

- Example

<source code>

Naïve Bayes classifier

- Problem with classification with C classes 1, 2, …, C. Assume we have a data point $x \in \mathbb{R}^d$. We will calculate the probability of this data point fall into class c:

$$p(y = c|s)$$

$$\Rightarrow c = arg \max_{c \in \{1,2,\dots,C\}} p(c|x) = argmax_c \frac{p(c|x)}{p(x)}$$

$$\Rightarrow c = argmax_c \frac{p(x|c).p(c)}{p(x)} \approx argmax_c p(x|c)p(c)$$

$$\Rightarrow c = arg \max_{c \in \{1,\dots,C\}} p(c) \prod_{i=1}^{d} p(x_i|c)$$

$\Rightarrow$ we take log for both side of equation:

$$\Rightarrow c = arg \max_{c \in \{1,\dots,C\}} = \log\big(p(c)\big) + \sum_{i=1}^{d} \log(p(x_i|c))$$

$\Rightarrow p(c)$ can be call as frequently appear of class c in training data.

$\Rightarrow p(x_i|c)$ depends of the type of data. We have 3 types popular data are Gaussian Naïve Bayes, Multinomial Naïve Bayes, and Bernoulli Naïve.

Naïve Bayes classifier

- ***Gaussian Naïve Bayes***

With each dimension I and a class c, x_i will follow a standard deviation with expectation μ_i and covariance σ_{ci}^2

$$p(x_i|c) = p(x_i|\mu_{ci}, \sigma_{ci}^2) = \frac{1}{\sqrt{2\pi\sigma_{ci}^2}} \exp\left(-\frac{(x_i - \mu_{ci})^2}{2\sigma_{ci}^2}\right)$$

With parameters $\theta = \{\mu_{ci}, \sigma_{ci}^2\}$ will be evaluated by Maximum Likelihood:

$$(\mu_{ci}, \sigma_{ci}^2) = arg \max_{\mu_{ci}, \sigma_{ci}^2} \prod_{n=1}^{N} p(x_i^{(n)}|\mu_{ci}, \sigma_{ci}^2)$$

Overfitting

- Appear when over fit with the training data. Moreover the model is too complicated for expression data.
- Example

  (code)

References

- http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises/ex6/ex6.html
- https://machinelearningcoban.com/2017/01/08/knn/
- https://machinelearningcoban.com/2017/08/08/nbc/

Schedule and plan

- *Time:* 30/Sept – 30/Nov
- *Machine learning:* 1/Oct – 18/Oct
- *Deep learning:* 21/Oct – 29/Nov
- 2 times per week, 2 hours per time.

Time	Contents
1 – 4/Oct	AI Introduction & requirements & applications Linear regression Overfitting K-nearest neighbors K-mean clustering Naïve Bayes classifier Gradient descent
7 – 11/Oct	Perceptron learning algorithm Logistic regression Softmax regression Multilayer neural network and Backpropagation Multilayer perceptron Support Vector machine

Time	Contents
14 – 18/Oct	Deep neural network Convolutional neural network: faster R-CNN, SSD (Single Shot MultiBox Detector, YOLO (You Look Only Once)
21 – 25/Oct	Deep neural network Convolutional neural network: faster R-CNN, SSD (Single Shot MultiBox Detector, YOLO (You Look Only Once)
28 – 1/Nov	Deep neural network Convolutional neural network: faster R-CNN, SSD (Single Shot MultiBox Detector, YOLO (You Look Only Once)
4 – 8/Nov	Convolutional neural network: faster R-CNN, SSD (Single Shot MultiBox Detector, YOLO (You Look Only Once)
11 – 15/Nov	Time Series: seq-to-seq modeling, RNN, LSTM, GRU Time Series: seq-to-seq modeling, RNN, LSTM, GRU Introduction to Reinforcement learning

THANK YOU SO MUCH!