Machine Learning Techniques

Wee Hyong Tok, Ph.D.

October 8, 2018

Some Machine Learning References

- General
  - Jiawei Han, Data Mining: Concepts and Techniques, (The Morgan Kaufmann Series in Data Management Systems)
  - Tom Mitchell, *Machine Learning*, McGraw Hill, 1997
  - Christopher Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995

Overview

- Supervised and Unsupervised Machine Learning
- How machine learns
- Common Pitfalls in Machine Learning
- Linear Regression
- Logistic regression
- Naïve Bayesian Classifier (method required to solve homework)
- Lab in Python

Supervised vs. Unsupervised Learning

- Supervised learning: You have a set of independent variables X, and a target variable Y. Your machine learning task is to build a map from X → Y.
  - Classification and regression models both belong to supervised learning

- Unsupervised learning (clustering)
  - Class labels of the data are unknown
  - Given a set of data, the task is to establish the existence of classes or clusters in the data

Examples of Supervised Machine Learning

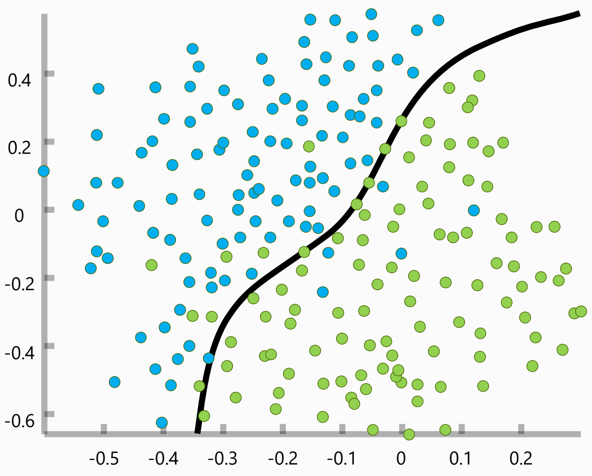
- Fraud transaction: we know which transactions in the training data were fraud (1), which were not (0)

- Readmission: we know which patients were readmitted to hospital within a certain time window after discharge

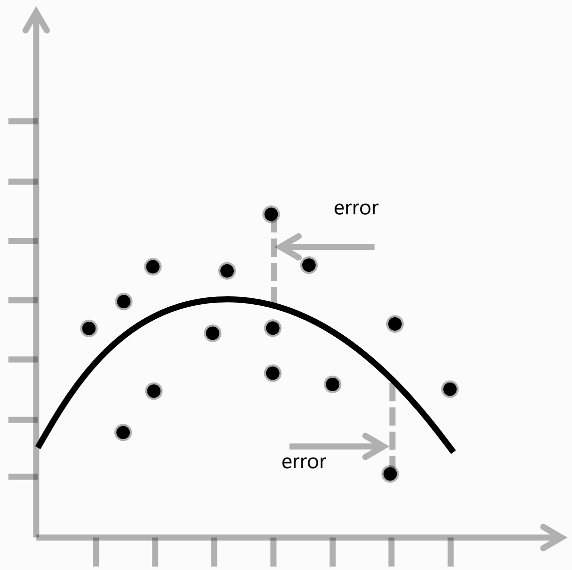
- Recommendation: we know which items were presented to customers, and which items were clicked, added to cart, or purchased.

Three typical supervised machine learning tasks: Classification, Regression and Recommendation

Classification



Regression



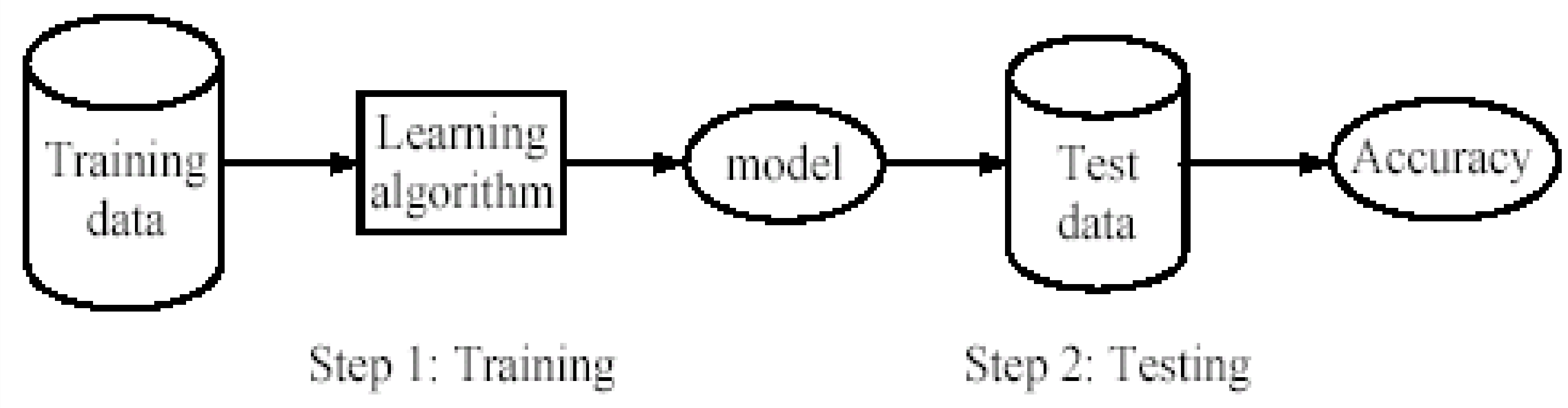
Recommenders

Supervised learning process: two steps

Learning (training): Learn a model using the training data

Testing: Test the model using unseen test data to assess the model accuracy

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$



Fundamental assumption of learning

Assumption: The distribution of training examples is identical to the distribution of test examples (including future unseen examples).

- In practice, this assumption is often violated to certain degree.
- Strong violations will clearly result in poor classification accuracy.
- To achieve good accuracy on the test data, training examples must be sufficiently representative of the test data.

The machine learning framework

$$y = f_\theta(\mathbf{x}) + \varepsilon$$

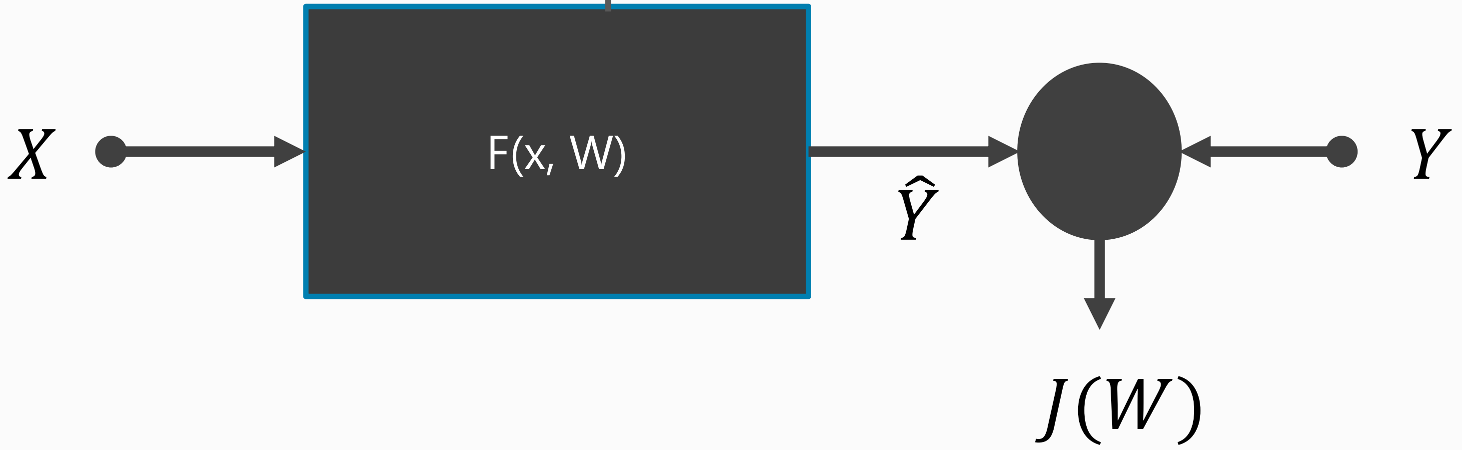
Observed dependent variable | prediction function | Independent variables | Random noise

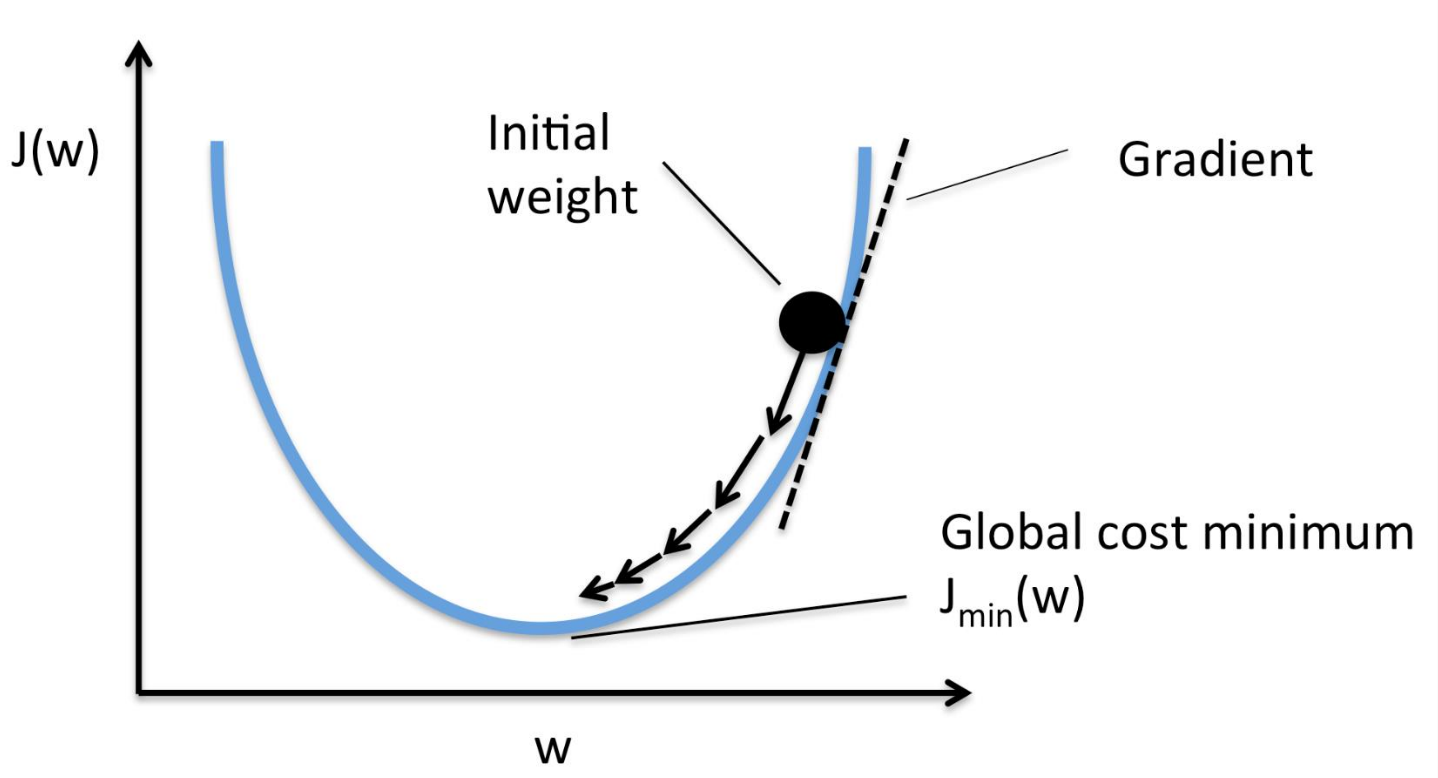
- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f and parameters θ which minimizes the prediction error on the training set

$$E_\theta(Y,X) = \sum_{i=1}^{N}\left(y_i - \widehat{f}_\theta(x_i)\right)^2$$

- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

How to learn model parameters θ?



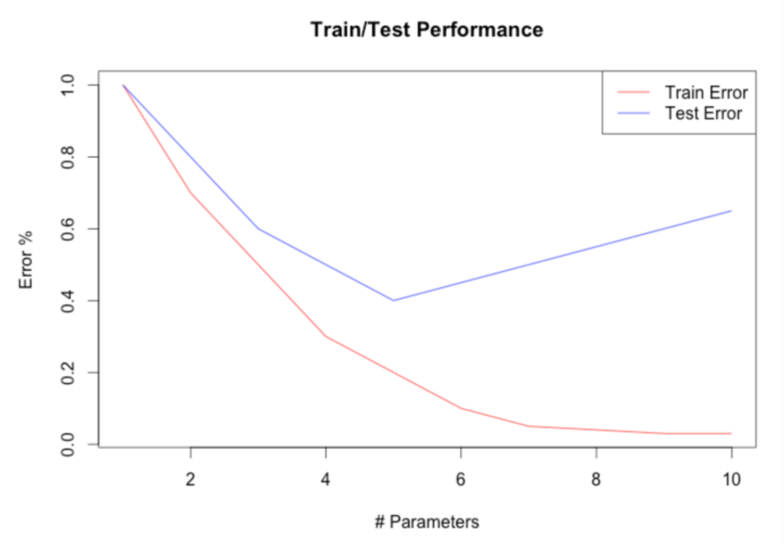


Generalization

- What does the model generalization mean?
  - We say a model generalizes well, meaning the model achieve similar performance on the training and validation data
  - We need to split the original dataset into training and validation, in order to test the generalization of models. Usually 70-80% in training, and remainder in validation.
- Underfitting: model is too "simple" to represent all the relevant class characteristics
  - High training error and high validation error
- Overfitting: model is too "complex" and fits irrelevant characteristics (noise) in the data
  - Low training error and high validation error (big gap between training and validation performance)

Common Pitfalls in Machine Learning

- Overfitting



- Target leaking
- Model has good performance on validation, but not applicable
  - Have to think about when the model is in production, whether you have data available for the variables of this model when prediction is made

Linear Regression (Review)

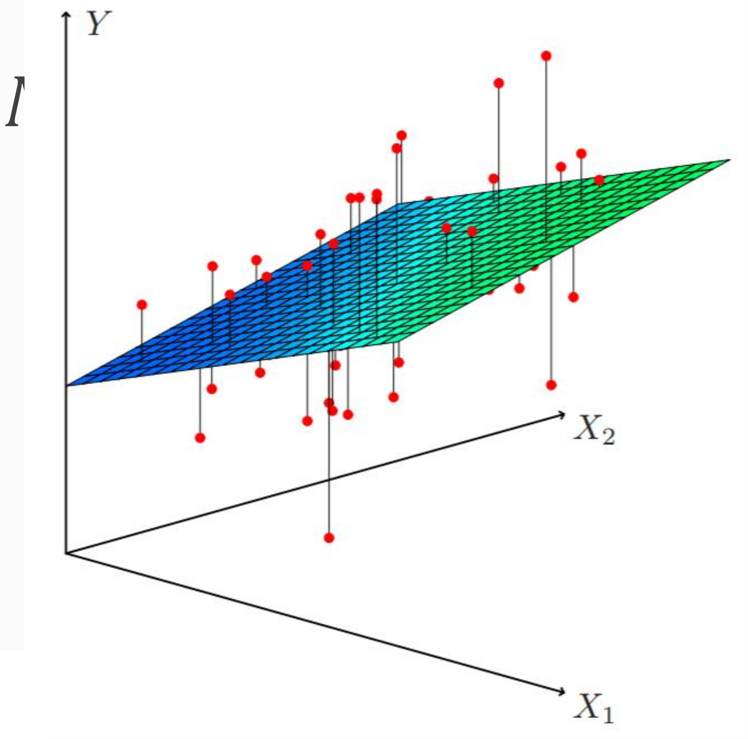
- Mathematical Equation

$$y_i = \boldsymbol{X}_i\boldsymbol{\theta} + \varepsilon_i, i = 1, 2, 3, \ldots N$$

$$\boldsymbol{\beta}=[\beta_0, \beta_1, \ldots, \beta_d]^{\boldsymbol{T}}$$

$$\boldsymbol{X}_i=[1, x_1^{(1)}, x_1^{(2)}, \ldots, x_1^{(d)}]^{\mathrm{T}}$$

$$\varepsilon \sim N(0, \sigma^2)$$



- Close form solution:

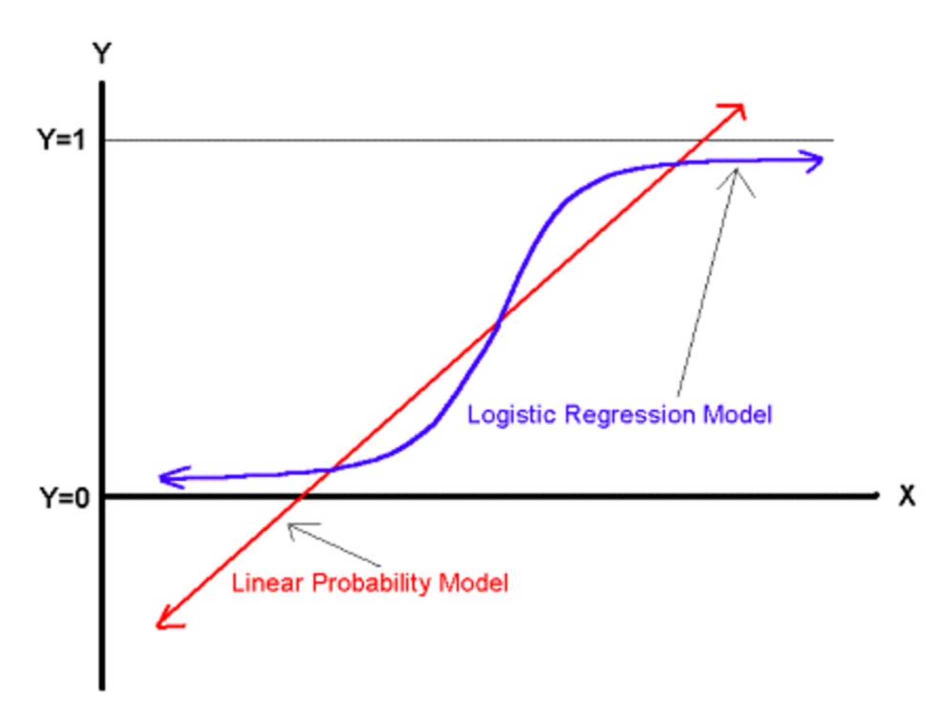
$$\boldsymbol{\theta} = (\boldsymbol{X}^{\top}\boldsymbol{X})^{-1}\boldsymbol{X}^{\top}\boldsymbol{y}$$

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1^{(1)} & \ldots & x_d^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(i)} & \ldots & x_d^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(n)} & \ldots & x_d^{(n)} \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

Logistic Regression

- Logistic regression (Review):

$$p = Prob(Y = 1)$$
$$\log\left(\frac{p_i}{1-pi}\right) = \boldsymbol{X}_i\boldsymbol{\beta} + \varepsilon_i$$
$$p_i = \frac{e^{\boldsymbol{\beta}\boldsymbol{X}_i}}{1 + e^{\boldsymbol{\beta}\boldsymbol{X}_i}}$$



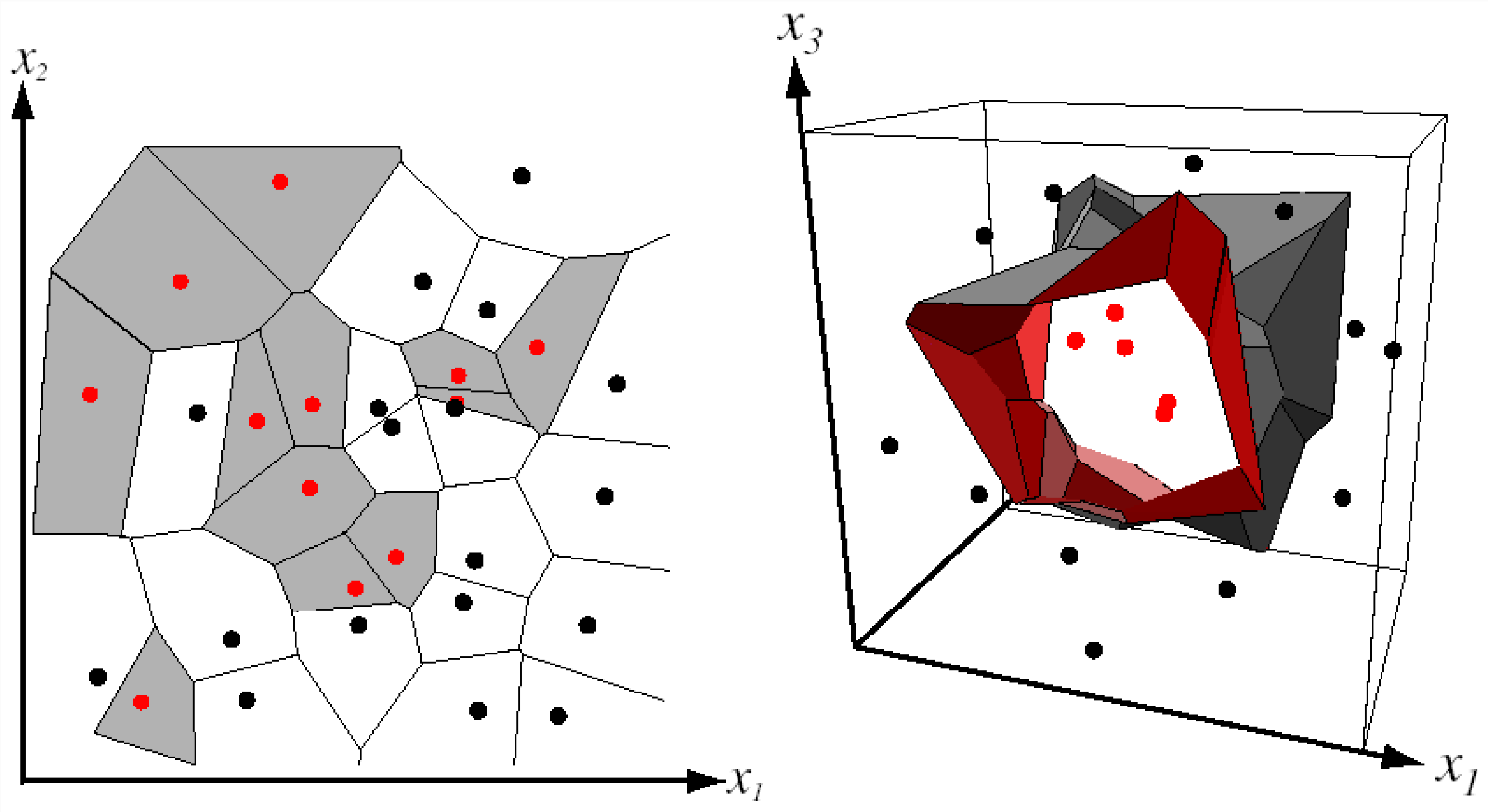
- Multiclass Logistic Regression: one vs others

$$p(y = c \mid \boldsymbol{x}; \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C) = \frac{\exp(\boldsymbol{\theta}_c^{\mathsf{T}} \boldsymbol{x})}{\sum_{c=1}^{C} \exp(\boldsymbol{\theta}_c^{\mathsf{T}} \boldsymbol{x})}$$

– Called the **softmax** function

Nearest Neighbor Classifier

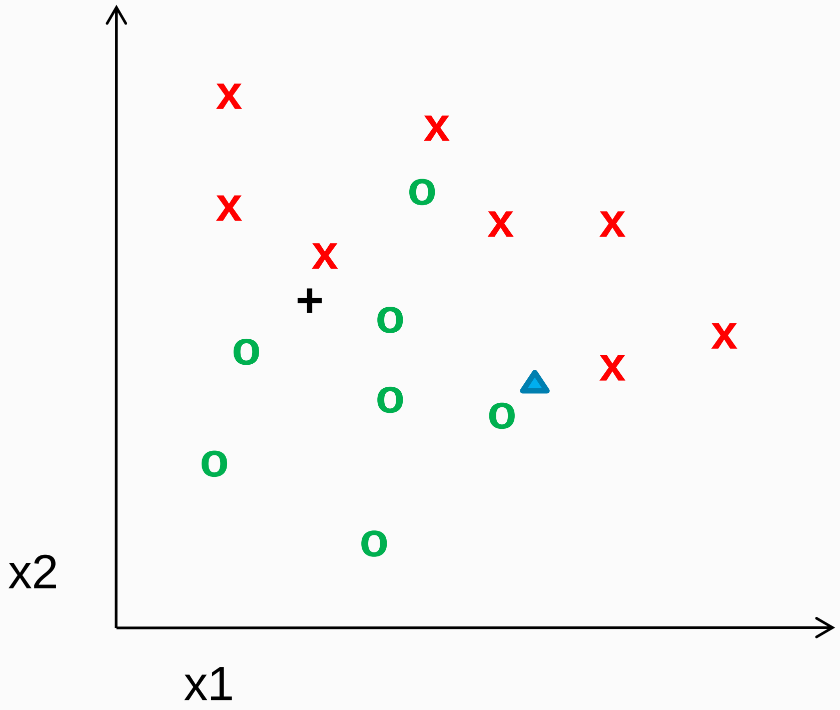
- Assign label of nearest training data point to each test data point



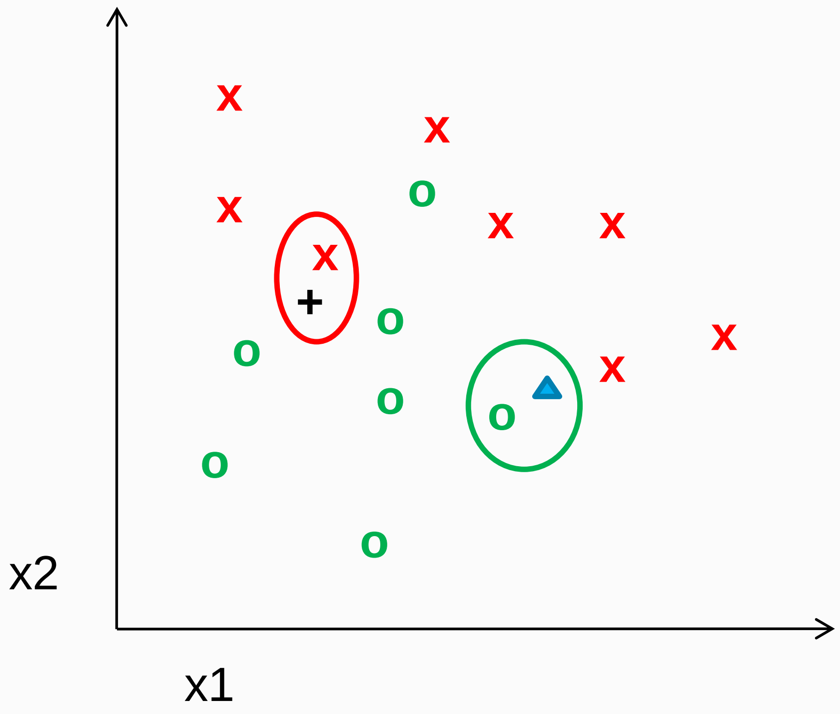
from Duda *et al.*

Voronoi partitioning of feature space
for two-category 2D and 3D data

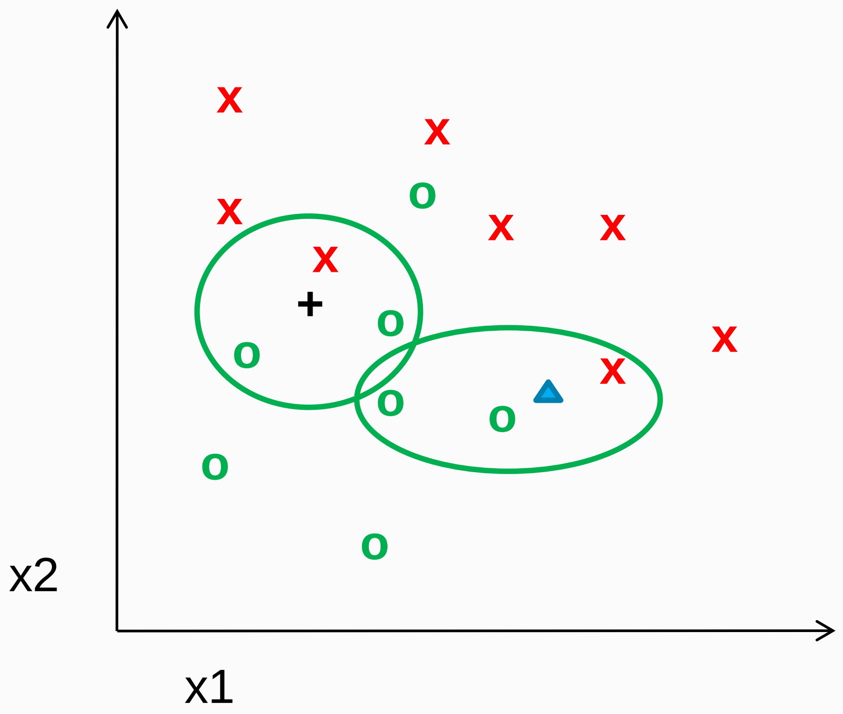
K-nearest neighbor



1-nearest neighbor



3-nearest neighbor



Using K-NN

- Simple, an easy one to implement
- Computationally intensive:
  - For N observations in training data, M observations in validation data, compute complexity is N*M.
- Sensitive to the choice of K and the similarity measurement
  - If K is small, very sensitive to noise in training data
  - If K is large, predicted value tends to be the mean of the training mean. Not helpful.
  - Similarity measurement matters as well.

Naïve Bayesian Classifier

$$Pr(yi = c|\boldsymbol{X}_i) = \frac{\Pr(yi = c, \boldsymbol{X}_i)}{\Pr(\boldsymbol{X}_i)}$$

$$= \frac{\Pr(\boldsymbol{X}_i|yi{=}c)\cdot\Pr(yi{=}c)}{\Pr(\boldsymbol{X}_i)}$$

- $\Pr(\boldsymbol{X}_i)$ is a common factor for all classes
- Only need to compare $\Pr(\boldsymbol{X}_i|y_i = c) \cdot \Pr(y_i = c)$

$$\boxed{Posterior = \frac{Likelihood \times Prior}{Evidence}}$$

How to Derive Likelihood from Data?

$$Pr(\boldsymbol{X}_i|y_i = c) = \Pr(x_{i1}, x_{i2}, \dots, x_{id}|\ y_i = c)$$

Naïve Bayesian Classification: Assuming independence among $x_{i1}, x_{i2}, \dots, x_{id}$ **condition on** $y_i = c$

$$\begin{aligned}\Pr(x_{i1}, x_{i2}, \dots, x_{id}|\ y_i = c) &= \Pr(x_{i1}|\ x_{i2}, \dots, x_{id}, y_i = c)\Pr(x_{i2}, \dots, x_{id}|\ y_i = c)\\ &= \Pr(x_{i1}|\ y_i = c)\ \Pr(x_{i2}, \dots, x_{id}|\ y_i = c)\\ &= \dots\\ &= \Pr(x_{i1}|\ y_i = c)\ \Pr(x_{i2}|\ y_i = c)\dots \Pr(x_{id}|\ y_i = c)\end{aligned}$$

NBC for Discrete Features

- **Algorithm: Discrete-Valued Features**
  - Learning Phase: Given a training set S of F features and L classes,

For each target value of c_i $(c_i = c_1, \cdots, c_L)$

$\hat{P}(c_i) \leftarrow$ estimate $P(c_i)$ with examples in S;

For every feature value x_{jk} of each feature x_j $(j = 1, \cdots, F; k = 1, \cdots, N_j)$

$\hat{P}(x_j = x_{jk} \mid c_i) \leftarrow$ estimate $P(x_{jk} \mid c_i)$ with examples in S;

Output: $F * L$ conditional probabilistic (generative) models

  - Test Phase: Given an unknown instance $\mathbf{x}' = (a'_1, \cdots, a'_n)$

"Look up tables" to assign the label c^* to $\mathbf{X}'$ if

$$[\hat{P}(a'_1 \mid c^*) \cdots \hat{P}(a'_n \mid c^*)]\hat{P}(c^*) > [\hat{P}(a'_1 \mid c_i) \cdots \hat{P}(a'_n \mid c_i)]\hat{P}(c_i), \quad c_i \neq c^*, c_i = c_1, \cdots, c_L$$

Example of NBC

- Tennis.csv

outlook	temp	humidity	windy	play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

Derive Conditional Probabilities, and Prior Probabilities of Each Features: Usage of Training Samples

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=Yes) = 9/14 \qquad P(\text{Play}=No) = 5/14$

Use the Learned Probabilities to Predict Testing Cases

- Consider a testing case:

X=(Outlook=Sunny, Temperature=Cool, Humidity=High, Wind=Strong)

$\Pr(y = play|Sunny, Cool, High, Strong) \propto$ Pr(Sunny|play)*Pr(Cool|play)*Pr(High|play)
*Pr(Strong|play)*Pr(play)
=2/9*3/9*3/9*3/9*9/14=0.00529

$\Pr(y = no\ play|Sunny, Cool, High, Strong) \propto$ Pr(Sunny|no play)*Pr(Cool|no play)*Pr(High|no play)
*Pr(Strong|no play)*Pr(no play)
=3/5*1/5*4/5*3/5*5/14=0.02057

So, we should assign label *No Play* to this condition.

Algorithm of NBC with Continuous Features

- Algorithm: Continuous-valued Features
  - Numberless values taken by a continuous-valued feature
  - Conditional probability often modeled with the normal

$$\hat{P}(x_j \mid c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(x_j - \mu_{ji})^2}{2\sigma_{ji}^2} \right)$$

μ_{ji} : mean (avearage) of feature values x_j of examples for which $c = c_i$

σ_{ji} : standard deviation of feature values x_j of examples for which $c = c_i$

$$\text{for } \mathbf{X} = (X_1, \cdots, X_F),\ C = c_1, \cdots, c_L$$

  - Learning Phase: $F \times L$ $\qquad P(C = c_i)\ i = 1, \cdots, L$

    Output: normal distributions and $\mathbf{X}' = (a'_1, \cdots, a'_n)$
  - Test Phase: Given an unknown instance
    - Instead of looking-up tables, calculate conditional probabilities with all the normal distributions achieved in the learning phrase
    - Apply the MAP rule to assign a label (the same as done for the discrete case)

NBC with Continuous Features Example

- Example: Continuous-valued Features
  - Temperature is naturally of continuous value.

    Yes: 25.2, 19.3, 18.5, 21.7, 20.1, 24.3, 22.8, 23.1, 19.8

    No: 27.3, 30.1, 17.4, 29.5, 15.1
  - Estimate mean and variance for each class

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad \sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu)^2$$

$$\mu_{Yes} = 21.64, \ \sigma_{Yes} = 2.35$$
$$\mu_{No} = 23.88, \ \sigma_{No} = 7.09$$

  - **Learning Phase**: output two Gaussian models for P(temp|C)

$$\hat{P}(x \mid Yes) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{2\times 2.35^2}\right) = \frac{1}{2.35\sqrt{2\pi}} \exp\left(-\frac{(x-21.64)^2}{11.09}\right)$$

$$\hat{P}(x \mid No) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{2\times 7.09^2}\right) = \frac{1}{7.09\sqrt{2\pi}} \exp\left(-\frac{(x-23.88)^2}{50.25}\right)$$

Zero conditional probability

- If no example contains the feature value
  - In this circumstance, we face a zero conditional probability problem during test

$$\hat{P}(x_1 | c_i) \cdots \hat{P}(a_{jk} | c_i) \cdots \hat{P}(x_n | c_i) = 0 \quad \text{for } x_j = a_{jk}, \ \hat{P}(a_{jk} | c_i) = 0$$

  - For a remedy, class conditional probabilities re-estimated with

$$\hat{P}(a_{jk} | c_i) = \frac{n_c + mp}{n + m}$$

(m-estimate)

n_c : number of training examples for which $x_j = a_{jk}$ and $c = c_i$

n : number of training examples for which $c = c_i$

p : prior estimate (usually, $p = 1/t$ for t possible values of x_j)

m : weight to prior (number of "virtual" examples, $m \geq 1$)

Zero conditional probability

- Example: P(outlook=overcast|no)=0 in the play-tennis dataset
  - Adding m "virtual" examples (m: up to 1% of #training example)
    - In this dataset, # of training examples for the "no" class is 5.
    - We can only add m**=1** "virtual" example in our m-esitmate remedy.
  - The "outlook" feature can takes only 3 values. So $p=1/3$.
  - Re-estimate P(outlook|no) with the m-estimate

$$P(\text{overcast}|\text{no}) = \frac{0+1*\left(\frac{1}{3}\right)}{5+1} = \frac{1}{18}$$

$$P(\text{sunny}|\text{no}) = \frac{3+1*\left(\frac{1}{3}\right)}{5+1} = \frac{5}{9} \qquad P(\text{rain}|\text{no}) = \frac{2+1*\left(\frac{1}{3}\right)}{5+1} = \frac{7}{18}$$