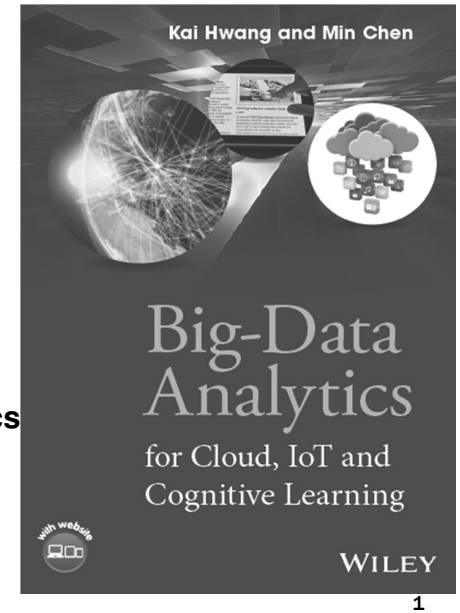
USC EE 542

Lectures 17 and 18,
Oct. 23 and 25, 2017

Lecture 17:
Linear and Logistic Regression Methods

Lecture 18:
Bayesian Classifiers and Social-Media Analytics

Professor Kai Hwang
University of Southern California



Reading Assignments in Chapter 6: Machine Learning (ML) Algorithms and Model Fitting

Sec. 6.1: Machine Learning Taxonomy

Sec. 6.2.1: Linear/Logistic Regression

Sec. 6.2.3: Supervised Bayesian Classifier

Sec. 6.3.1: Unsupervised Clustering

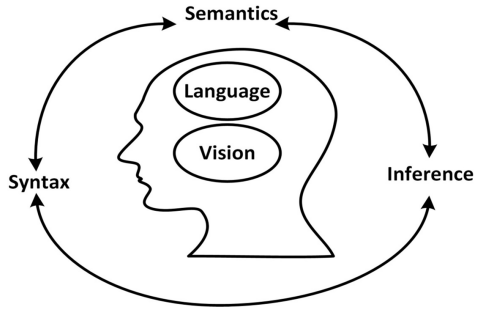
Sec. 6.4 : Model Fitting (lectures 23-24)

Basic Concept of Artificial Intelligence (AI)

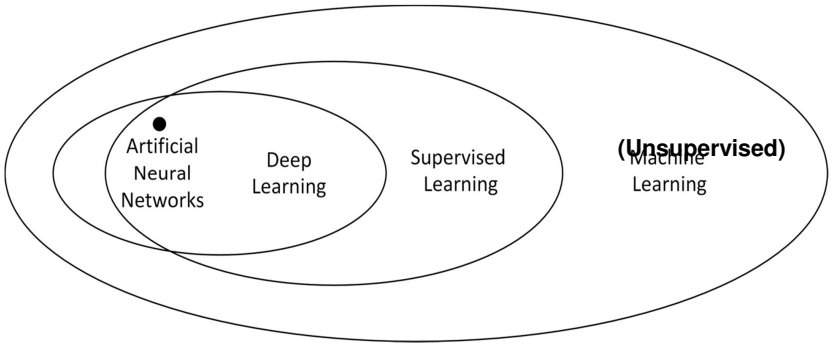
- AI has been a research field for at least 60 years ever since Alan Turing who suggested the very first "Thinking Machine" known as Turing Machines
- Intelligent machines must have two fundamental elements:
  - One is cognitive power such as 5 senses of cognition: Vision, hearing, taste, touch, and smell.
  - Another is language skill that human has to understand, express and comprehend things (Animals have none)
- Turing did not realize intelligent machines at his times, but he pioneered the key concept which we are now producing some smart machines with hardware, software and sensing devices

AI Pioneers: Marvin Minski (MIT) and Terry Minograd (Stanford)

Three Things that must interact to achieve AI: Syntax (structure), Semantics (Meaning), and Inference (reasoning). In fact, human intelligence also requires these three ingredients.



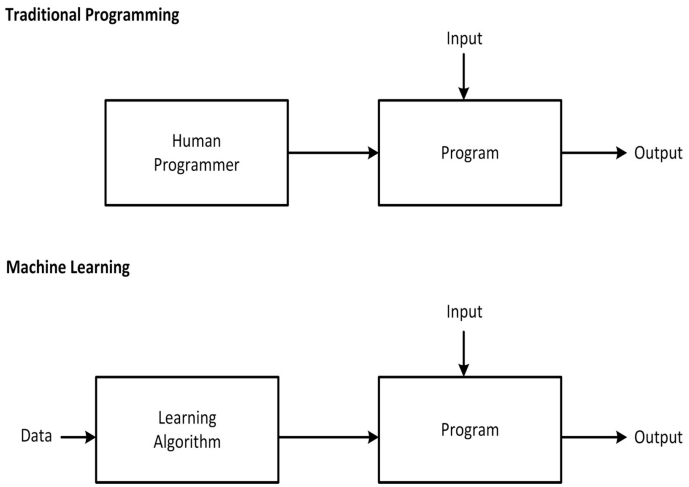
Evolution of Deep Learning from ML and The ANN Approach



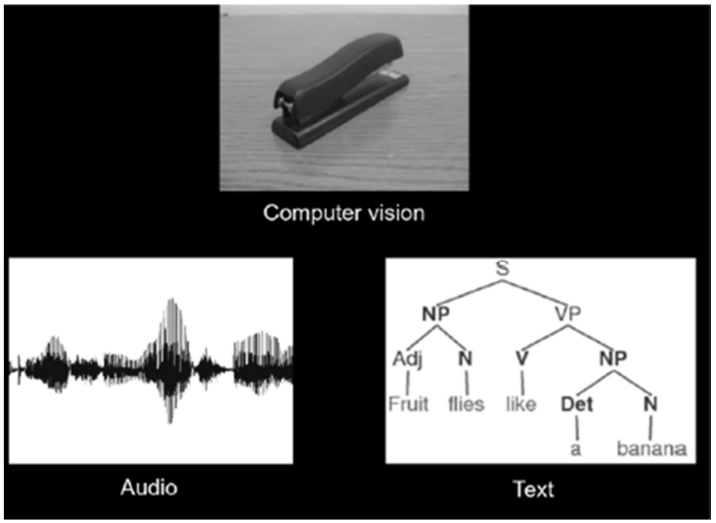
Basic Concept of Machine Learning

- Machine Learning is a scientific discipline that explores the construction and study of algorithms that can learn from data.
- Machine learning algorithms operate by building a model from executing example inputs and using that model to make predictions or decisions.
- Machine Learning is a subfield of computer science stemming from research into artificial intelligence. It has strong ties to statistics and mathematical optimization.

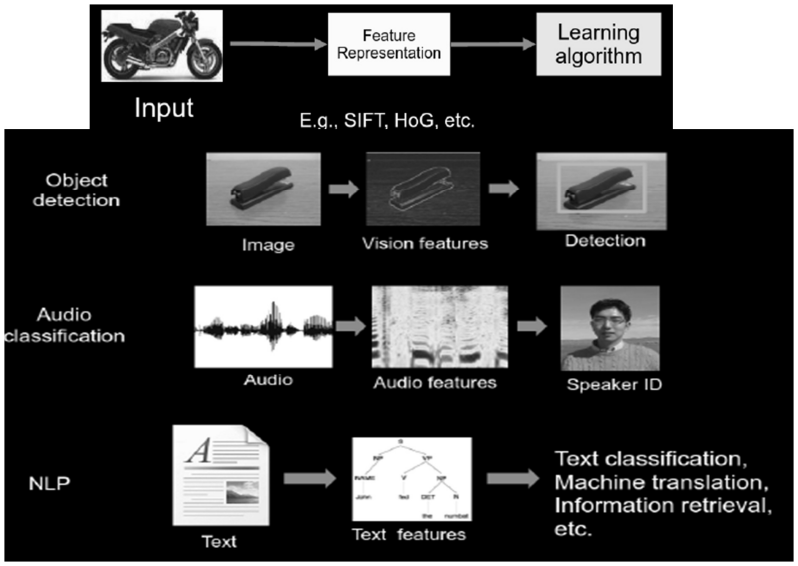
Traditional Programming Model versus Machine Learning Model



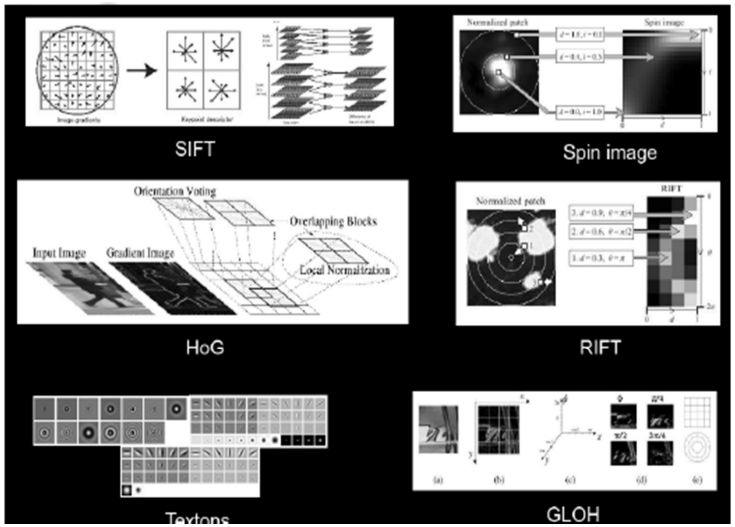
Computer Vision, Audio Recognition, and Text Understanding



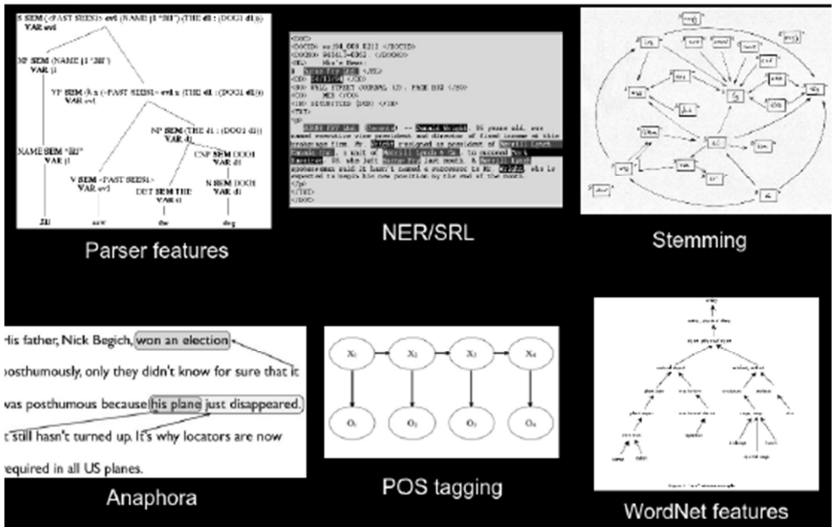
What is Computer Perception ?



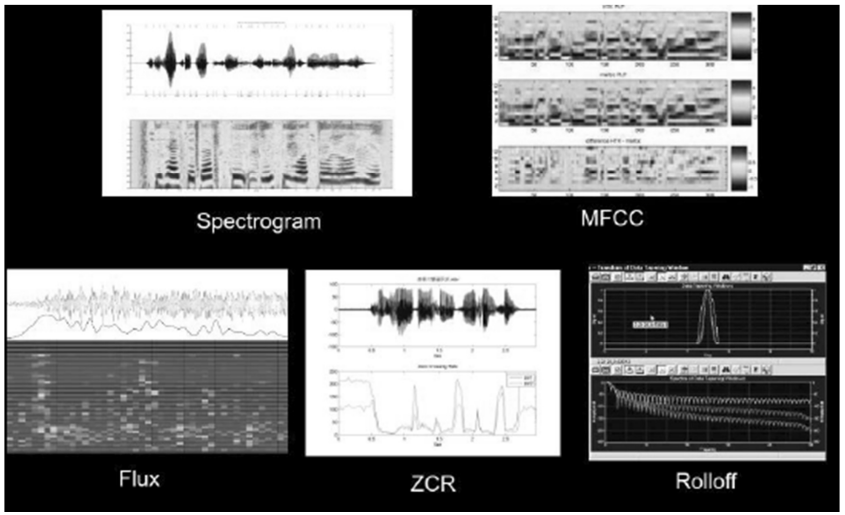
Computer Vision Features



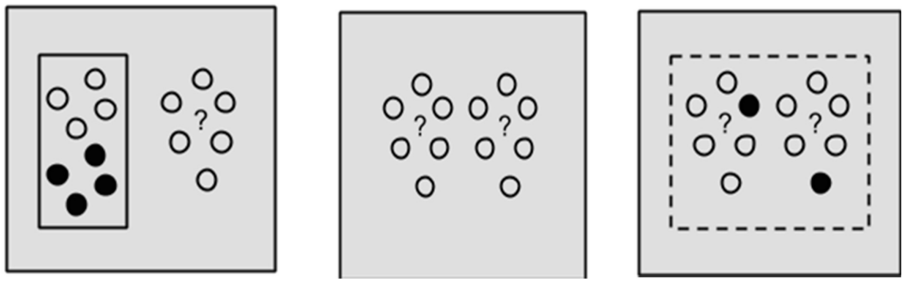
Audio Recognition Features



Nature Language Processing (NLP) Features



Machine Learning Algorithms

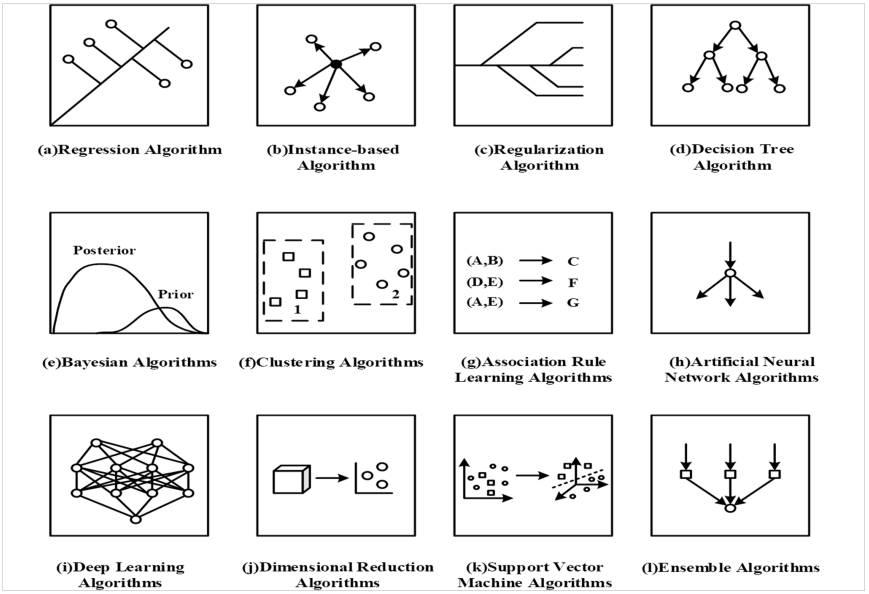


Supervised learning with Training Data

Unsupervised without Training Samples

Semi-Supervised with Partial Samples

12 Machine Learning Methods



Machine Learning Approaches: (1)

1. Decision Tree Learning : Using multi-way tree to make categorical decisions
2. Association Rule Learning : Discovering interesting relations between variables in large databases
3. Artificial Neural Networks (ANN) : Learning algorithm that is inspired by the structure and functional aspects of bilological neural networks
4. Inductive Logic Programming (ILP): Uisng logical programming to represent input data, background knowledge, and hypotheses

Machine Learning Approaches (2)

5. Support Vector Machines (SVM) : Using a set of related supervised learning methods for classification and regression.
6. Clustering Analysis : grouping sample data into clusters with similar properties or some predefined criteria.
7. Bayesian Networks : A belief network or a probablistic DAG model that represent a set of random variables and their conditional independencies.
8. Reinforcement Learning : Based on how an agent ought to take actions in an environment to maximize some notion of long-term reward.

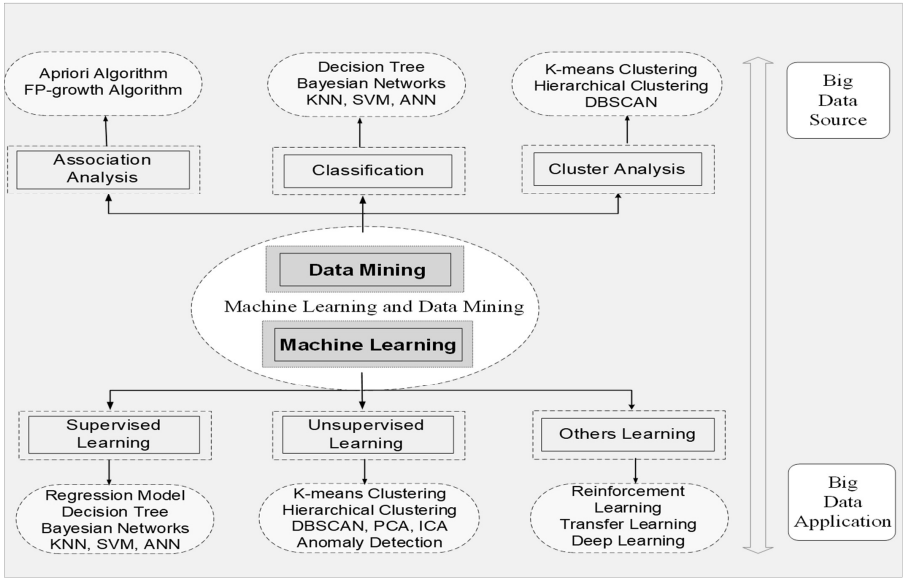
Machine Learning Approaches (3)

9. Representative Learning: Based on preserving the input information and transforming it as a pre-processing process for other classification algorithms.
10. Similarity and Metric Learning : To learn from a similarity function or distance metric function
11. Sparse Dictionary Learning : Each datum is represented as a linear combinayion of basic functions, and the cofficients are assumed sparse.
12. Genetic Algorithms (GA) : A research heuristic that mimics the process of natural selection and uses methods such as mutation and crossover to generate genotype towards making better decision

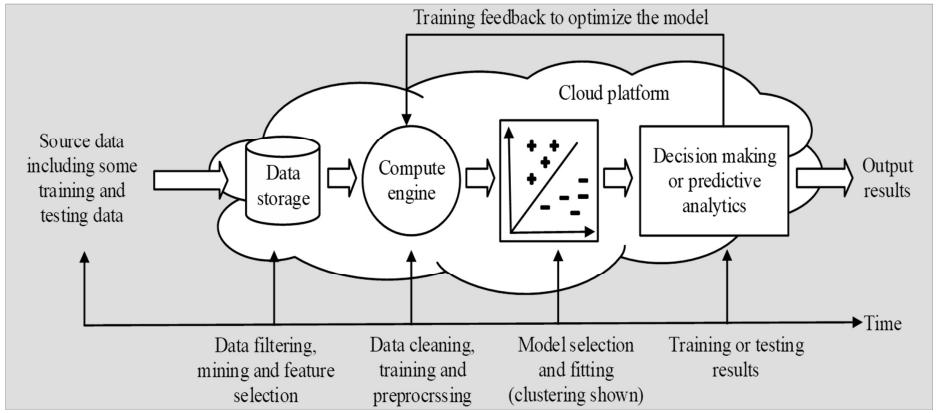
Machine Learning and Cloud Analytics,

- Machine Learning is a scientific discipline that explores the construction and study of algorithms that can learn from data.
- Machine learning algorithms operate by building a model from executing example inputs and using that model to make predictions or decisions.
- Machine Learning is a subfield of computer science stemming from research into artificial intelligence. It has strong ties to statistics and mathematical optimization.

Data Mining and Machine learning



Lecture 18: October 25, 2017
Model Fitting in Machine Learning



Machine Learning Pipeline Engine

Basic Concepts of Regression Analysis

- Regression analysis performs a sequence of parametric or non-parametric estimations. The method finds the causal relationship between the input and output variables. The estimation function can be determined by experience using a priori knowledge or visual observation of the data.

- Regression analysis is aimed to understand how the typical values of the output variables change, while the input variables are held unchanged.
  Thus regression analysis estimates the average value of the dependent variable when the independent variables are fixed.

The Model of A Linear Regression Process

Consider a set of data points in a 2-dimensional sample space. $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$. If they can be approximated by a straight line, then we obtain the following linear regression expression.

$$y = ax + b + \varepsilon \tag{7.2}$$

where x stands for an explanatory variable, y is a continuous variable in the real number range, a and b are corresponding coefficients, and ε is a random error, which follows an independent normal distribution. One needs to figure out the expectation by using a linear regression expression $y = ax + b$ illustrated in Fig.7.3.

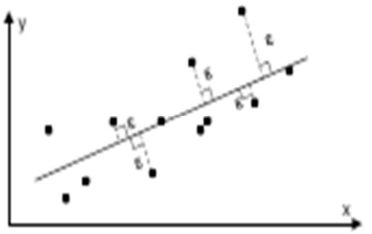


Figure 7.3: Linear regression analysis with a single input variable

Three Cases To be Modeled in A Regression Process

- When $N < k$, most classical regression analysis methods can be applied. Since the defining equation is underdetermined, there are not enough data to recover the unknown parameters β.
- When $N = k$ and the function f is linear, the equations $Y = f(X, \beta)$ can be solved exactly without approximation, because there are N equations to solve N components in β. The solution is unique as long as the X components are linearly independent. If f is nonlinear, many solutions may exist or no solution at all.
- In general, we have the situation that $N > k$ data points. This implies that there is enough information in the data that can estimate a unique value for β under an overdetermined situation.

Consider an example to toss a small ball in the air. We measure its heights of ascent h at various time instant t. The relationship is modeled as

$$h = \beta_1 t + \beta_2 t^2 + \varepsilon$$

where β_1 determines the initial velocity of the ball, β_2 is proportional to the standard gravity, and ε is due to the measurement error.

Here, linear regression is used to estimate the values of β_1 and β_2 from the measured data. This model is non-linear in time variable t, but it is linear with respect to the unknown parameters β_1 and β_2.

Logistic Regression Method: This is a linear regression analysis model extended to a broader application for prediction and classification, it is commonly used in fields such as

Logistic Regression Method:

This is a linear regression analysis model extended to a broader application for prediction and classification, it is commonly used in fields such as data mining, automatic diagnosis for diseases and economical prediction.

The logistic model may only be used to solve problem of dichotomy. As for logistic classification, the principle is to conduct classification to sample data with a logistic function, known as a sigmoid function defined by :

$$f(x) = 1 / (1 + e^{-x})$$

The input domain of the sigmoid function is in the range (0, 1). In this sense, the sigmoid function is a probability density function for the sample data shown in Fig,7.4.

Logistic Regression for Classification

The basic idea of logistic regression is to consider vector *x* with m independent input variables . Each dimension of *x* stands for one attribute (feature) of the sample data (training data). In logistic regression, multiple features of the sample data are combined into one feature by using linear function.

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \quad \beta_m x_m$$

We need to figure out the probability of the feature with designated data and apply the sigmoid function to act on that feature. We obtain the logistic regression as plotted in Fig. 7.5.

$$\begin{cases} P(Y=1 \mid x) = \pi(x) = \dfrac{1}{1+e^{-z}} \\ z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \quad \beta_m x_m \end{cases} \rightarrow \begin{cases} x \in 1 \text{ , } if \;\; P(Y=1 \mid x) > 0.5 \\ x \in 0 \text{ , } if \;\; P(Y=0 \mid x) < 0.5 \end{cases}$$

Logistic Regression Method

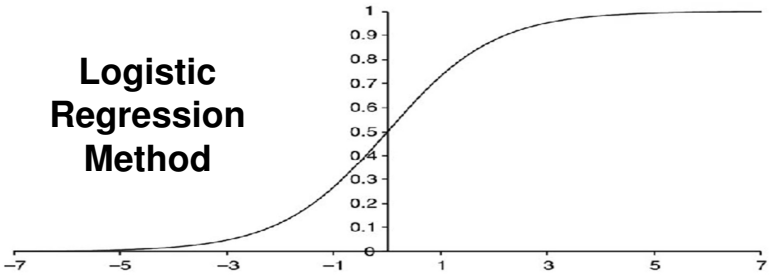


Figure 3.2 Bounding Function for Logistic Regression

For every possible value of z, the outcome is always between 0 and 1. Hence, by combining the linear regression with the bounding function, we get the following logistic regression model:

$$P(response = yes|age, income, gender) = \frac{1}{1+e^{-(\beta_0+\beta_1 age+\beta_2 income+\beta_3 gender)}}$$

The outcome of the above model is always bounded between 0 and 1, no matter what values of age, income, and gender are being used, and can as such be interpreted as a probability.

The general formulation of the logistic regression model then becomes:

$$P(Y=1|X_1,\ldots,X_n) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\quad +\beta_N X_N)}},$$

or, alternately,

$$P(Y=0|X_1,\ldots,X_N) = 1 - P(Y=1|X_1,\ldots,X_N)$$
$$= 1 - \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\quad +\beta_N X_N)}} = \frac{1}{1+e^{(\beta_0+\beta_1 X_1+\quad +\beta_N X_N)}}$$

Hence, both $P(Y=1|X_1,\ldots,X_N)$ and $P(Y=0|X_1,\ldots,X_N)$ are bounded between 0 and 1.

The Sigmoid Function

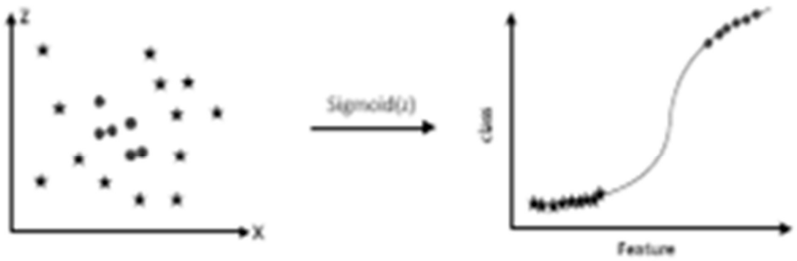


Figure 7.5: Fundamental concept of using logistic regression for classification purpose.

Four Clustering Techniques

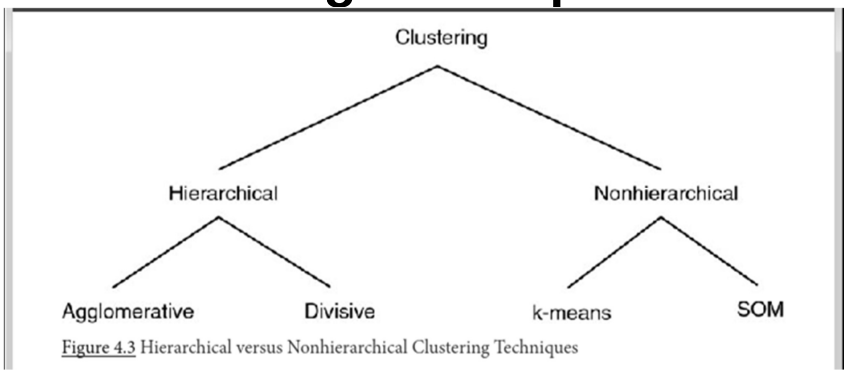


Figure 4.3 Hierarchical versus Nonhierarchical Clustering Techniques

Agglomerative and Divisive work in opposite directions over subdivided clustering steps, K-means apply initial seeds as centroids and eventually converge to K centroids. SOM : Self-Organizing Maps are feedfoward neural networks

kNN and k-Mean Clustering

k-Nearst Neighbor (kNN) Clustering:

- This is a kind of *lazy learning* method or a type of *instance-based* learning which is the simplest to implement. The objective function is only approximated locally with a deferred classification.
- The idea is to consider the input data elements as among the *k* closest training examples in the feature space. The output depends on whether kNN is used for classification or for regression, as defined below:
- For *kNN classification*, an object is classified by a majority vote of its neighbors, meaning the element being classified as a member by the most common among its *k* nearest neighbors.
- *For kNN regression*, the output is the property value for the data object, which is the average of the values of its *k* nearest neighbors. This means that the data object is weighted by the nearer neighbors.
-

Divisive vs. Agglomerative Hierarchical Clustering

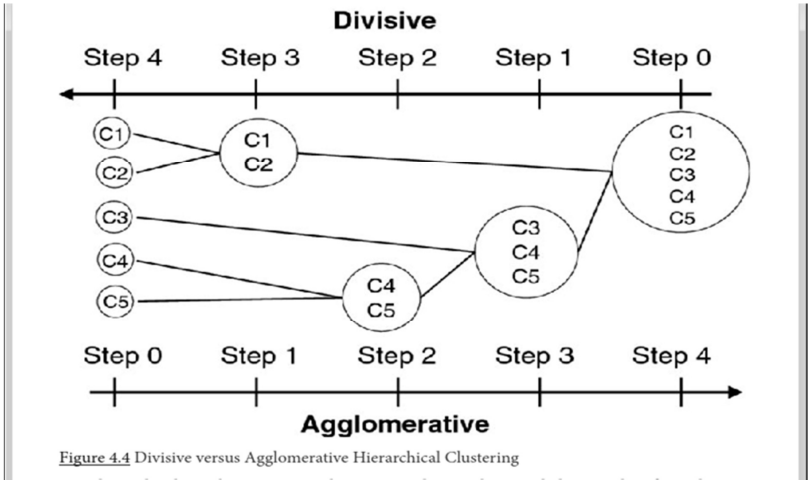
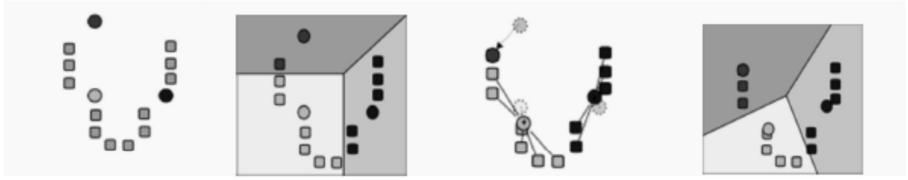


Figure 4.4 Divisive versus Agglomerative Hierarchical Clustering

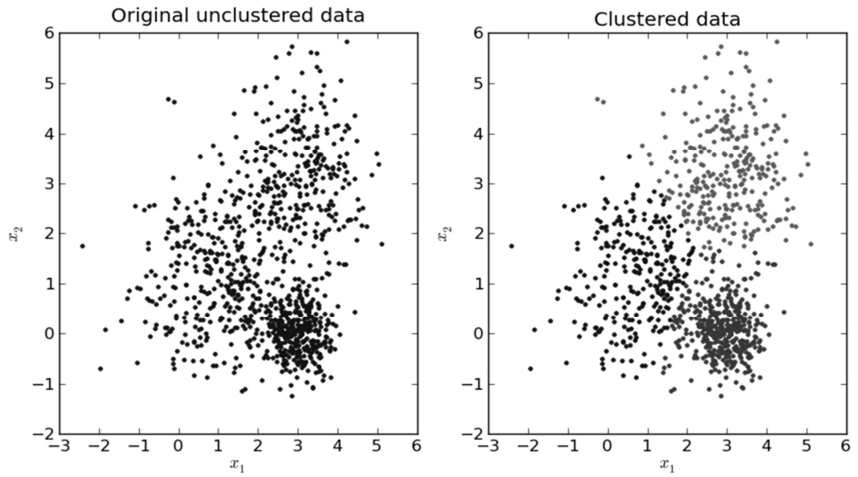
Concept of k-mean Clustrering



(a) Choose 3 initial means for k = 3 (b) 3 initial clusters with the nearest mean (c) Centroids in 3 clusters become new means (d) After repeating steps 2 and 3 to converge

Figure 7. 8: Four steps to generate 3 clusters out of 15 data elements (Reprint with permission from Wikipedia ??????????????????????????????????)

Cluster Analysis



k-Mean Clustering Method: The objective of K- means is described as follows. For a given data set and given

k

find a group of clusters

$C_1, C_2, \ldots, C_k$

in ordering to minimize the objective function

$$E = \sum_{c=1}^{k} \sum_{x \in C_c} [d(x, \bar{x}_{C_c})]^2$$

$$\min E = \min \sum_{c=1}^{k} \sum_{x \in C_c} [d(x, \bar{x}_{C_c})]^2 \quad (6\text{-}29)$$

Example 6.6: Using k-mean Clustering to Classify the Iris Flowers (contributed by Wenhao Zhang, USC 2016)

- Consider solving an iris flower classification problem with *k*-mean clustering for *k* =3 clusters. Given a data set of 150 unlabeled data points on Iris flowers. These flowers are classified into three clusters, namely *Iris Setosa*, *Iris Versicolour,* and *Iris Virginica*.
- The complete flower data set, denoted as "iris.csv" is from UC Irvine Machine Learning Repository. There are 4 features in this dataset: *(1) sepal length in cm, (2) sepal width in cm, (3) petal length in cm, a nd (4) petal width in cm.*
- To simplify the final display of the clustering results, we consider the data points with only two most important features {3, 4}. Identify the clustering centers (centroids) in successive steps, and also draw a 2-D Euclidean space diagram (x-axis: petal.length; y-axis: petal.width) to show the final clustering results.


```
inputData <- read.csv("./iris.csv")          //Read the input dataset
initialCenters <-matrix(c(2, 4, 6, 0.8, 1.6, 2.5), 3, 2)   // Select initial means of 3 clusters.
results <- kmeans(inputData, initialCenters, iter.max = 1, trace=TRUE)          // First iteration
firstCenters <- results$centers          //# save the clustering centers after one iteration
results <- kmeans(inputData, initialCenters, iter.max = 2, trace=TRUE)          // Second iteration
plot(inputData, pch = results$cluster)     // Plot the partitioning results after it converges
points(initialCenters, pch = 16, cex = 2)   // Plot the initial centers
points(firstCenters, pch = 15, cex = 2)     //Plot the centers after 1st iteration
points(results$centers, pch = 17, cex = 2)  //Plot the centers after 2nd iteration , convergence
```


Bayesian Classifiers (1)

- Consider each attribute and class label as random variables
- Given a record with attributes (A_1, A_2,…,A_n)
  - Goal is to predict any class denoted as C
  - Specifically, we want to find the value of C that maximizes P(C| A_1, A_2,…,A_n)
- Can we estimate P(C| A_1, A_2,…,A_n) directly from data?

Bayes Classifier (2)

- A probabilistic framework for solving the classification problems
- Conditional Probabilities:

$$P(C \mid A) = \frac{P(A,C)}{P(A)}$$

$$P(A \mid C) = \frac{P(A,C)}{P(C)}$$

- Bayes theorem:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

Bayesian Classifiers (3)

- Compute the posterior probability P(C | A_1, A_2, …, A_n) for all values of C using the Bayes theorem

$$P(C \mid A_1 A_2 \quad A_n) = \frac{P(A_1 A_2 \quad A_n \mid C) P(C)}{P(A_1 A_2 \quad A_n)}$$

  - Choose value of C that maximizes P(C | A_1, A_2, …, A_n)
  - Equivalent to choosing value of C that maximizes P(A_1, A_2, …, A_n|C) P(C)
  - How to estimate P(A_1, A_2, …, A_n | C)?

Naïve Bayes Classifier (4)

- Assume statistical independence among all n attributes A_i when class C_j is given:
  P(A_1, A_2, …, A_n |C_j)
  = P(A_1| C_j) P(A_2| C_j)… P(A_n| C_j)
  - Can estimate P(A_i| C_j) for all A_i and C_j from the sample data points with labels.
  - New testing data point is classified to class C_j , if P(C_j) P(A_1| C_j) P(A_2| C_j)… P(A_n| C_j) is maximal.

Estimate Probabilities from Data (5)

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

- Class: P(C) = N_c/N

  P(No) = 7/10, P(Yes) = 3/10

- For discrete attributes:

  P(A_i | C_k) = |A_{ik}|/ N_c

  - where |A_{ik}| is number of k instances having attribute A_i belonging to class C_k
  - Examples:

    P(Status=Married | No) = 4/7

    P(Refund=Yes | Yes)= 0

Naïve Bayes Classifier (6)

- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

$$\text{m - estimate}: P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$$

c: Number of classes

p: Prior probability

m: Parameter

Example of Bayesian Classifier (7)

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Consider an unlabeled testing data item characterized by an attribute vector: $A^* = < A1, A2, A3, A4> = <yes, no, yes, no>$.

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

Table 6.6. Pre-Test Attribute Probability for Sample Data in Table 6.5

Attributes / Statistics		Give Birth		Can Fly		Live in Water		Have Legs	
		Yes	No	Yes	No	Yes	No	Yes	No
Counts	M	6	1	6	1	2	5	2	5
	N	1	12	10	3	10	3	4	9
Probability	M	6/7	1/7	6/7	1/7	2/7	5/7	2/7	5/7
	N	1/13	12/13	10/13	3/13	10/13	3/10	4/13	9/13

Example of Naïve Bayes Classifier (8)

A: attributes; M: mammals; N: non-mammals

$$P(A \mid M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A \mid N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A \mid M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A \mid N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

P(A|M)P(M) = 0.06 x (7/20) = 0.021 > P(A|N)P(N) = 0.0042 x (13/20) = 0.0027

Predicted Outcome: Mammals

Now, let us analyze the accuracy of using the Baysian classifier by testing 4 creatures using the above method. we list the following results as listed below. We obtain the posterior probabilities P(M | A1, A2, A3, A4) and P(N | A1, A2. A3, A4) for each of the 4 testing animals. Choose the class with highest probability as the predicted class.

Animal Name	Give Birth	Can Fly	Live in Water	Have Legs	Predicted Class	Actual Class	Prediction Status
Dog	yes	no	no	yes	M	M	TP
Monostream	no	no	no	yes	N	M	FN
Alligator	no	no	yes	yes	N	N	TN
Horse	yes	no	no	yes	M	M	TP

Comparing the predicted results with the actual classes, we discover 4 possible prediction statuses at the rightmost column. The TP (true positive) refers a true case correctely predicted, TN (True Negative) for a true case incorrectly predicted, FP (False Positive) means a false case correctly predicted and FN (False negative) for the false case that is incorrectly predicted. Base the comparison results, we have the following performance results: TP = 2/4 = 0.5, TN =1/4=0.25, FP=0, and FN=1/4=0.25. Then, we use two performance metrics to assess the accuracy of the Baysian classifier.

Prediction accuracy = (TP+TN) / (TP+TN+FP+FN) = 0.75

Prediction error = (FP+FN)/ (TP+TN+FP+FN) = 0.25 ■

Prediction Accuracy Analysis (9)

Table 7.6 Predicted Results of 4 Animals Compared with Their Actual Classes

Animal Name	Give Birth	Can Fly	Live in Water	Have Legs	Predicted Class	Actual Class	Prediction Status
Dog	yes	no	no	yes	M	M	TP
Monostream	no	no	no	yes	N	M	FN
Alligator	no	no	yes	yes	N	N	TN
Horse	yes	no	no	yes	M	M	TP

Dimensionality Reduction

- **Purpose:**
  - **Avoid curse of dimensionality**
  - **Reduce amount of time and memory required by data mining algorithms**
  - **Allow data to be more easily visualized**
  - **Help eliminate irrelevant features or reduce noise**
- **Techniques**
  - **Principle Component Analysis**
  - **Singular Value Decomposition**
  - **Others: supervised and non-linear techniques**

Support Vector Machines : (1)

Extending linear programming with multiple decision boundaries (hyperplanes) to separate the classes

Two key shortcomings of neural networks are the fact that the objective function is nonconvex (and hence may have multiple local minima) and the effort that is needed to tune the number of hidden neurons. Support vector machines (SVMs) deal with both of these issues.[21]

The origins of classification SVMs date back to the early dates of linear programming.[22] Consider the following linear program (LP) for classification:

$$\min e_1 + e_2 + \quad + e_{ng} + \quad + e_{nb}$$

subject to

$$w_1 x_{i1} + w_2 x_{i2} + \quad + w_n x_{in} \geq c - e_i, 1 \leq i \leq n_g$$

$$w_1 x_{i1} + w_2 x_{i2} + \quad + w_n x_{in} \leq c + e_i, n_g + 1 \leq i \leq n_g + n_b$$

$$e_i \geq 0$$

The LP assigns the good customers a score above the cut-off value c, and the bad customers a score below c. n_g and n_b represent the number of goods and bads, respectively. The error variables e_i are needed to be able to solve the program because perfect separation will typically not be possible. Linear programming has been very popular in the early days of credit scoring. One of its benefits is that it is easy to include domain or business knowledge by adding extra constraints to the model.

A key problem with linear programming is that it can estimate multiple optimal decision boundaries, as illustrated in Figure 3.19, for a perfectly linearly separable case.

Support Vector Machines (2)

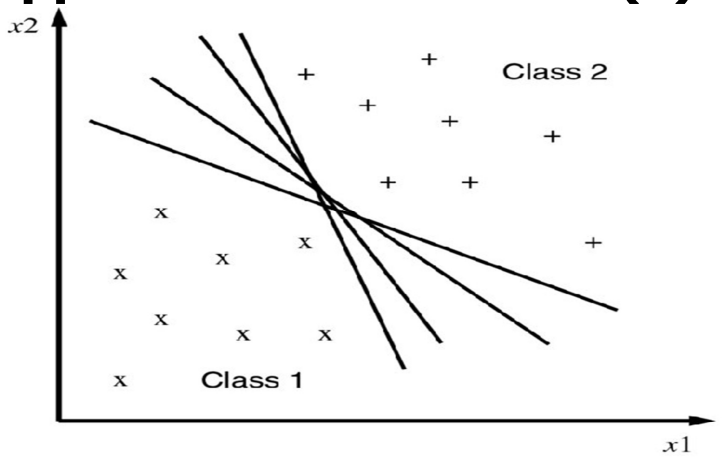
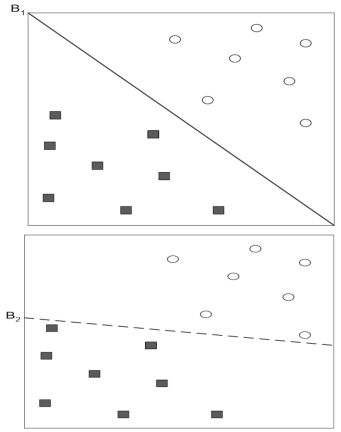


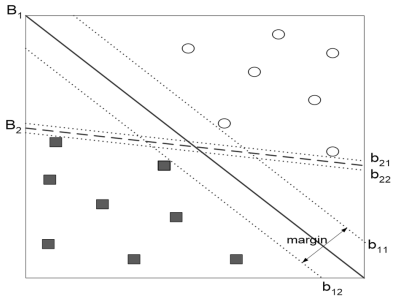
Figure 3.19 Multiple Separating Hyperplanes

SVMs add an extra objective to the analysis. Consider, for example, the situation depicted in Figure 3.20. It has two hyperplanes sitting at the edges of both classes and a hyperplane in between, which will serve as the classification boundary. The perpendicular distance from the first hyperplane H1 to the origin equals $| b-1|/|| w ||$, whereby $||w||$ represents the Euclidean norm of w calculated as $\| w \| = \sqrt{w_1^2 + w_2^2}$. Likewise, the perpendicular distance from H2 to the origin equals $| b + 1|/|| w ||$. Hence, the margin between both hyperplanes equals $2/|| w ||$. SVMs will now aim at maximizing this margin to pull both classes as far apart as

Support Vector Machines (3)

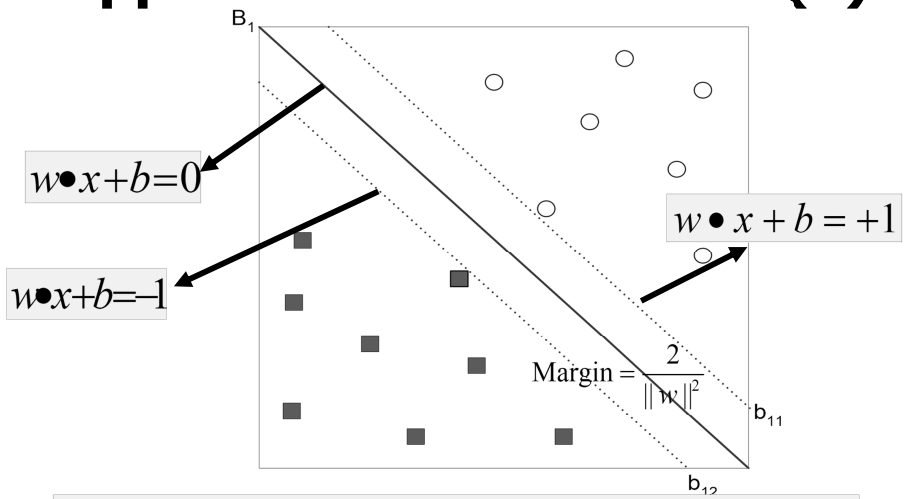
- Find a linear hyperplane (decision boundary) that separates the data





- Find a hyperplane to maximize the margin => B1 is better than B2

Support Vector Machines (4)



$$f(x) = \begin{cases} 1 & \text{if } w \bullet x + b \geq 1 \\ -1 & \text{if } w \bullet x + b \leq -1 \end{cases}$$

Support Vector Machines (5)

- We want to maximize: $\text{Margin} = \frac{2}{\| w \|^2}$

  Which is equivalent to minimizing:

$$L(w) = \frac{\| w \|^2}{2}$$

  – But subjected to the following constraints:

$$f(x_i) = \begin{cases} 1 & \text{if } w \bullet x_i + b \geq 1 \\ -1 & \text{if } w \bullet x_i + b \leq -1 \end{cases}$$

  - This is a constrained optimization problem
    - Numerical approach to solving the problem. (e.g., Quadratic programming)

Predictive Analytics Methods

- Linear Regression
- Logistic Regression
- Decision Trees
- Bayesian Classifier
- Artificial Neural Networks (ANN) (Lecture 19, Oct.31, 2016)
- Deep Neural networks (CNN and RNN) (Lecture 20, Nov. 2, 2016)
-