Costa Rica Big Data School: Introduction to Machine Learning I

Weijia Xu
Research Scientist, Group Manager
Data Mining & Statistics
Texas Advanced Computing Center
University of Texas at Austin
Dec. 2018

TACC

What is Machine Learning?

- Definition from T. Mitchell (1997). *Machine Learning book*:

> "A computer program is said to learn from experience $\mathcal{E}$ with respect to some class of tasks $\mathcal{T}$ and performance measure $\mathcal{P}$, if its performance at the tasks improves with the experiences."
>
> --- (Mitchell 1997)

  - **Learn from past experiences**
  - **Improves with experiences**

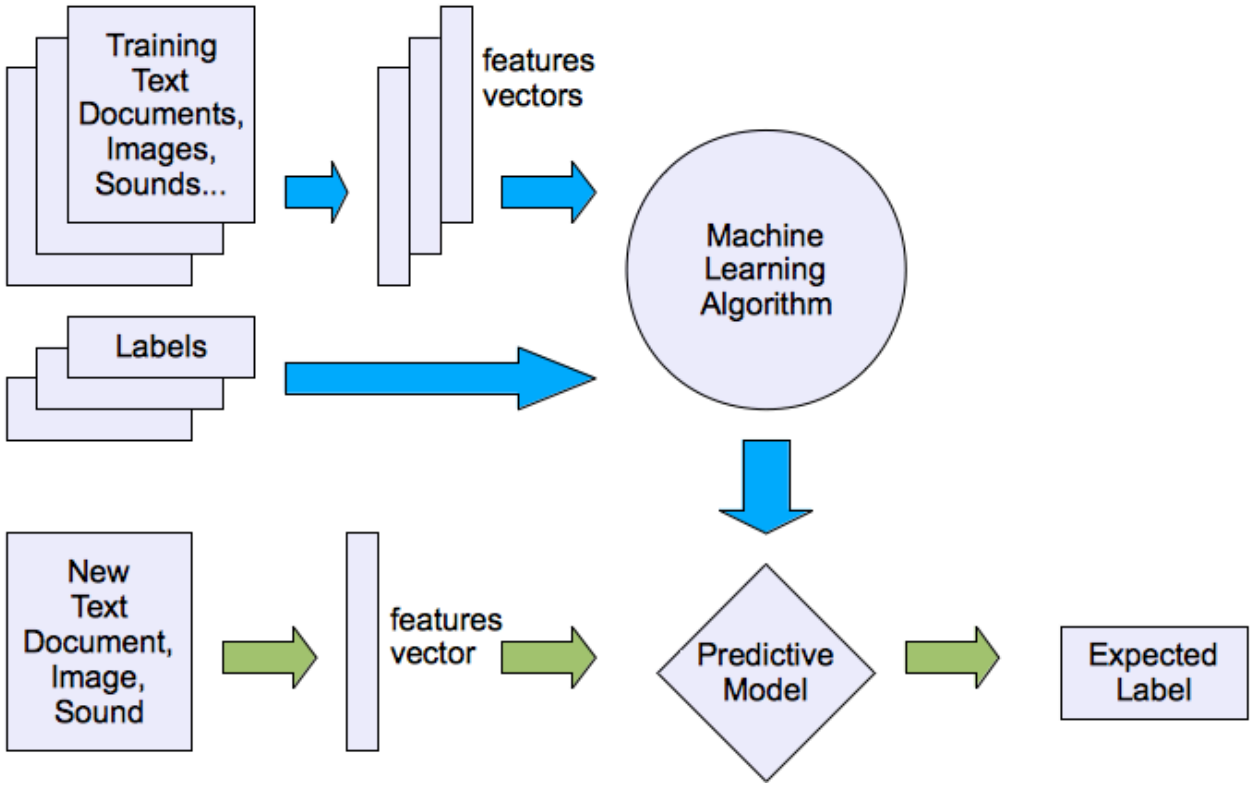
Goal of Machine Learning

- Learning general models from existing data
  - Data is cheap and abundant
  - Noises, hidden variables.
- Model is a good and useful approximation to the data.
- Model is “acquired” in the process different from existing theories/mathematical models.
  - Different from descriptive analysis, such as generating statistical summary.
  - Existing theories and mathematical models are basis and important to drive the machine learning.
  - Using available data to improve the model.

Training and Testing

- Training
  - The process of making the system to learn a model.
  - Data to be “observed” by the learning system
- Testing
  - Process to evaluate the performance of the model
  - Data are not observed by the learning system
- 80/20 split
  - 80% available data are randomly selected for training
  - The rest 20% are used for testing.
- Prediction
  - Apply model to data not in either training or testing data set.
  - Assume the input data and its prediction are from the same process producing the training data.

Supervised Learning



What are we learning?

Supervised Learning

- A model “explains” observed data in training data set and their labels/values.
  - Classification
    - Discrete labels
  - Regression
    - Real continuous values.
- Common learning techniques
  - Linear classifier.
  - Instance based functions (Non-parametric).
  - Probabilistic functions (Parametric),
  - Symbolic functions (Non-metric)
  - Aggregation/Ensemble methods.

Common Methods

- K Nearest Neighbor
- Regression Methods
- Support Vector machine
- Naïve Bayes Classifier
- Decision Tree
- Random Forest

K Nearest Neighbors

Nearest Neighbor Classification

- The basic idea:

  If two data objects are similar (close), they are likely are from the same class.

- General workflow

  Start with a set of data object with class labels

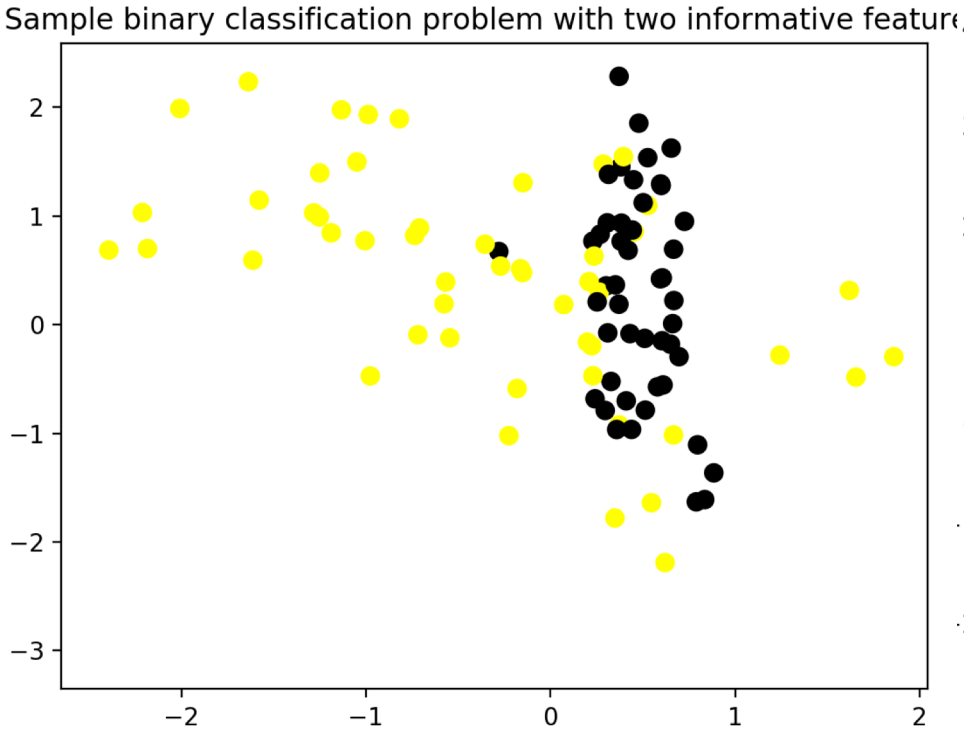
  Given a distance (similarity)measure of comparing data

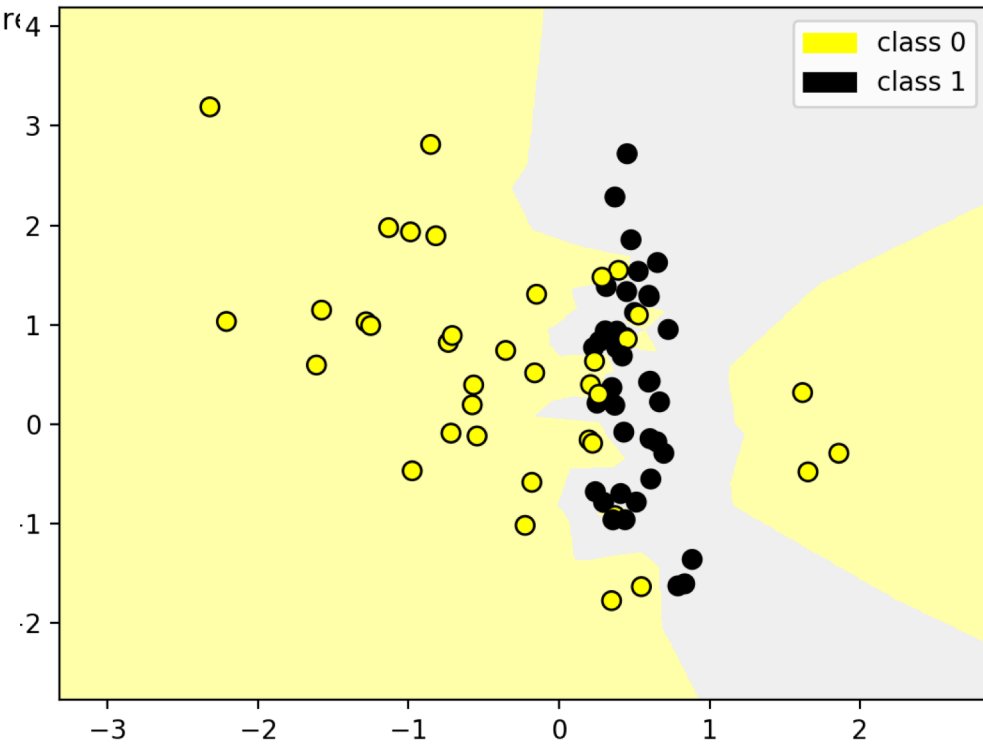
  Retrieve (k) nearest neighbor for the unknown data

  Assign class label based on the retrieved record.

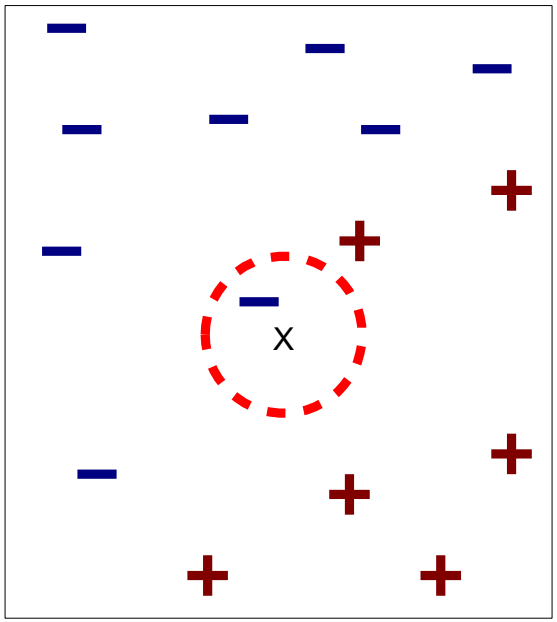
Nearest Neighbor Classification

- A simple binary classification case

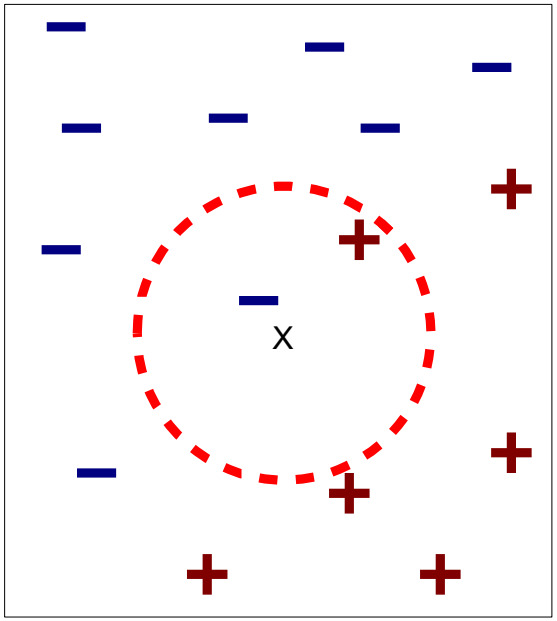




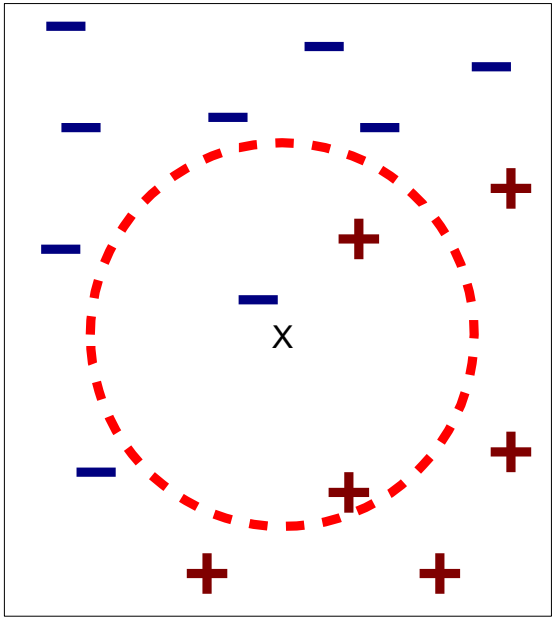
K-NN Classification



(a) 1-nearest neighbor



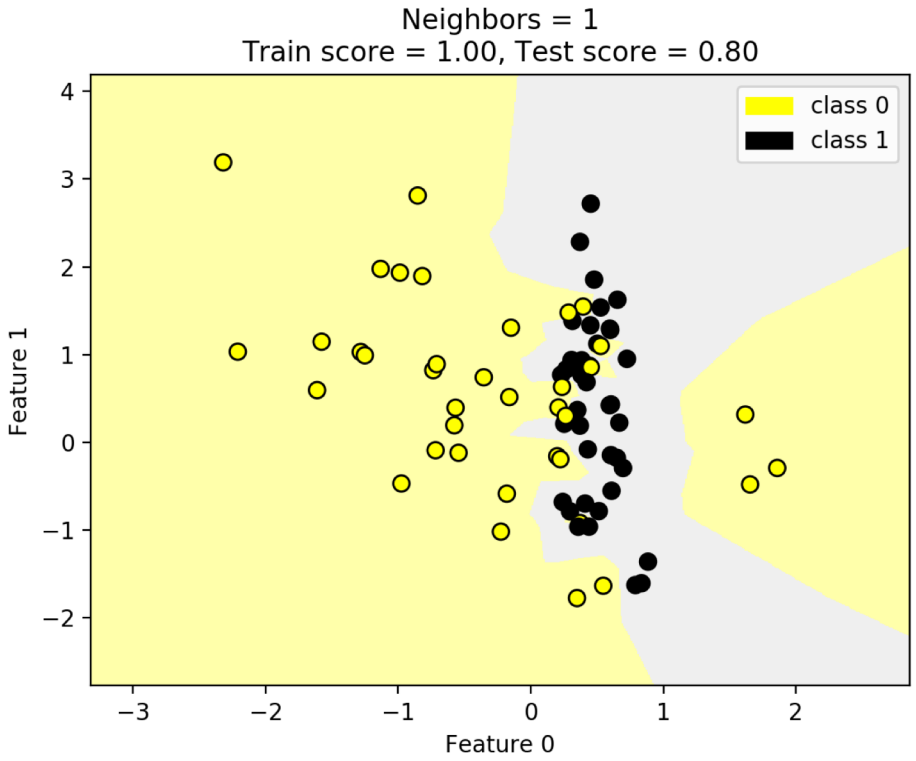
(b) 2-nearest neighbor

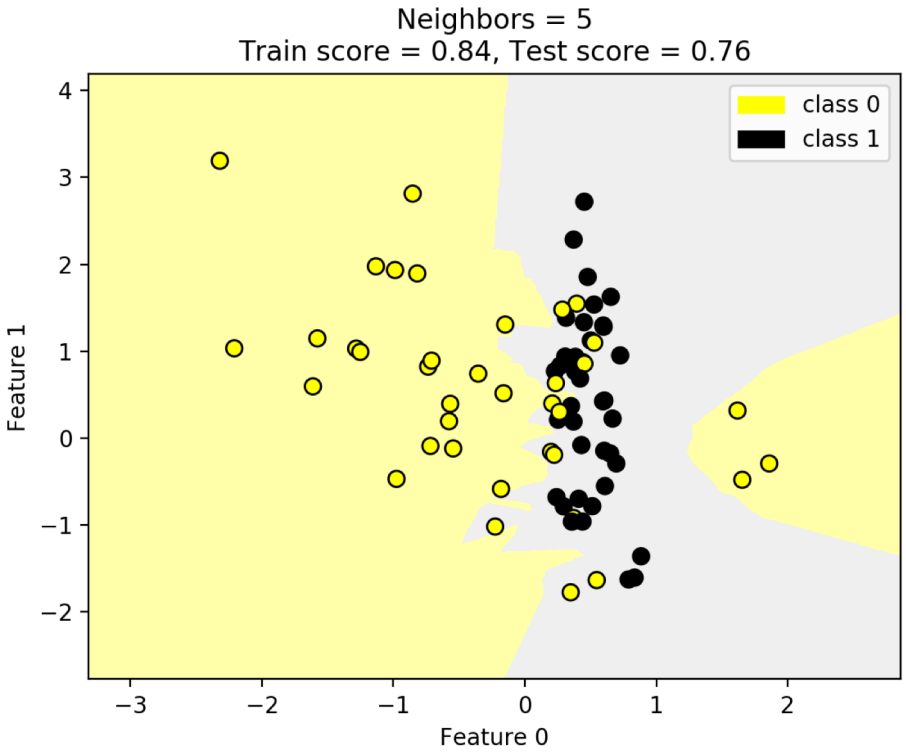


(c) 3-nearest neighbor

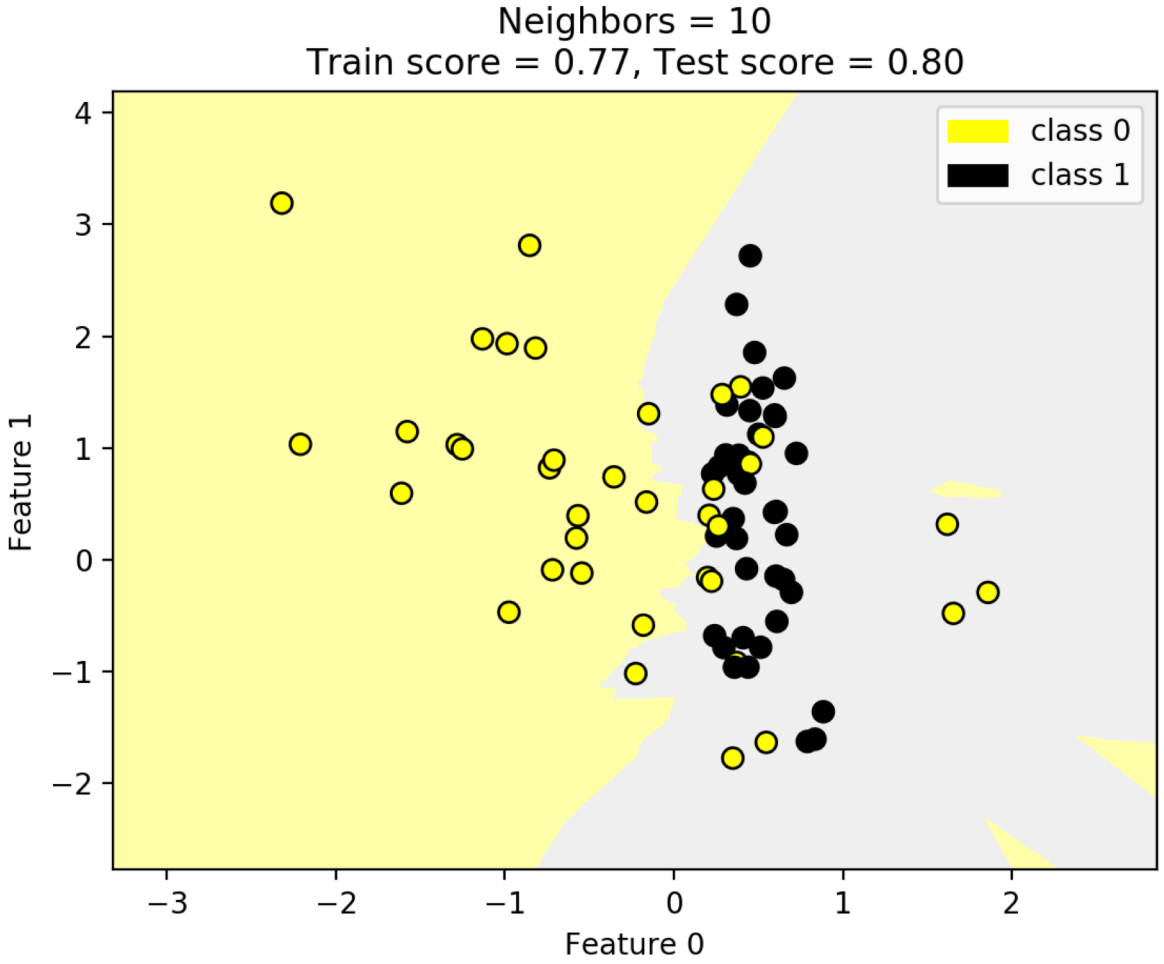
Choice of K

- What changes?





Choice of K


Neighbors = 10
Train score = 0.77, Test score = 0.80
class 0
class 1
Feature 1
Feature 0
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2

Concerns and Limitations

- Distance/similarity measures might be misleading

1 1 1 1 1 1 1 1 1 1 1 0		1 0 0 0 0 0 0 0 0 0 0 0	
0 1 1 1 1 1 1 1 1 1 1 1	vs	0 0 0 0 0 0 0 0 0 0 0 1	
d = 1.4142		d = 1.4142	

- Less effective for high dimensional data

- Values of distance measures might be dominated by one dimension

Regression

Regression

- A statistical measure to determine (the strength) of the relationship between one dependent variable and one or a series of changing variables
- Help us understand the relationship between variables
- Prediction values for unknown target

A Typical Regression Model

- The independent variables X
- The dependent variables Y
  outcome, target, or criterion variable
- The unknown parameters $\boldsymbol{\theta}$
- Predict/estimate Y with F(X, $\boldsymbol{\theta}$)

Common Goal of the Regression

- Learn the parameters to minimize the cost/prediction errors

- Ordinary least squares (OLS): to minimize

$$\sum (ax^{(i)} - y^{(i)})^2$$

- Least absolute deviations: to minimize

$$\sum |ax^{(i)} - y^{(i)}|$$

Common Metrics of Performance

- Coefficient of determination, aka, R^2
- The proportion of the variance in the dependent variable that is predictable from the independent variable(s)

$$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}},$$

- Total sum of square

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

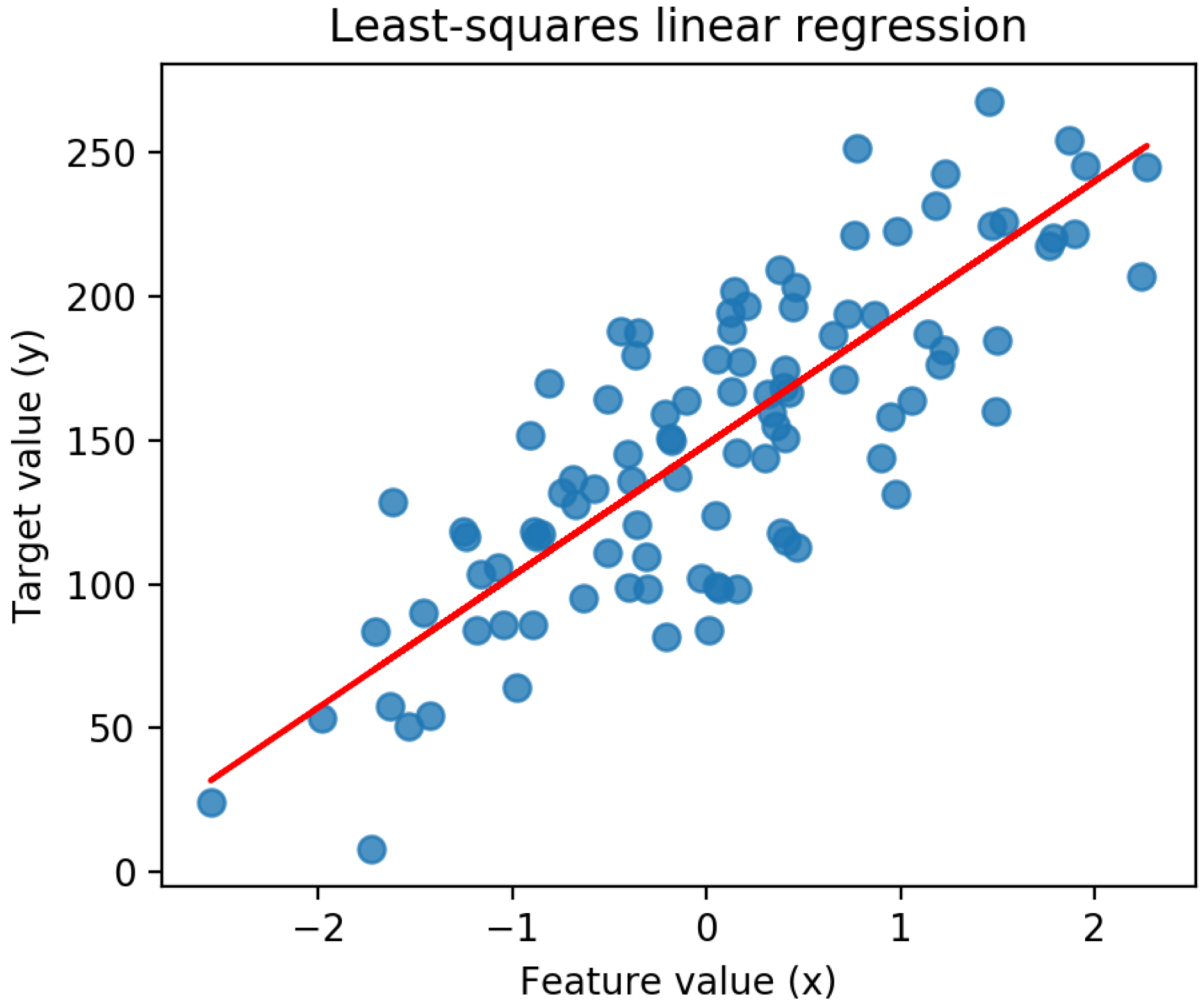
- Residue sum of squares

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2$$

Simple Linear Regression

- Only one independent variable (y) and one dependent variable (x)
- The outcome variable is related to a single predictor
- Assuming label and feature are connected through a function e.g. $y = a x + b$

Least Square Solution



Regularization

- A parameter to tune the “strength” of learning
- Why? To avoid over fitting
- Constrains machine learning algorithm to improve out-of-sample error and noise.
- An example is Ridge regression

Ridge Regression

- OLS tries to minimize

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

- Ridge regression adds a regularization term

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2$$

- λ is a positive parameter, when λ=0 ridge regression is effectively the same as OLS.

Multiple Linear Regression

- More than one explanatory variable
- x is a vector of features, $x \in R^N$
- $\boldsymbol{\theta}$ is a vector of weights, $\boldsymbol{\theta} \in R^N$

As for the ordinary least squares:

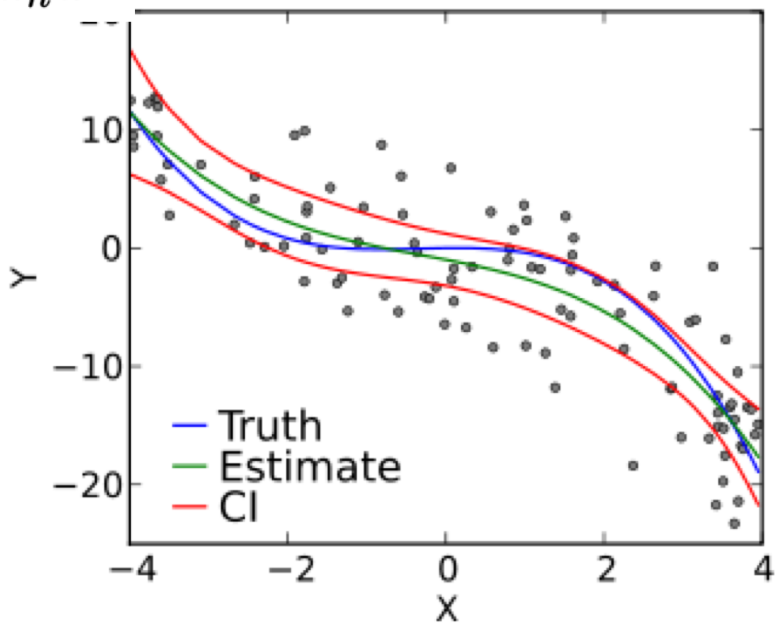
$$\theta = (A^T A)^{-1} A^T b$$

Polynomial regression

- a special case of **multiple linear regression**

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^m \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$

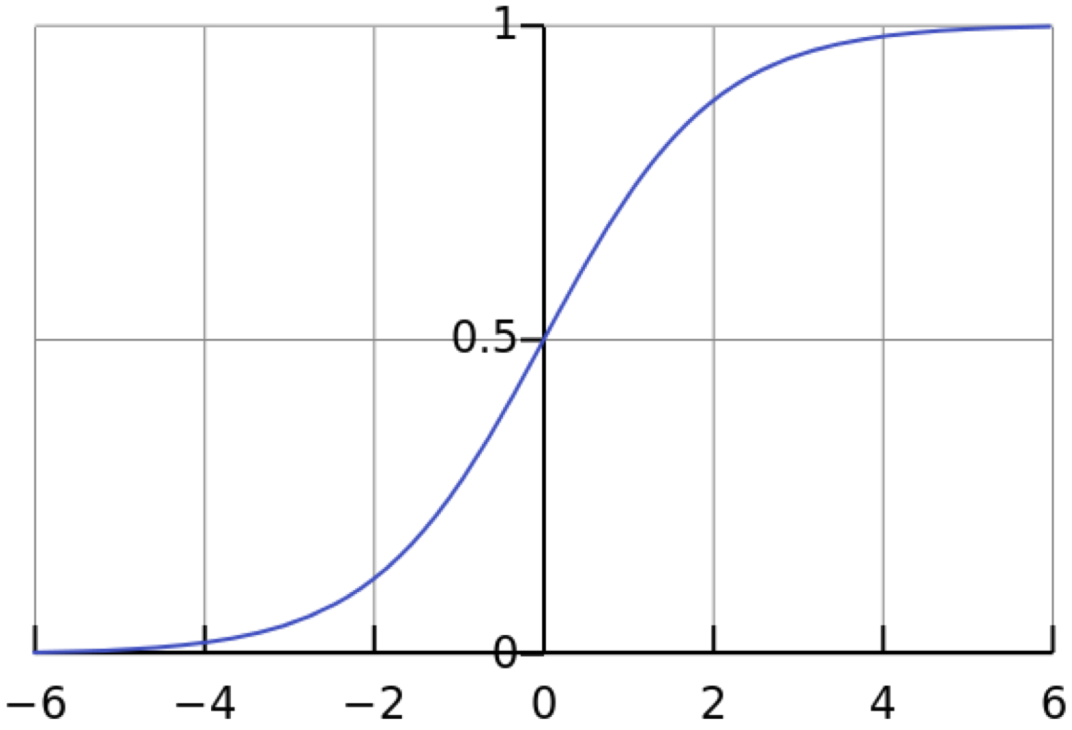


(Binary) Logistic Regression

- (Binary) dependent variable
- ”y” can only take two values (usually 0 and 1)
  Positive/Negative:
  Pass/fail
  Heathy/sick
- One or more independent variables x
- Predict the probability of a binary response

The Standard Logistic Function

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$



Logistic Regression

If z is a linear function of a single independent variable x

$$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

$$F(x) = \sigma(z) = \frac{1}{1 + e^{-\theta_0 - \theta_1 x_1 - \theta_2 x_2}}$$

Support Vector Machine

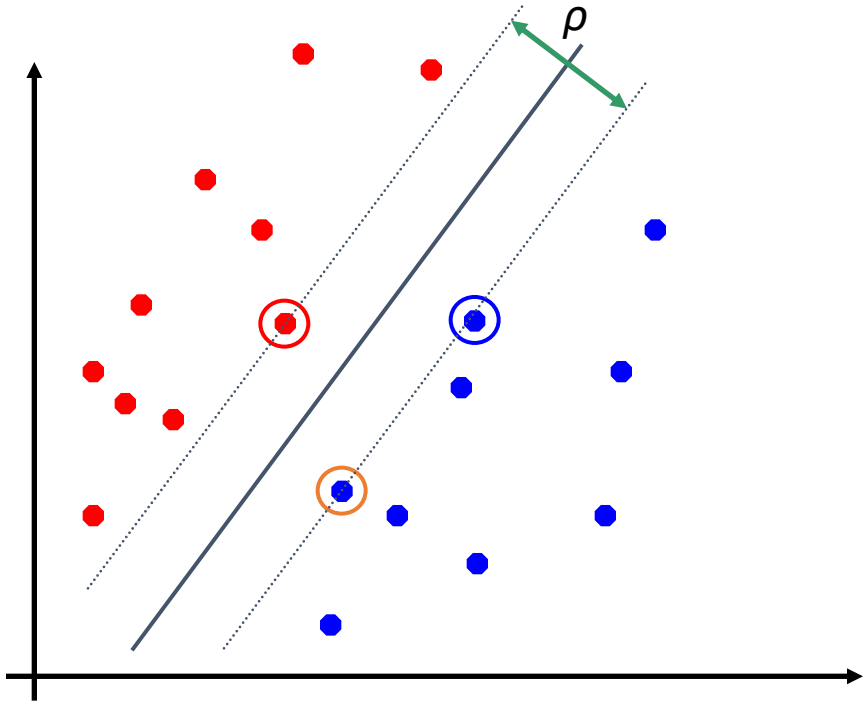
- A discriminative classifier
- Given a few sets of labeled training data
  supervised learning
- Generate an optimal hyperplane
-
- Categorizes new examples.

Binary Classification with Linear Separator

Red and blue dots are representations of objects from two classes in the training data

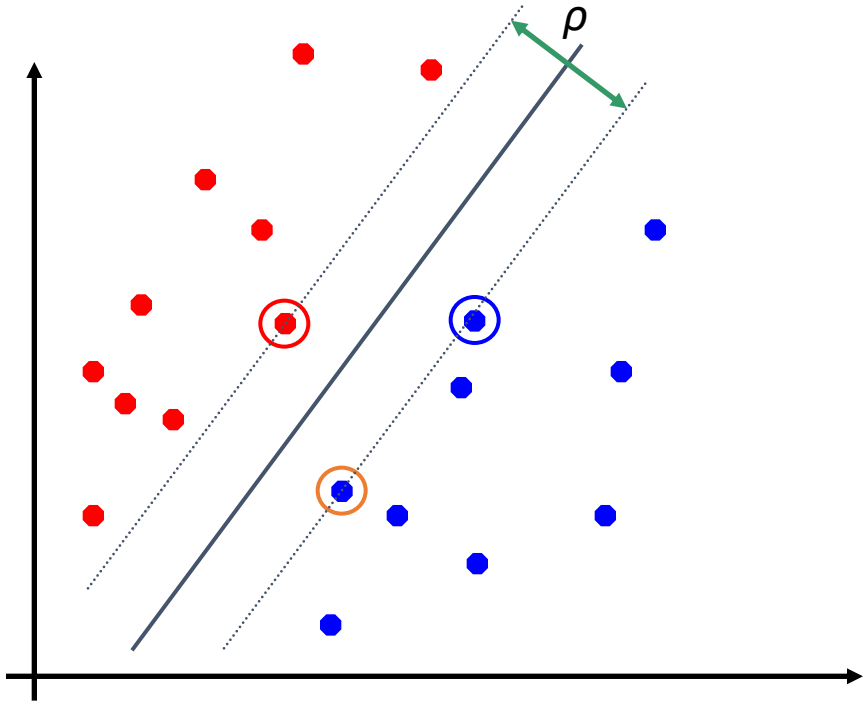
The line is a linear separator for the two classes

The closets objects to the hyperplane is the support vectors



Binary Classification with Linear Separator

- Find a line passing as far as possible fro both points
- The optimal separating hyperplane *maximizes* the margin of the training data.
- A line is bad if it passes too close to some points (noise sensitive)



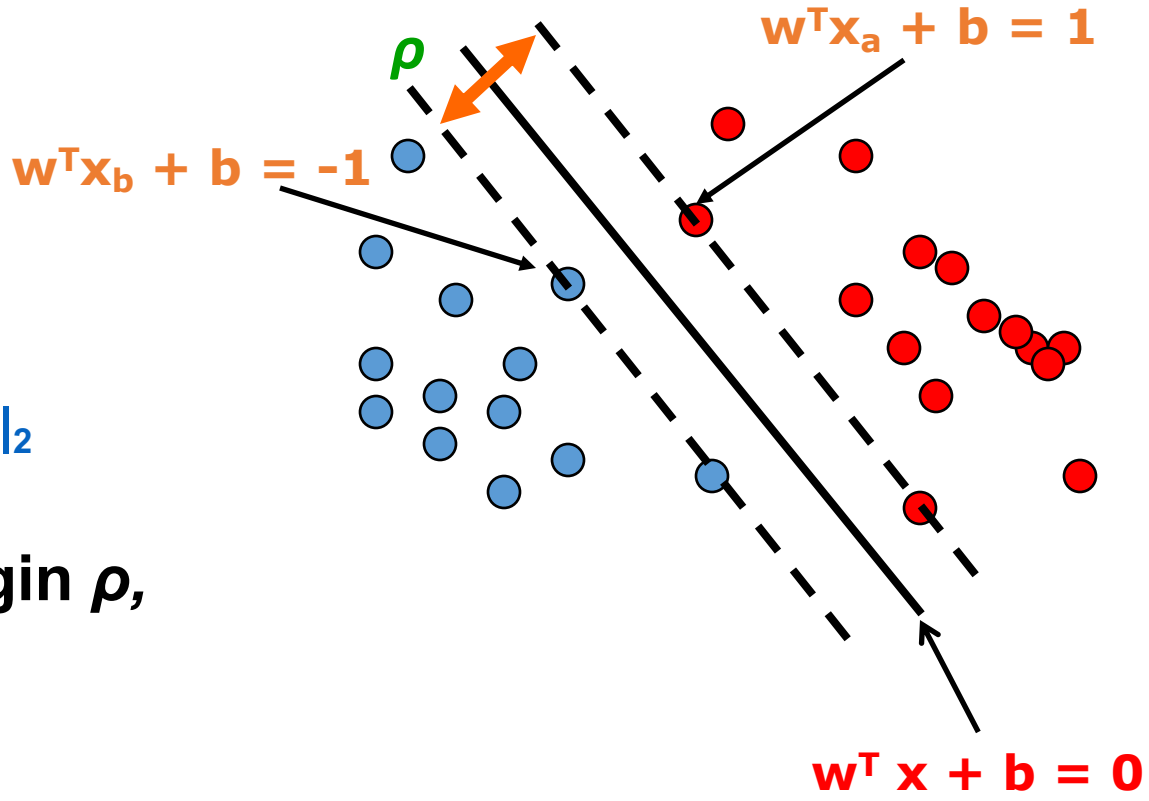
Linear Support Vector Machine

Hyperplane

$\mathbf{w}^T \mathbf{x} + b = 0$

$w^T(x_a - x_b) = 2$

$\rho = ||x_a - x_b||_2 = 2/||w||_2$



Maximize the margin *ρ*,

Minimize **||w||**

Classification with SVMs

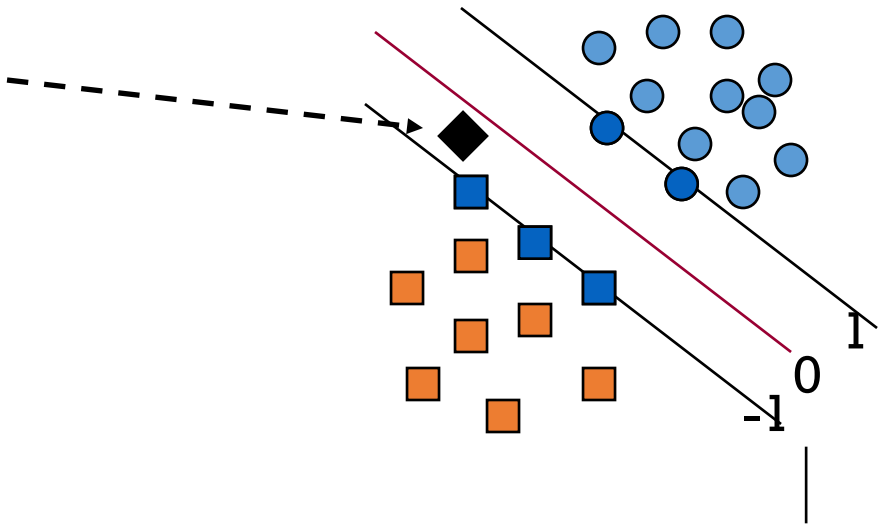
- Given a new point **x**, we can score its projection onto the hyperplane normal:
  i.e., compute score: $\mathbf{w}^T\mathbf{x} + b$
  Decide class based on whether < or > 0

- Can set confidence threshold *t*.

Score > *t*: yes

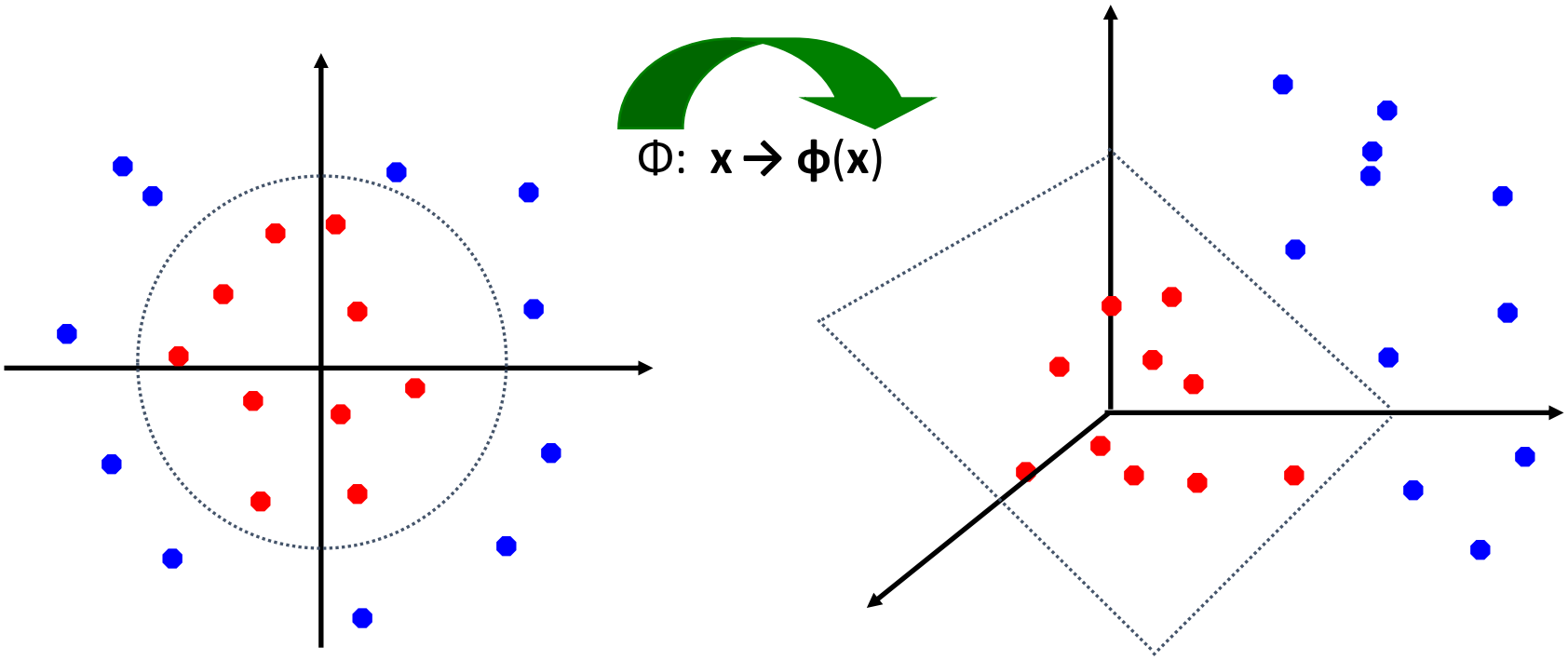
Score < -*t*: no

Else: don't know



Non-linear SVM

Kernels functions: Mapping the original feature space to some higher-dimensional feature space where the training set is separable



Naïve Bayes Classifier

Naïve Bayes Classifier

- The Basic idea
  - Treat the data and its label as some statistic process
  - From historical data
    - → estimate the parameters
  - From test data
    - → compute the probability to be in certain class.

- General Statistic Inference

From joint probability

$$P(AB|I) = P(A|BI)P(B|I)$$

$$P(BA|I) = P(B|AI)P(A|I)$$

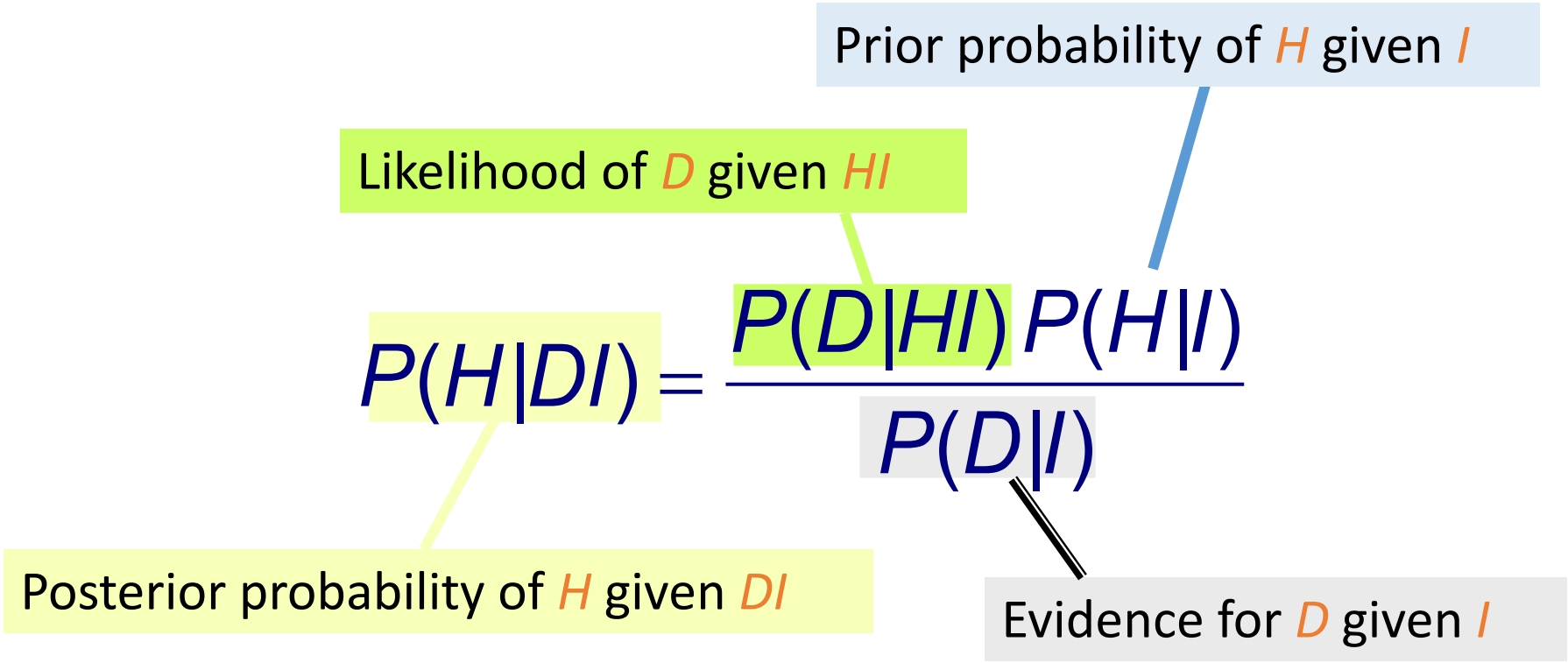
and $AB = BA$ we get Bayes Theorem

$$P(B|AI) = \frac{P(A|BI)\,P(B|I)}{P(A|I)}$$

- Bayes Theorem derives from the axioms of probability calculus and therefore exists in both Frequentist and Bayesian statistics
- In Bayesian statistics it plays a central role

Bayes Theorem for Inference

Let's use H (Hypothesis) and D (Data) instead of A, B



Bayes Theorem: update hypothesis based on the Data

Bayesian Classifiers

Consider each attribute and class label as random variables

Given a record with attributes (A_1, A_2,…,A_n)

- Goal is to predict class 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?

Bayesian Classifiers

Approach:

compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes theorem

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

Choose value of C that maximizes

$P(C \mid A_1, A_2, \ldots, A_n)$

Equivalent to choosing value of C that maximizes

$P(A_1, A_2, \ldots, A_n \mid C)\, P(C)$

How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?

Naïve Bayes Classifier

Assume independence among attributes A_i when class is given:

$$P(A_1, A_2, \ldots, A_n \mid C) = P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$$

Can estimate $P(A_i \mid C_j)$ for all A_i and C_j.

New point is classified to C_j if $P(C_j) \prod P(A_i \mid C_j)$ is maximal.

How to Estimate Probabilities from Data?

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 probability:

$P(C) = N_c/N$

e.g., P(No) = 7/10,

P(Yes) = 3/10

Attribute probability

For discrete attributes:

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

k

where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C_k

Examples:

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

P(Refund=Yes|Yes)=0

Estimation using Probability Density function

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

Normal distribution:

$$P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

One for each (A_i, c_i) pair

For (Income, Class=No):

If Class=No

sample mean = 110

sample variance = 2975

$$P(Income = 120 \mid No) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Example of Naïve Bayes Classifier

Given a Test Record:

$$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$$

naive Bayes Classifier:

P(Refund=Yes|No) = 3/7
P(Refund=No|No) = 4/7
P(Refund=Yes|Yes) = 0
P(Refund=No|Yes) = 1
P(Marital Status=Single|No) = 2/7
P(Marital Status=Divorced|No)=1/7
P(Marital Status=Married|No) = 4/7
P(Marital Status=Single|Yes) = 2/7
P(Marital Status=Divorced|Yes)=1/7
P(Marital Status=Married|Yes) = 0

For taxable income:
If class=No: sample mean=110
sample variance=2975
If class=Yes: sample mean=90
sample variance=25

- P(X|Class=No) = P(Refund=No|Class=No)
  × P(Married| Class=No)
  × P(Income=120K| Class=No)
  = 4/7 × 4/7 × 0.0072 = 0.0024

- P(X|Class=Yes) = P(Refund=No| Class=Yes)
  × P(Married| Class=Yes)
  × P(Income=120K| Class=Yes)
  = 1 × 0 × 1.2 × 10^{-9} = 0

Since P(X|No)P(No) > P(X|Yes)P(Yes)

Therefore P(No|X) > P(Yes|X)

=> Class = No

Naïve Bayes (Summary)

Robust to isolated noise points

Handle missing values by ignoring the instance during probability estimate calculations

Robust to irrelevant attributes

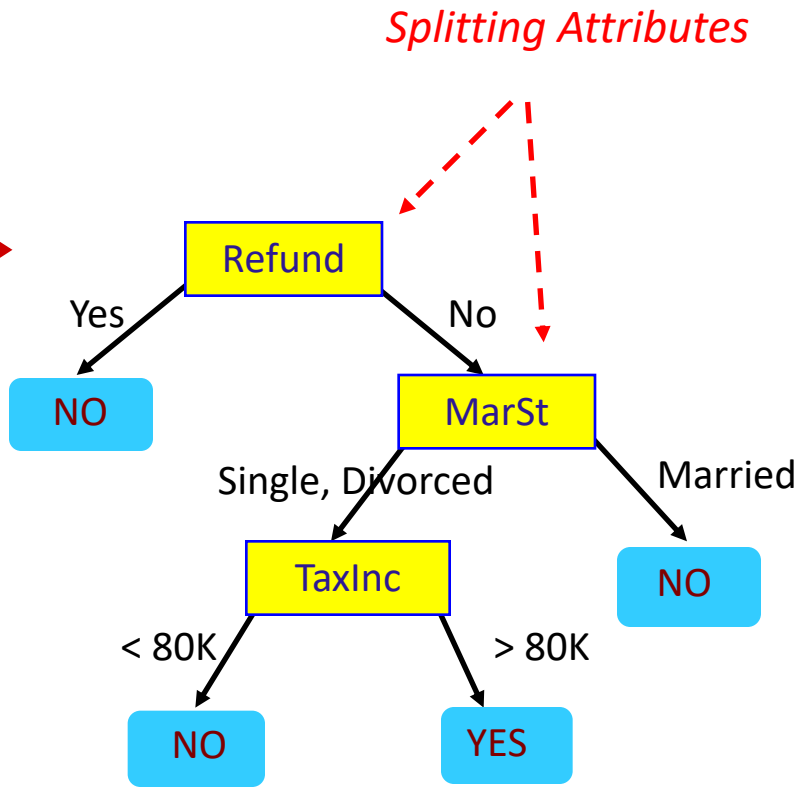
Independence assumption may not hold for some attributes

Decision Tree Classifier

Decision Tree

Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
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

Training Data

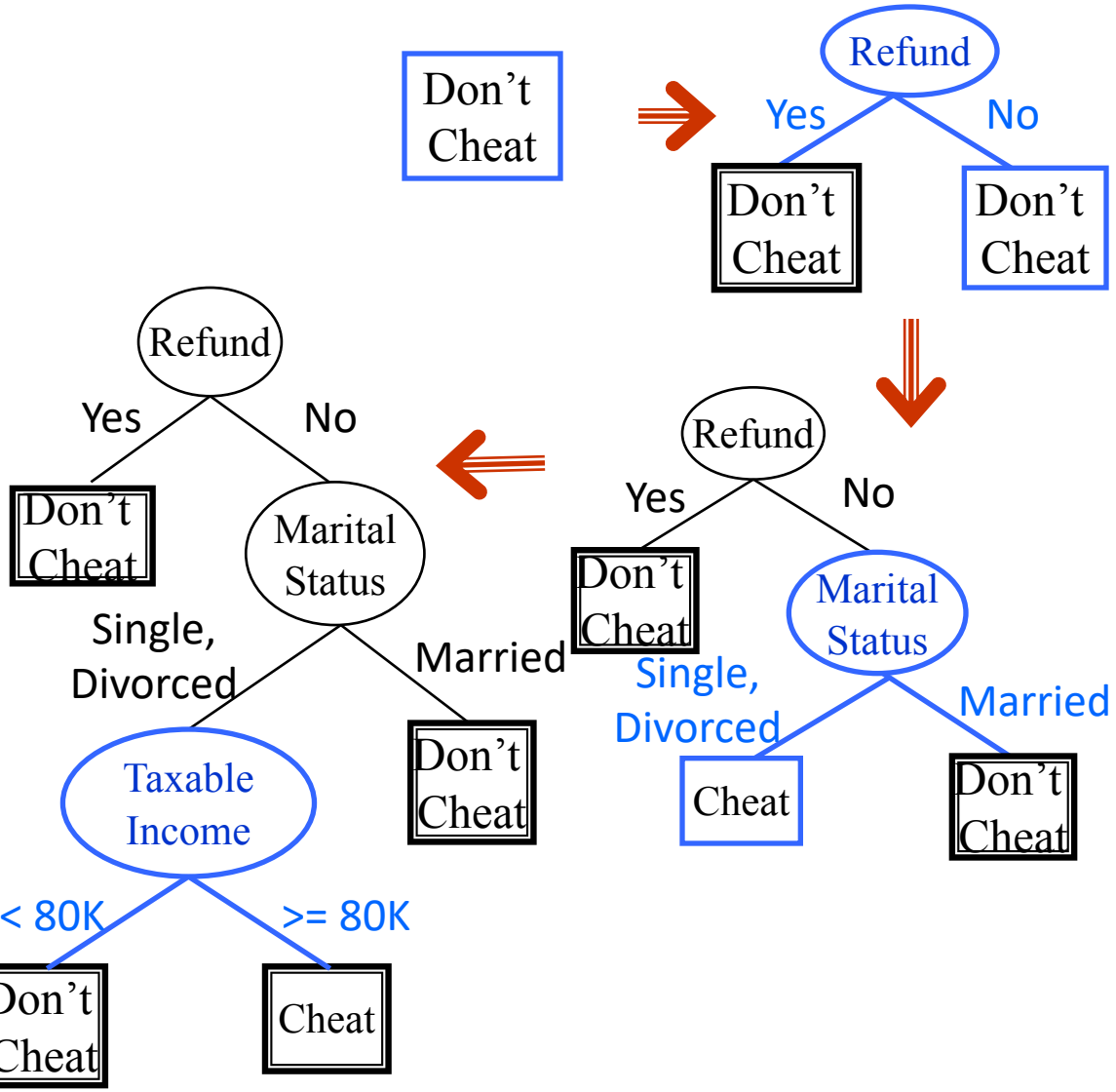


Model: Decision Tree

Hunt's Algorithm

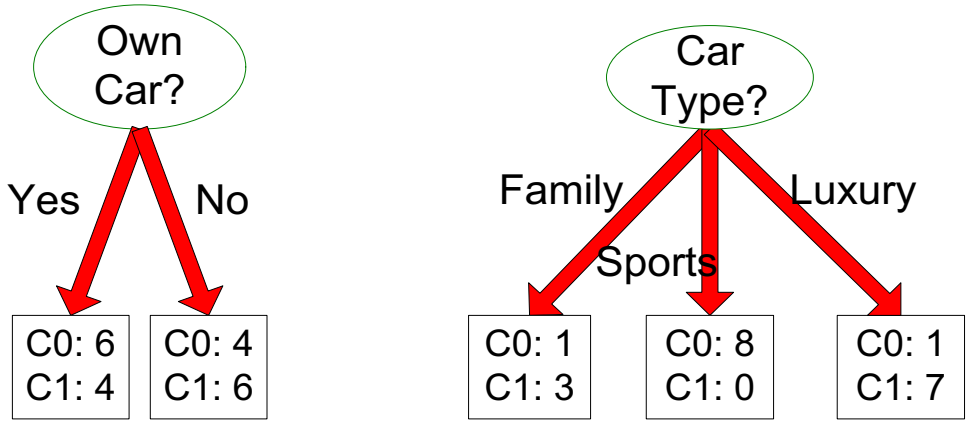
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Class 1: Don't cheat (default class)
Class 2: Cheat



Which split is the best?

Before Splitting: 10 records of class 0,
10 records of class 1



Non-homogeneous,

High degree of impurity

Homogeneous,

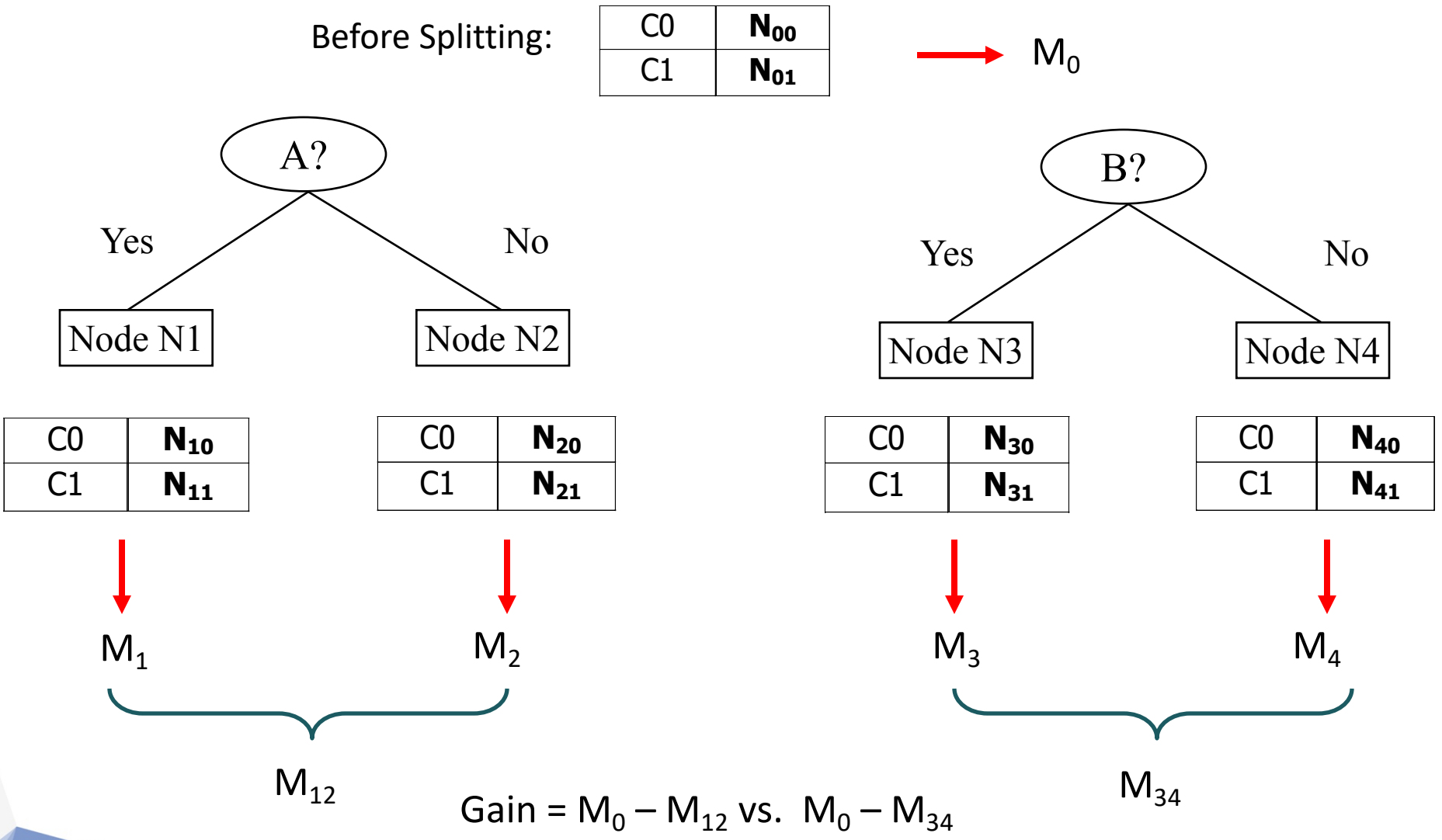
Low degree of impurity

Homogeneous,

But too many branches…

- Nodes with homogeneous class distribution are preferred
- Impurity of Nodes

How to Find the Best Split



Measuring Impurity with Entropy

Entropy at a given node t:

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- Maximum ($\log n_c$) when records are equally distributed among all classes implying least information
- Minimum (0.0) when all records belong to one class, implying most information

Examples for computing Entropy

$$Entropy(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Information Gain from Splitting Data

Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

n_i is number of records in partition i

- Measures reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
- Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

Splitting Based on Entropy

Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions

n_i is the number of records in partition i

Adjusts Information Gain by the entropy of the partitioning (SplitINFO).

Higher entropy partitioning (large number of small partitions) is penalized!

Other Measures of Node Impurity

Gini Index

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

Maximum (1 - 1/n_c) when records are equally distributed among all classes, implying least interesting information

Minimum (0.0) when all records belong to one class, implying most interesting information

Misclassification error

$$Error(t) = 1 - \max_i P(i \mid t)$$

Maximum (1 - 1/n_c) when records are equally distributed among all classes, implying least interesting information

Minimum (0.0) when all records belong to one class, implying most interesting information

Decision Tree Based Classification

- Advantages:
  - Inexpensive to construct
  - Extremely fast at classifying unknown records
  - Easy to interpret for small-sized trees
  - Accuracy is comparable to other classification techniques for many simple data sets

- Less effective when
  - High background noise.
  - Large scale data
  - Data with high dimension

Random Forest

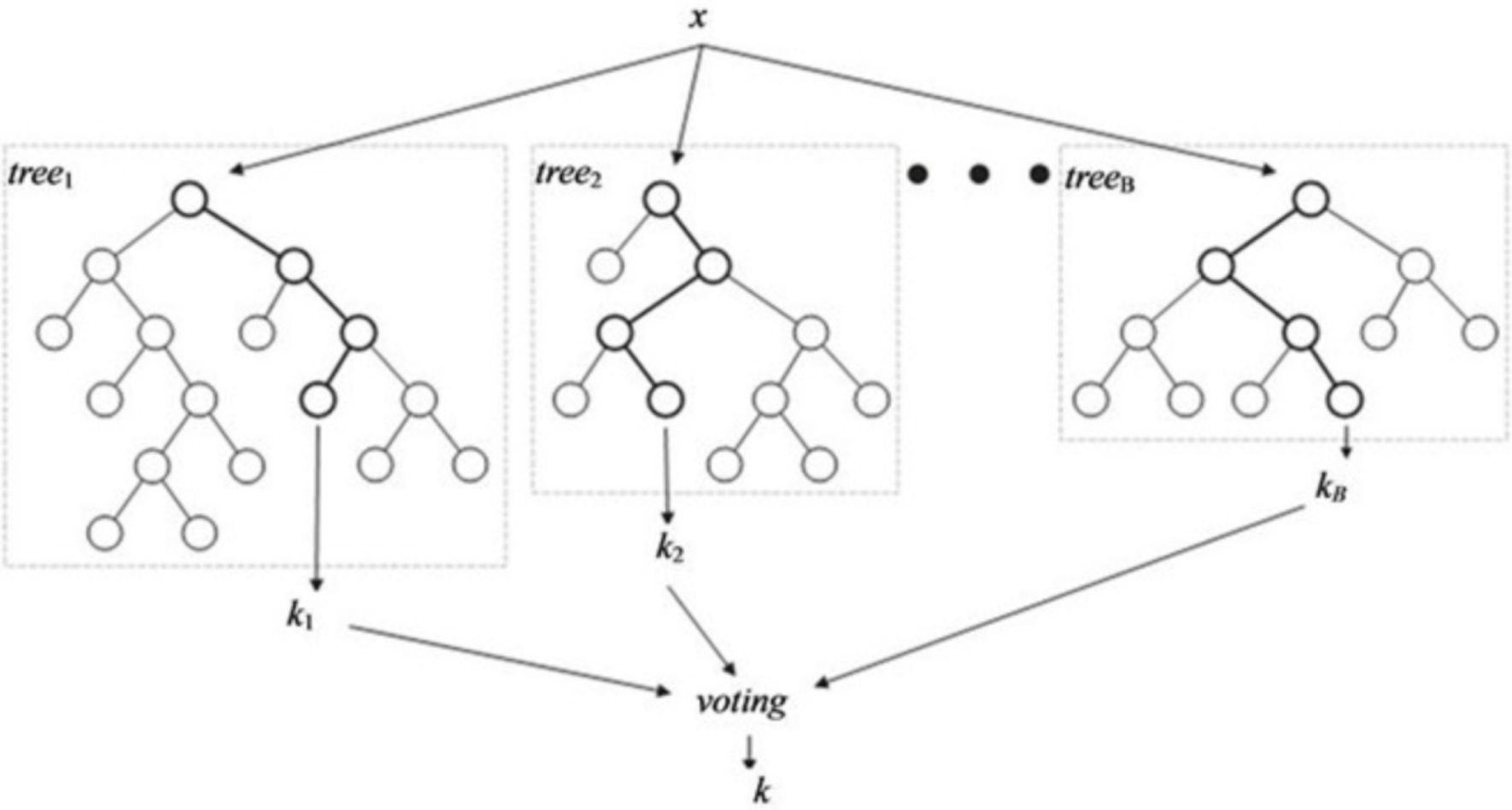
A collection of decision trees

Building stage

- Only a subset of attributes are chosen to make decision at each node of the tree
- Select best split from selected subset of attributes.

Classifying

- All trees are used to make a decision.
- The final class label is determined statistically e.g. majority rule.


x
tree1
tree2
treeB
k1
k2
kB
voting
k

Random Forest

Advantage

- Accurate
- Good for high dimensional data
- Reduce the need of feature detection and selection.
- Give estimate on important features.

Disadvantage

- More computational requirement
- Less effective with noisy training data.
- The result is an statistical ensemble and might be hard to interpret.