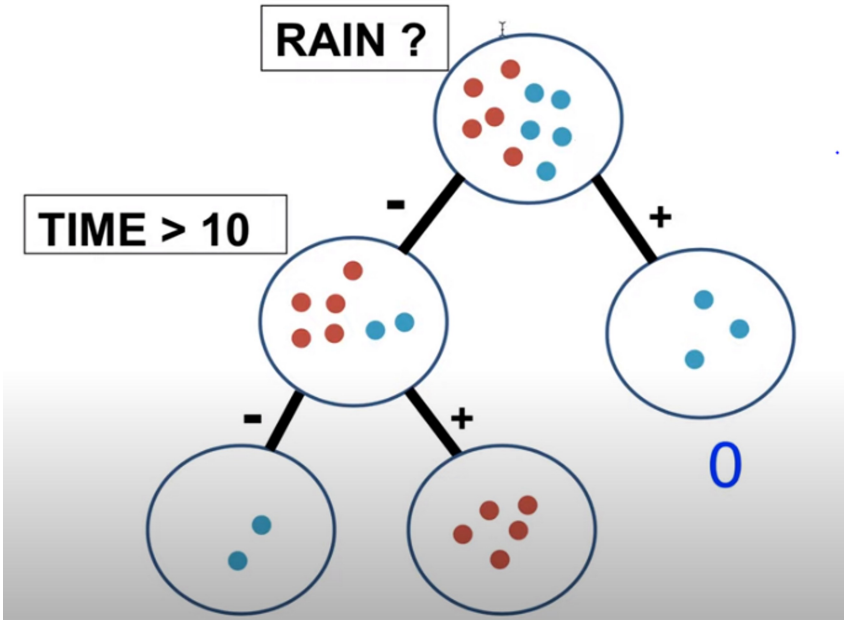
Tasks 9-13

14 January 2021 12:32 AM

Decision Tree

Rain	Time	Walk
1	30	NO
1	15	NO
1	5	NO
0	10	NO
0	5	NO
0	15	YES
0	20	YES
0	25	YES
0	30	YES
0	30	YES



Entropy

$$E = -\sum p(X) \cdot \log_2(p(X))$$

$$p(X) = \frac{\#x}{n}$$

Example

S = [0,0,0,0,0,1,1,1,1,1]

$$E = -\frac{5}{10} \cdot \log_2(\frac{5}{10}) - \frac{5}{10} \cdot \log_2(\frac{5}{10}) = -0.5 \log_2(-0.5) - 0.5 \log_2((-0.5) = 1$$

$$E = -0.5 \cdot (-1) - 0.5 \cdot (-1) = 1$$

Information Gain

$$IG = E(parent) - [weighted\ average] \cdot E(children)$$

Example

S = [0,0,0,0,0,1,1,1,1,1], S1 = [0,0,1,1,1,1,1], S2 = [0,0,0]

$$IG = E(S) - [(7/10) * E(S1) + (3/10) * E(S2)]$$

$$IG = 1 - [(7/10) * 0.863 + (3/10) * 0] = 0.395$$

Approach

Train algorithm := Build the tree

- Start at the top node and at each node select the best split based on the best information gain.
- Greedy search: Loop over all features and over all thresholds (all possible feature values).
- Save the best split feature and split threshold at each node.
- Build the tree recursively.
- Apply some stopping criteria to stop growing
  e.g. here: maximum depth, minimum samples at node, no more class distribution in node.
- When we have a leaf node, store the most common class label of this node

Predict := Traverse tree

- Traverse the tree recursively.
- At each node look at the best split feature of the test feature vector x and go left or right
  depending on x[feature_idx] <= threshold

When we reach the leaf node, we return the most stored class label

Random Forest

Logistic Regression

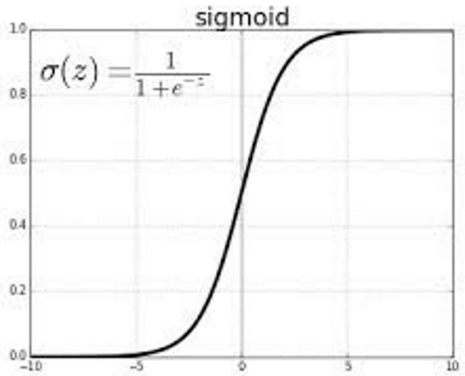
Approximation

$$f(w, b) = wx + b$$

$$\hat{y} = h_\theta(x) = \frac{1}{1 + e^{-wx+b}}$$

Sigmoid Function

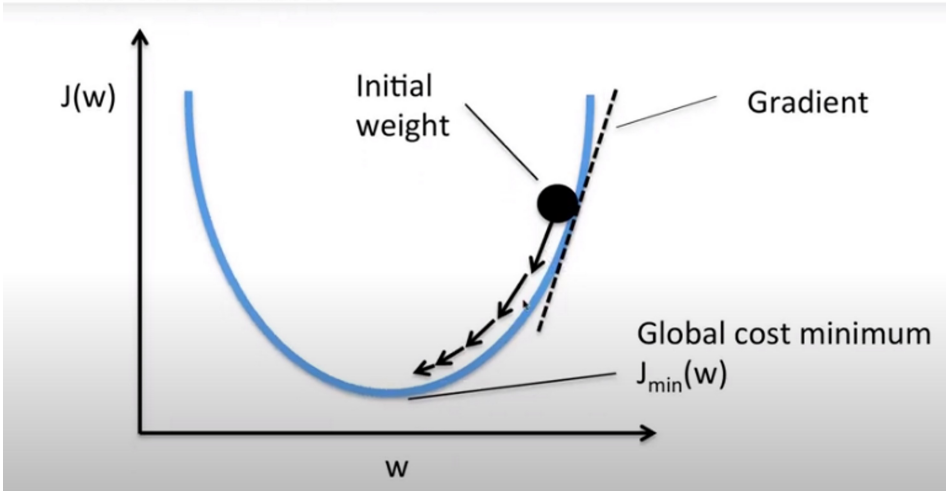
$$s(x) = \frac{1}{1 + e^{-x}}$$



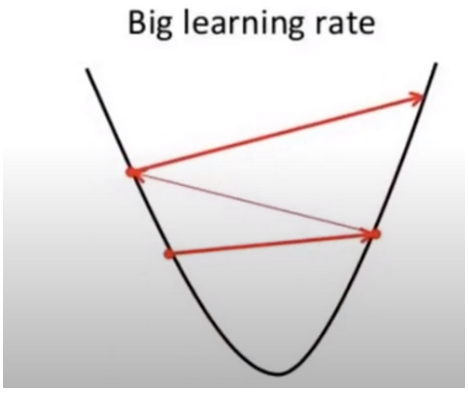
Cost function

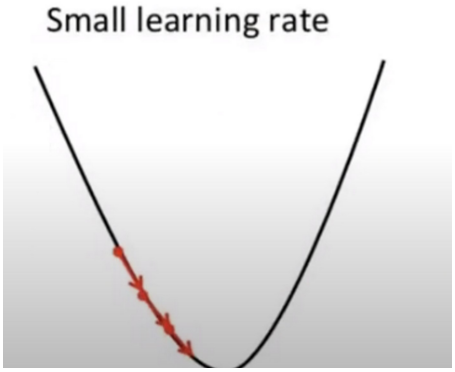
$$J(w, b) = J(\theta) = \frac{1}{N} \sum_{i=1}^{n} [y^i log(h_\theta(x^i)) + (1 - y^i) log(1 - h_\theta(x^i))]$$

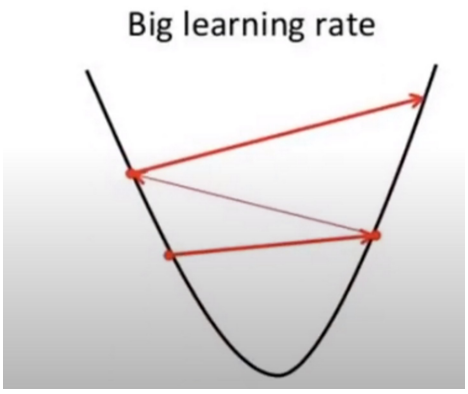
Gradient Descent

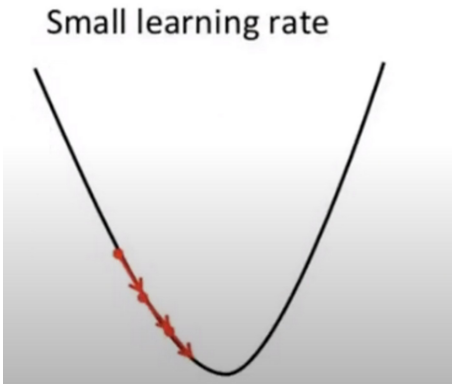


Learning Rate









Update Rules

$$w = w - \alpha \cdot dw$$
$$b = b - \alpha \cdot db$$

$$J'(\theta) = \begin{bmatrix} \frac{dJ}{dw} \\ \frac{dJ}{db} \end{bmatrix} = [\dots] = \begin{bmatrix} \frac{1}{N} \sum 2x_i(\hat{y} - y_i) \\ \frac{1}{N} \sum 2(\hat{y} - y_i) \end{bmatrix}$$

K Means

Goal

Cluster a data set into k different clusters. The data set is unlabeled (unsupervised learning).
Each sample is assigned to the cluster with the nearest mean.

Iterative optimization

1. Initialize cluster centers (e.g. randomly)
2. Repeat until converged:
   - Update cluster labels: Assign points to the nearest cluster center (centroid)
   - Update cluster centers (centroids): Set center to the mean of each cluster

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

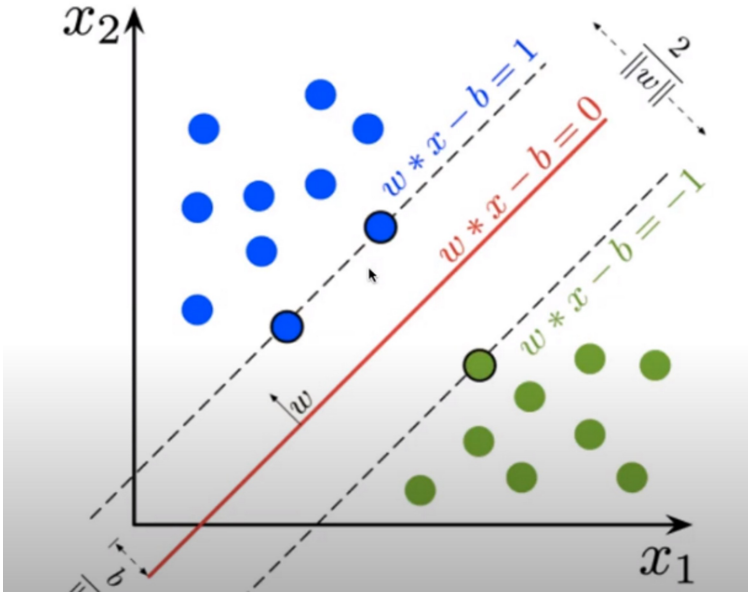
p, q = two points in Euclidean n-space

q_i, p_i = Euclidean vectors, starting from the origin of the space (initial point)

n = n-space

Image Source: Science Direct — Euclidean Distance Formula

SVM



Linear Model

$w \cdot x - b = 0$

$$\begin{aligned} w \cdot x_i - b &\geq 1 \quad \text{if } y_i = 1 \\ w \cdot x_i - b &\leq -1 \quad \text{if } y_i = -1 \end{aligned}$$

$y_i(w \cdot x_i - b) \geq 1$

Cost Function

Hinge Loss

$l = max(0, 1 - y_i(w \cdot x_i - b))$

$$l = \begin{cases} 0 & \text{if } y \cdot f(x) \geq 1 \\ 1 - y \cdot f(x) & \text{otherwise.} \end{cases}$$

Add Regularization

$$J = \lambda \|w\|^2 + \frac{1}{n} \sum_{i=1}^{n} max(0, 1 - y_i(w \cdot x_i - b))$$

if $y_i \cdot f(x) \geq 1$:

$$J_i = \lambda \|w\|^2$$

else:

$$J_i = \lambda \|w\|^2 + 1 - y_i(w \cdot x_i - b)$$

Gradients

if $y_i \cdot f(x) \geq 1$:

$$\frac{dJ_i}{dw_k} = 2\lambda w_k$$

$$\frac{dJ_i}{db} = 0$$

else:

$$\frac{dJ_i}{dw_k} = 2\lambda w_k - y_i \cdot x_i$$

$$\frac{dJ_i}{db} = y_i$$

Update Rule:

For each training sample xi:

w = w - a.dw

b = b - a.db

Naive Bayes

Bayes Theorem

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

In our case

$$P(y|X) = \frac{P(X|y) \cdot P(y)}{P(X)}$$

with feature vector X

$$X = (x_1, x_2, x_3, \ldots, x_n)$$

Assume that all features are mutually independent

Assume that all features are mutually independent

$$P(y|X) = \frac{P(x_1|y) \cdot P(x_2|y) \cdot \ldots \cdot P(x_n|y) \cdot P(y)}{P(X)}$$

Select class with highest probability

$$y = argmax_y P(y|X) = argmax_y \frac{P(x_1|y) \cdot P(x_2|y) \cdot \ldots \cdot P(x_n|y) \cdot P(y)}{P(X)}$$

$$y = argmax_y P(x_1|y) \cdot P(x_2|y) \cdot \ldots \cdot P(x_n|y) \cdot P(y)$$

$$y = argmax_y \log(P(x_1|y)) + \log(P(x_2|y)) + \ldots + \log(P(x_n|y)) + \log(P(y))$$

Prior probability P(y) : frequency

Class conditional probability P(x_i|y)

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \cdot exp(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2})$$