Classification: Alternative Techniques

Salvatore Orlando

Rule-Based Classifier

- **Classify records by using a collection of "if…then…" rules**

- **Rule:** **(*Condition*) → *y***
  - **where**
    - **Condition is a conjunctions of attributes**
    - ***y* is the class label**
  - ***LHS*: rule antecedent or condition**
  - ***RHS*: rule consequent**
  - **Examples of classification rules:**
    - **(Blood Type=Warm) ∧ (Lay Eggs=Yes) → Birds**
    - **(Taxable Income < 50K) ∧ (Refund=Yes) → Evade=No**

Rule-based Classifier (Example)

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds

R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

Application of Rule-Based Classifier

- **A rule *r* trigger an instance *x* if the attributes of the instance satisfy the condition of the rule**

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds

R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers the hawk => Bird

The rule R3 covers the grizzly bear => Mammal

Rule Coverage and Accuracy

- **Coverage of a rule:**
  - **Fraction of records in the training set that satisfy the antecedent of a rule**

- **Accuracy of a rule:**
  - **Fraction of records in the training set that satisfy both the antecedent and consequent of a rule**

Tid	Refund	Marital Status	Taxable Income	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

(Status=Single) → No

Coverage = 40%, Accuracy = 50%

How does Rule-based Classifier Work?

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds

R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule R3, so it is classified as a mammal

A turtle triggers both R4 and R5 (not **mutually exclusive)**

A dogfish shark triggers none of the rules (not **exhaustive**)

Characteristics of Rule-Based Classifier

- **Mutually exclusive rules**
  - **Classifier contains mutually exclusive rules if the rules are independent of each other**
  - **Every record is covered by at most one rule**

- **Exhaustive rules**
  - **Classifier has exhaustive coverage if it accounts for every possible combination of attribute values**
  - **Each record is covered by at least one rule**

From Decision Trees To Rules
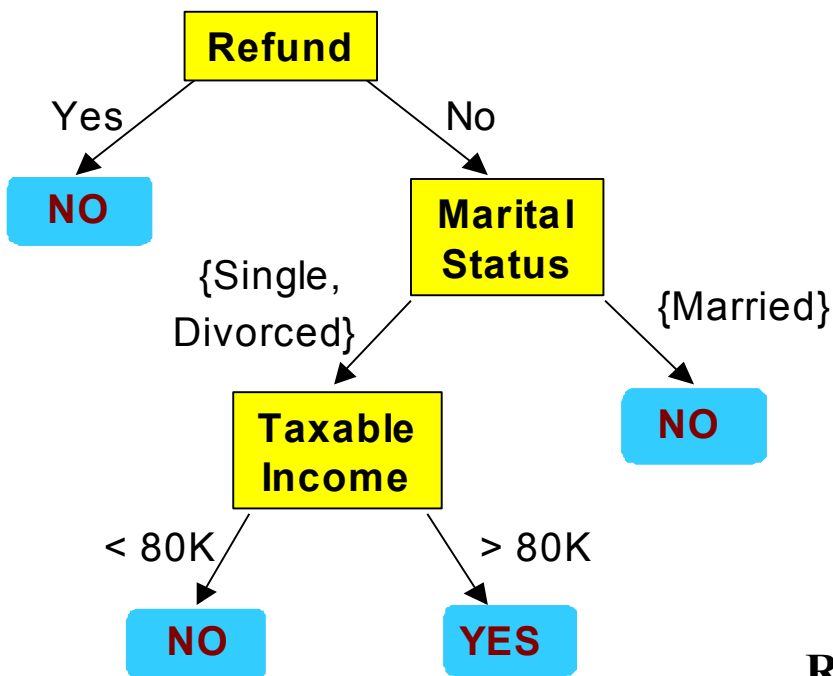
Classification Rules

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced}, Taxable Income<80K) ==> No
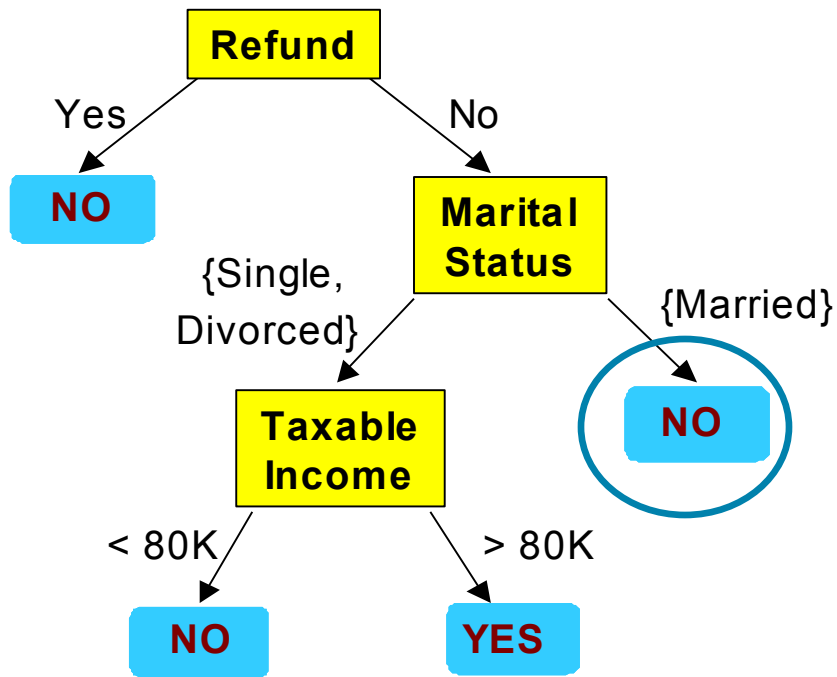
(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

Rules are mutually exclusive and exhaustive

Rule set contains as much information as the tree

Rules Can Be Simplified

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

Initial Rule: (Refund=No) ∧ (Status=Married) → No

Simplified Rule: (Status=Married) → No

Effect of Rule Simplification

- **Rules are no longer mutually exclusive**
  - **A record may trigger more than one rule**
  - **Solution?**
    - **Ordered rule set**
    - **Unordered rule set – use voting schemes**

- **Rules are no longer exhaustive**
  - **A record may not trigger any rules**
  - **Solution?**
    - **Use a default class**

Ordered Rule Set

- **Rules are rank ordered according to their priority**
  - **An ordered rule set is known as a decision list**
- **When a test record is presented to the classifier**
  - **It is assigned to the class label of the highest ranked rule it has triggered**
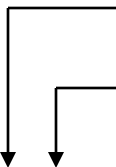 - **If none of the rules fired, it is assigned to the default class**

R1: (Give Birth = no) ∧ (Can Fly = yes) → Birds

R2: (Give Birth = no) ∧ (Live in Water = yes) → Fishes

R3: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

R4: (Give Birth = no) ∧ (Can Fly = no) → Reptiles

R5: (Live in Water = sometimes) → Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

Building Classification Rules

- **Direct Method:**
    - **Extract rules directly from data**
    - **e.g.: RIPPER, CN2, Holte's 1R**

- **Indirect Method:**
    - **Extract rules from other classification models (e.g. decision trees, neural networks, etc).**
    - **e.g: C4.5 rules**

Advantages of Rule-Based Classifiers

- **As highly expressive as decision trees**
- **Easy to interpret**
- **Easy to generate**
- **Can classify new instances rapidly**
- **Performance comparable to decision trees**
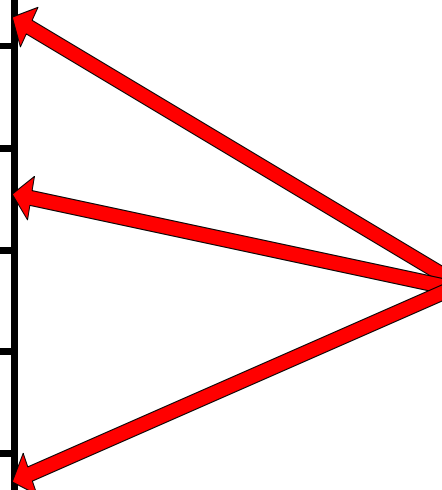
Instance-Based Classifiers

Set of Stored Cases

Atr1	………..	AtrN	Class
			A
			B
			B
			C
			A
			C
			B

- **Store the training records**

- **Use training records to predict the class label of unseen cases**

Unseen Case

| Atr1	………..	AtrN

Instance Based Classifiers

- **Examples:**
  - **Rote-learner**
    - **Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly**

  - **Nearest neighbor**
    - **Uses k "closest" points (nearest neighbors) for performing classification**

Nearest Neighbor Classifiers

- **Basic idea:**
  - **If it walks like a duck, quacks like a duck, then it's probably a duck**


Compute Distance

Test Record

Training Records

Choose k of the "nearest" records
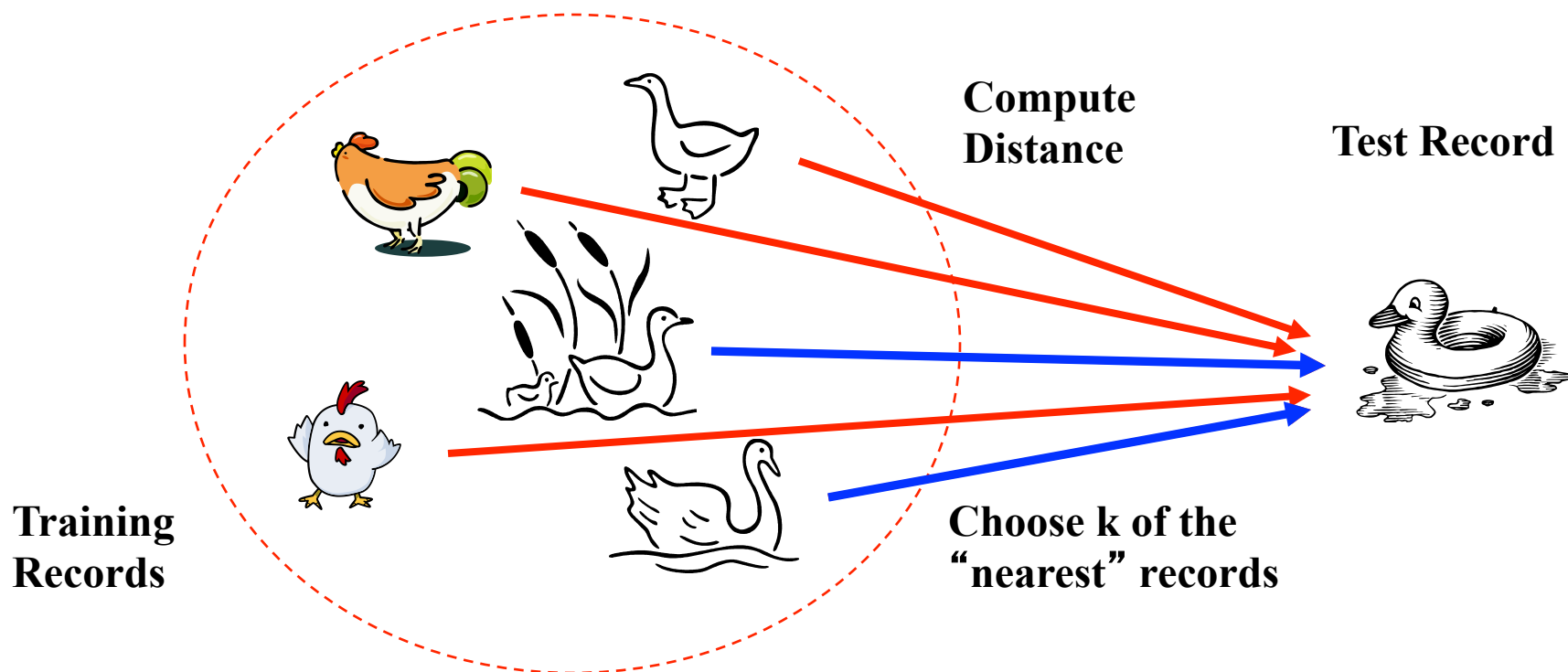
Nearest-Neighbor Classifiers
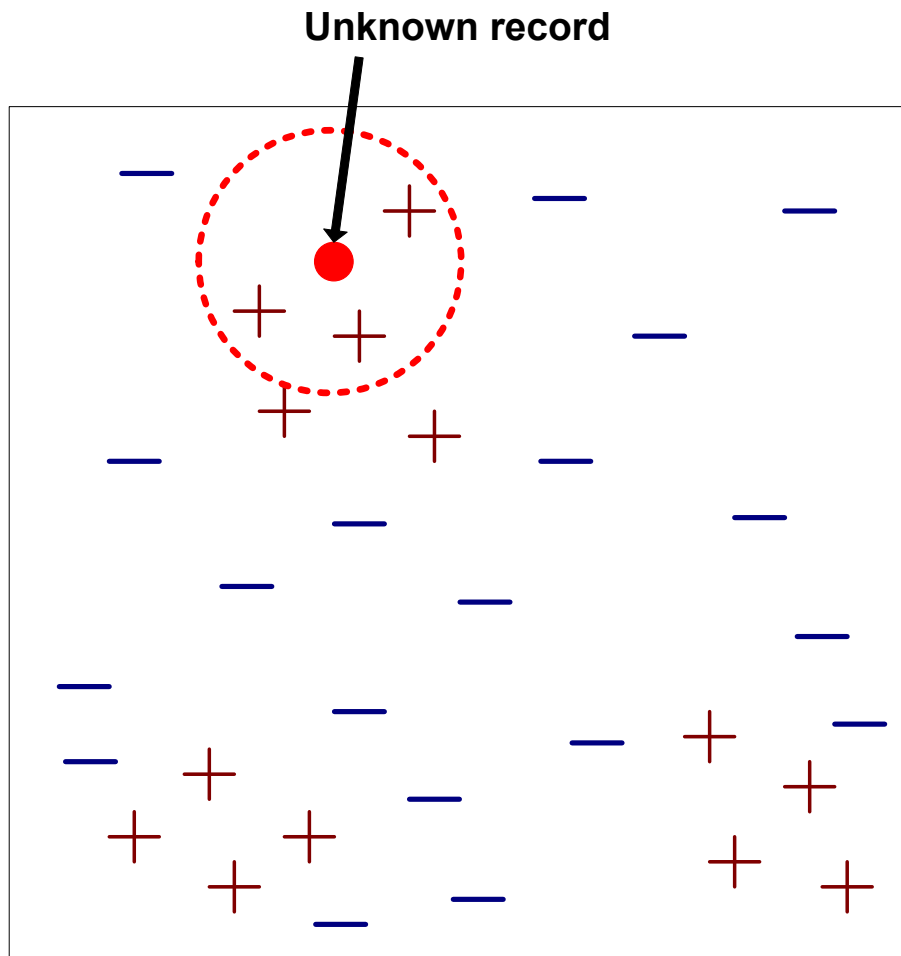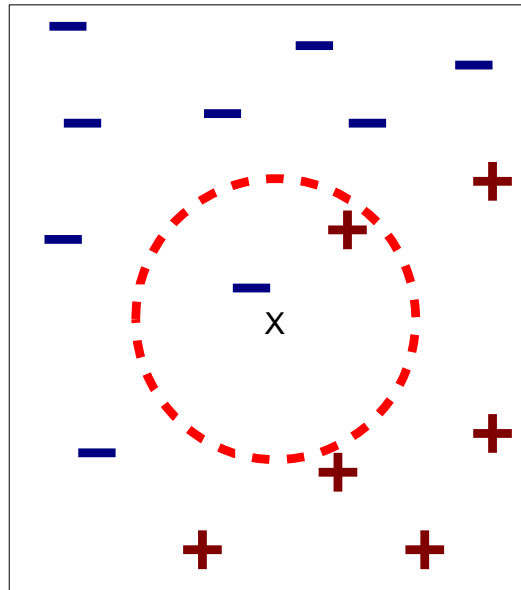
Unknown record

- Requires three things
  - The set of stored records
  - Distance Metric to compute distance between records
  - The value of k, the number of nearest neighbors to retrieve

- To classify an unknown record:
  - Compute distance to other training records
  - Identify k nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of the unknown record (e.g., by taking majority vote)
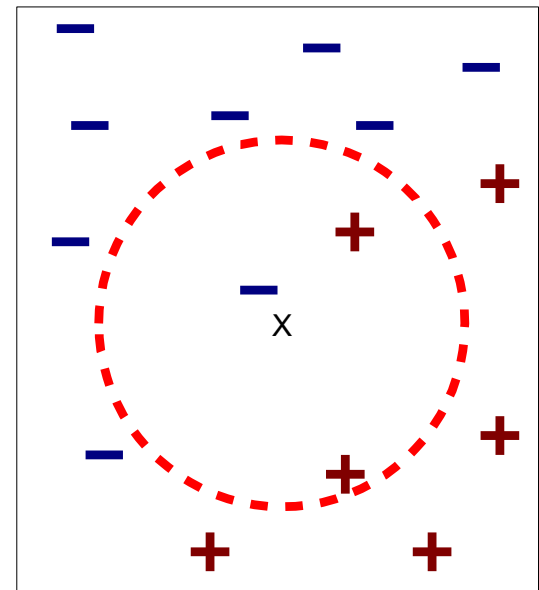
Definition of Nearest Neighbor
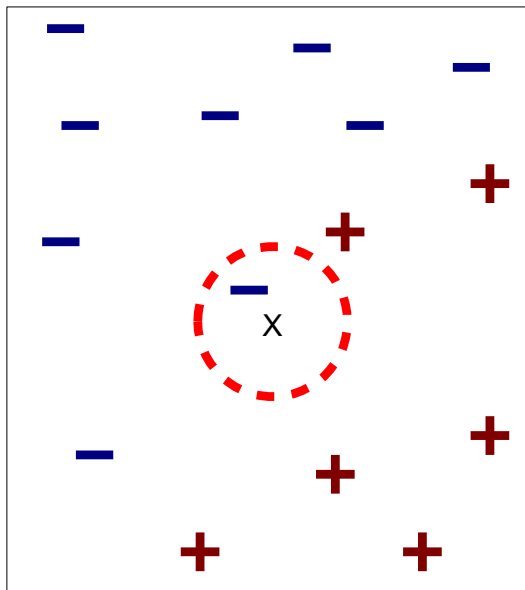
(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

- K-nearest neighbors of a record *x* are data points that have the K smallest distances to *x*

- Parameter K is a crucial choice for the effectiveness of the method

Nearest Neighbor Classification

- **Compute distance between two points:**
  - **Euclidean distance**

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- **Determine the class from nearest neighbor list**
  - **take the majority vote of class labels among the k-nearest neighbors**
  - **Weigh the vote according to distance**
    - **weight factor, w = $1/d^2$**

Nearest Neighbor Classification…

- **Choosing the value of k:**
  - **If k is too small, sensitive to noise points**
  - **If k is too large, neighborhood may include points from other classes**
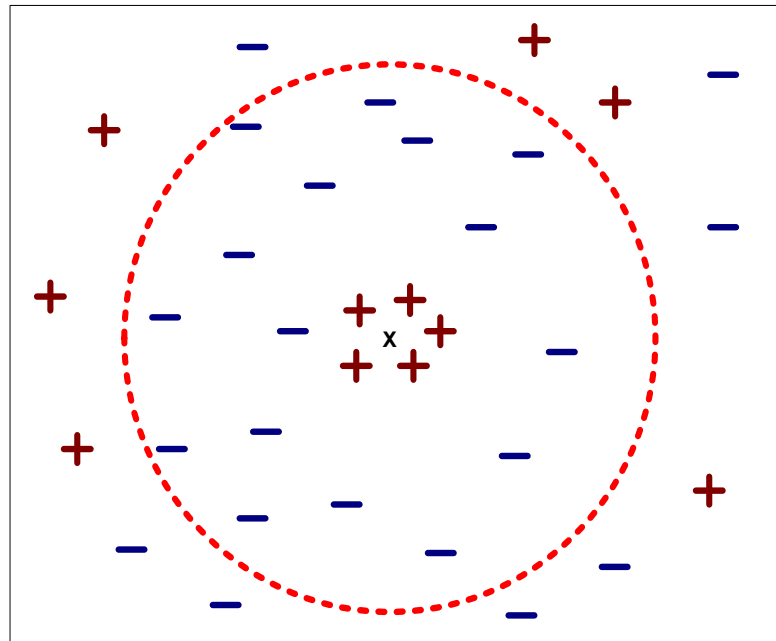
Nearest Neighbor Classification…

- **Scaling issues**
  - **Attributes may have to be scaled (normalized) to prevent distance measures from being dominated by one of the attributes**
  - **Example:**
    - **height of a person may vary from 1.5m to 1.8m**
    - **weight of a person may vary from 90lb to 300lb**
    - **income of a person may vary from $10K to $1M**

Nearest neighbor Classification…

- **k-NN classifiers are lazy learners**
  - **They do not build models explicitly**
  - **Unlike eager learners such as decision tree and rule-based systems**
  - **Classifying unknown records are relatively expensive, and may be made faster by building indexes over the training of data**
    - **Issues deriving from the huge data dimensionality**

Bayes Classifier

- **A probabilistic framework for solving classification problems**

- **Conditional Probability:**

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

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

C

A
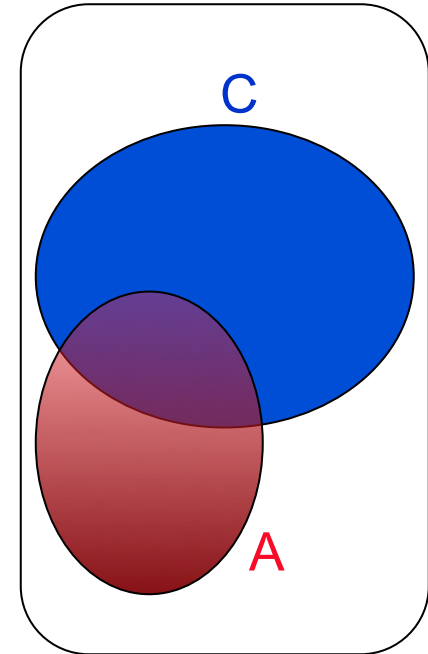
- **Bayes theorem:**

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

The posterior probability **that the hypothesis of class C holds given the observed data sample A**

P(C) **is the** prior probability**, or a priori probability of class C**

23

Example of Bayes Theorem

- **Given:**
  - **A doctor knows that meningitis causes stiff neck 50% of the time**
  - **Prior probability of any patient having meningitis is 1/50,000**
  - **Prior probability of any patient having stiff neck is 1/20**

- **If a patient has stiff neck, what's the probability he/she has meningitis?**

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Classifiers

- **Consider each attribute and class label as random variables**

- **Given a record with attributes $(A_1, A_2, \ldots, A_n)$**
  - **Goal is to predict class C**
  - **Specifically, we want to find the value of C that maximizes $P(C \mid A_1, A_2, \ldots, A_n)$**

- **Can we estimate $P(C \mid A_1, A_2, \ldots, A_n)$ 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:**

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

    - **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: $P(C) = N_c/N$**
  - e.g., $P(No) = 7/10$, $P(Yes) = 3/10$

- **For discrete attributes:**

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

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

    **$P(Status=Married|No) = 4/7$
    $P(Refund=Yes|Yes)=0$**

How to Estimate Probabilities from Data?

- **For continuous attributes:**
    - **Discretize** the range into bins
        - one ordinal attribute per bin
        - violates independence assumption
    - **Two-way split:** (A < v) or (A > v)
        - choose only one of the two splits as new attribute
    - **Probability density estimation:**
        - Assume attribute follows a normal distribution
        - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
        - Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$

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

- **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)
  $\times$ P(Married| Class=No)
  $\times$ P(Income=120K| Class=No)
  = 4/7 $\times$ 4/7 $\times$ 0.0072 = 0.0024

- P(X|Class=Yes) = P(Refund=No| Class=Yes)
  $\times$ P(Married| Class=Yes)
  $\times$ P(Income=120K| Class=Yes)
  = 1 $\times$ 0 $\times$ 1.2 $\times$ 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 Classifier

- **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 Naïve Bayes Classifier

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

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

A: attributes

M: mammals

N: non-mammals

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

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

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

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

P(A|M)P(M) > P(A|N)P(N)

**=> Mammals**

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**
  - **Use other techniques such as Bayesian Belief Networks (BBN)**

Artificial Neural Networks (ANN)

X₁	X₂	X₃	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0
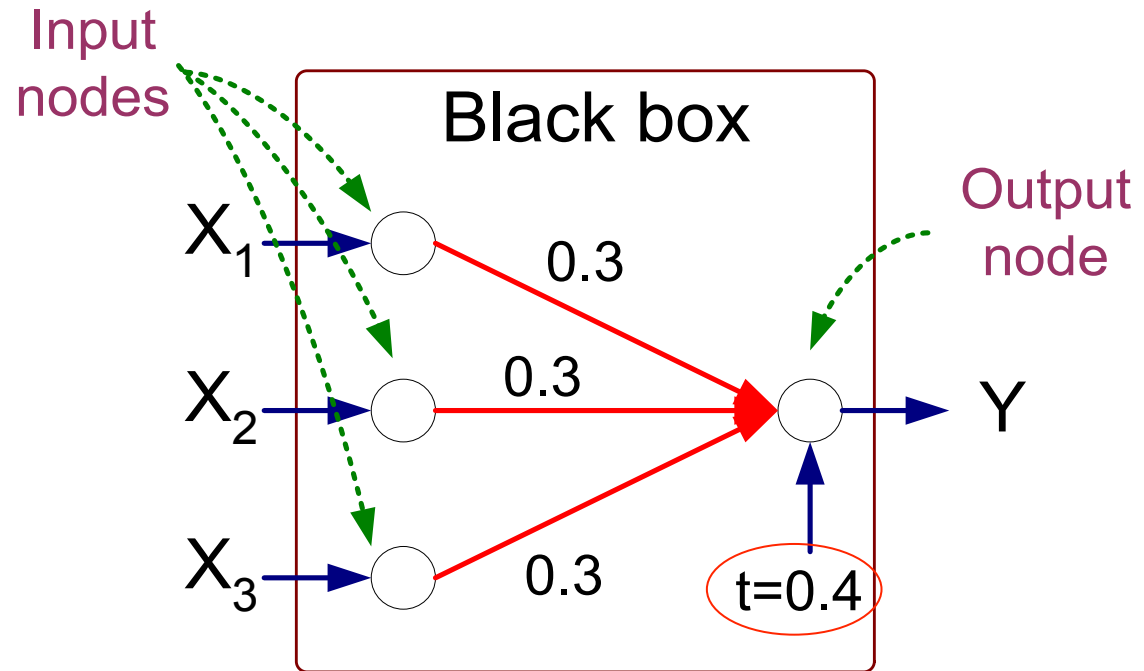
$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

$$\text{where } I(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Artificial Neural Networks (ANN)

- **Model is an assembly of inter-connected nodes and weighted links**

- **Output node sums up each of its input value according to the weights of its links**
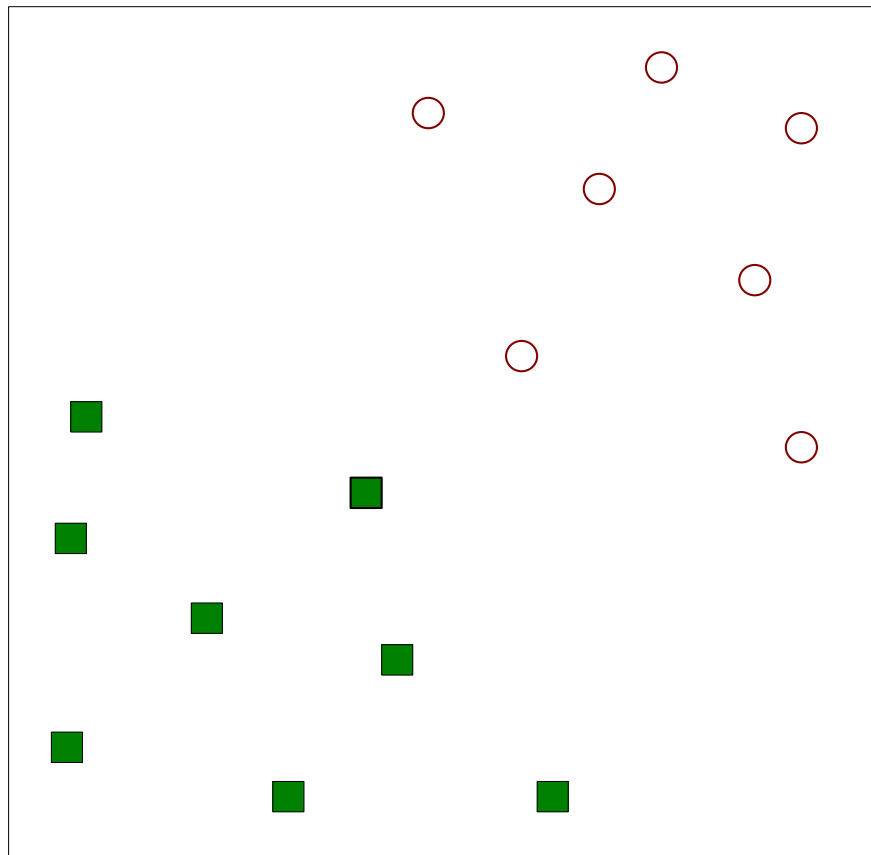
- **Compare output node against some threshold t**

Input nodes

Black box

Output node

X_1 w_1

X_2 w_2 Y

X_3 w_3

t
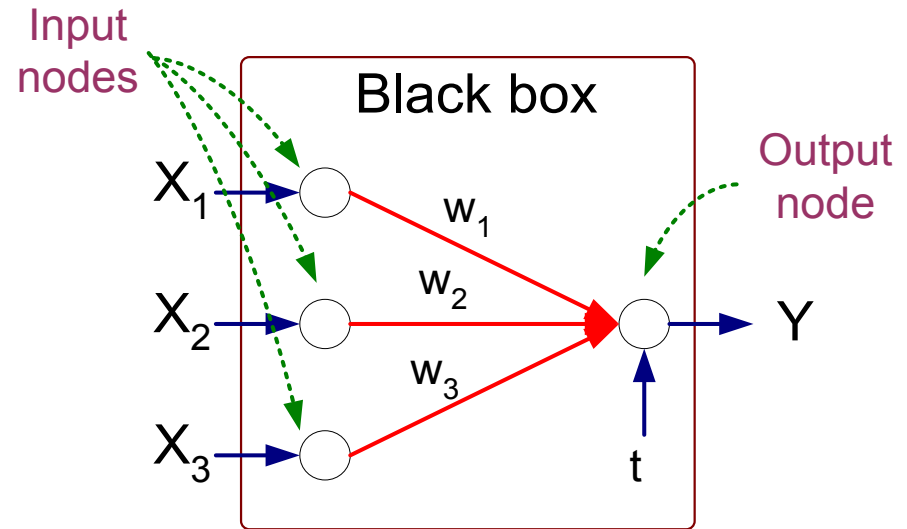
Perceptron Model

$$Y = I(\sum_i w_i X_i - t) \quad \text{or}$$
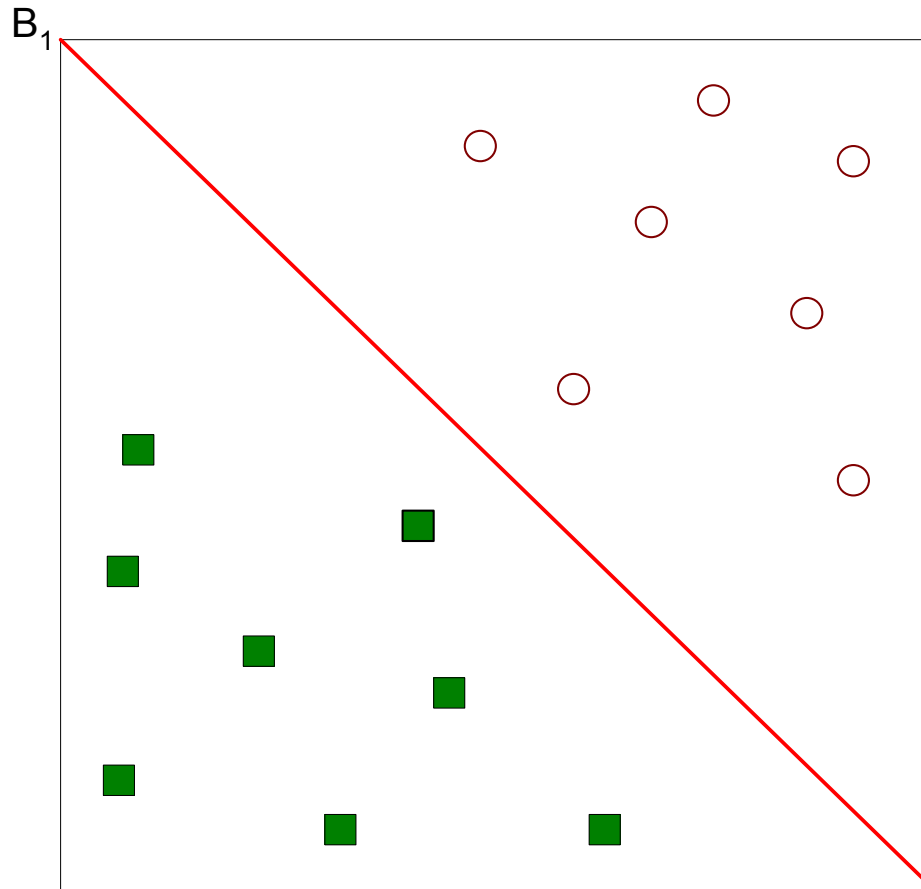
$$Y = sign(\sum_i w_i X_i - t)$$
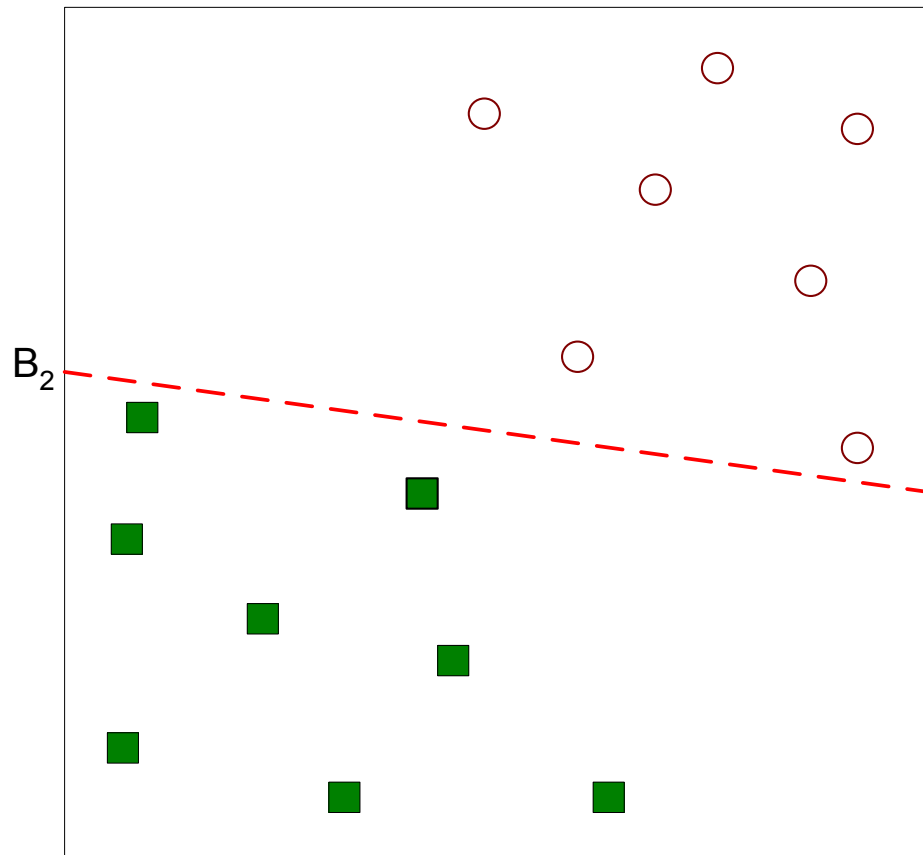
Support Vector Machines

- **Find a linear hyperplane (decision boundary) that will separate the data**
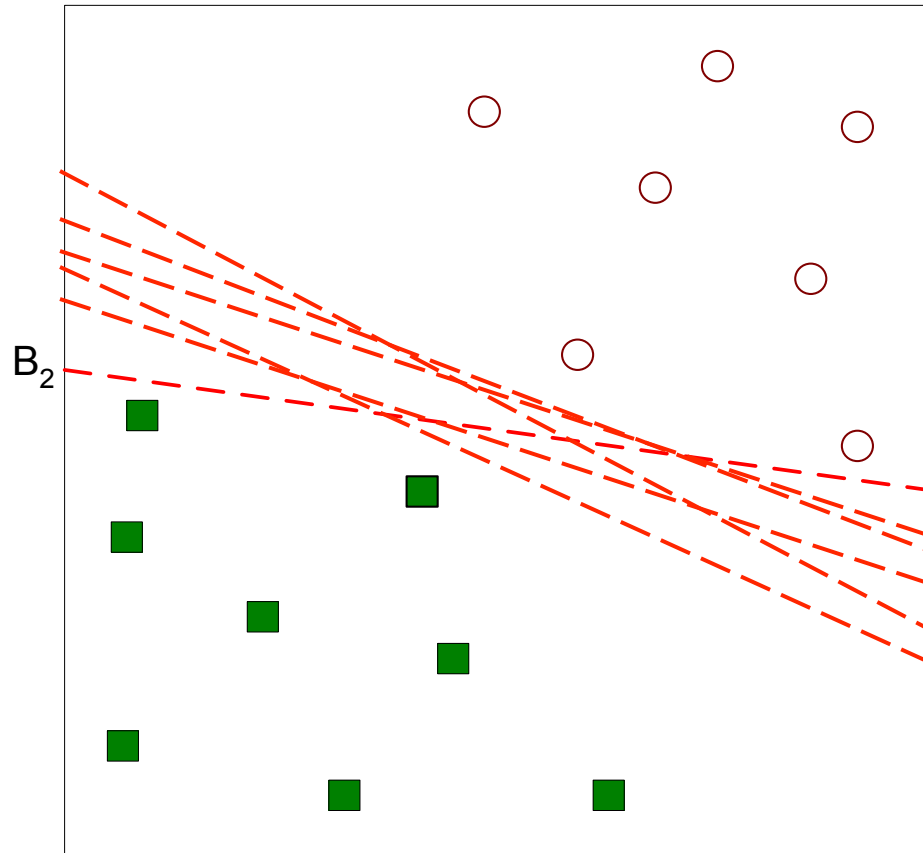
Support Vector Machines
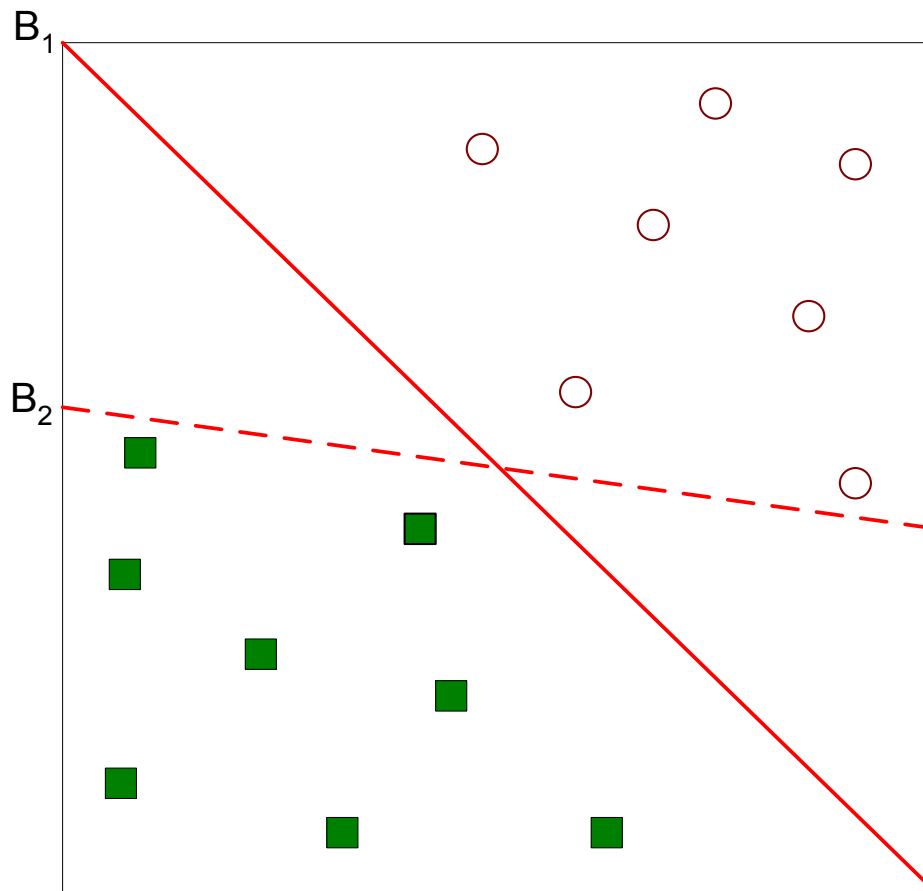

- **One Possible Solution**

Support Vector Machines

- **Another possible solution**

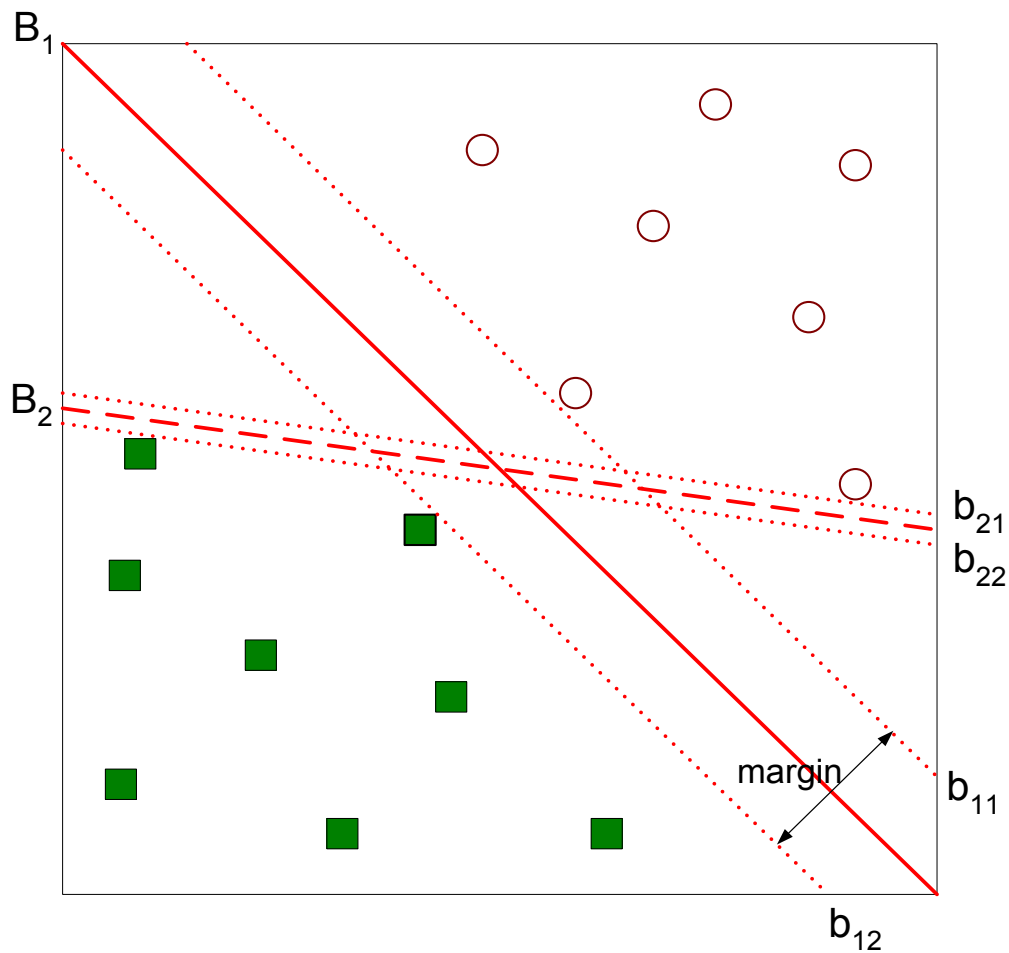# Support Vector Machines

- **Other possible solutions**

Support Vector Machines

- **Which one is better? B1 or B2?**
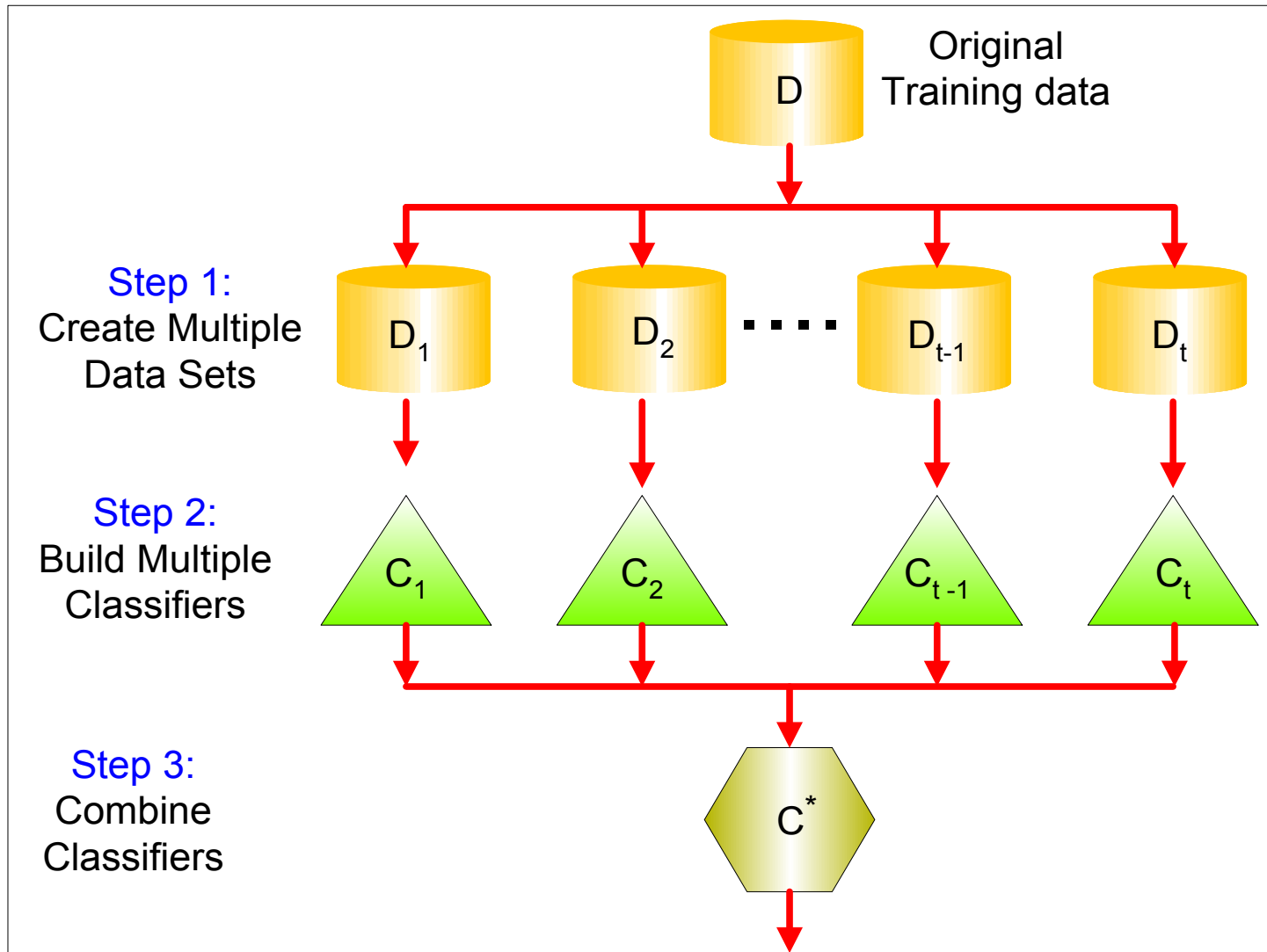- **How do you define better?**

Support Vector Machines

- **Find hyperplane maximizes the margin => B1 is better than B2**

Ensemble Methods

- **Construct a set of classifiers from the training data**

- **Predict class label of previously unseen records by aggregating predictions made by multiple classifiers**

General Idea

Original Training data: **D**

Step 1: Create Multiple Data Sets: D_1, D_2, $\cdots$, D_{t-1}, D_t

Step 2: Build Multiple Classifiers: C_1, C_2, C_{t-1}, C_t

Step 3: Combine Classifiers: C^*

Why does it work?

- **Suppose there are 25 base classifiers**
  - **Each classifier has error rate, $\varepsilon = 0.35$**
  - **Assume classifiers are independent, and that the majority of them (at least 13 of 25) make the wrong prediction**
  - **Probability that the ensemble classifier makes a wrong prediction:**

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

Examples of Ensemble Methods

- **How to generate an ensemble of classifiers?**
  - **Bagging**

  - **Boosting**

Bagging

- **Sampling with replacement**

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- **Build classifier on each bootstrap sample, whose size *n* is the same as the original data set**

- **Each item has probability $(1 - 1/n)^n$ of <u>NOT being</u> selected in a bootstrap composed of *n* items**
  - **So, each item has probability $1 - (1 - 1/n)^n$ of <u>being</u> selected in a bootstrap**
  - **For large *n* this converge to $1-1/e \cong 0.632$**
  - **Thus on average each bootstrap contains approximately 63% of the original training set**

Boosting

- **An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records**
  - Initially, all *n* records are assigned equal weights
  - Unlike bagging, weights may change at the end of boosting round

Boosting

- **Records that are wrongly classified will have their weights increased**

- **Records that are classified correctly will have their weights decreased**

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

• Example 4 is hard to classify

• Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds
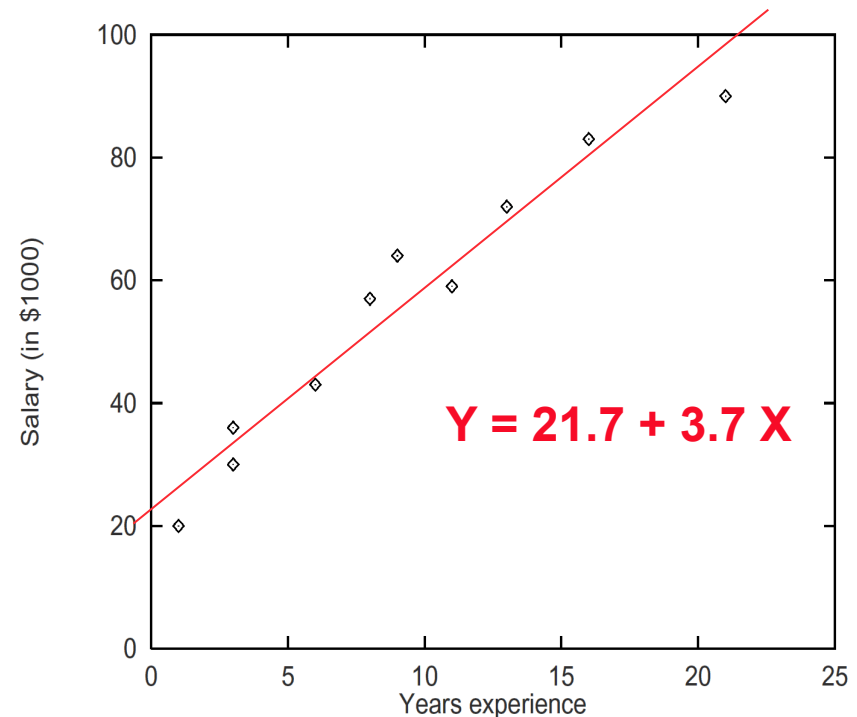
Prediction (vs. Classification)

- *"What if we would like to predict a continuous value, rather than a categorical label?"*
  - The prediction of continuous values can be modeled by statistical techniques
  - We need to define a model / function

- **The main prediction method is the regression**
  - Linear and multiple regression
  - Non-linear regression (nonlinear problem can be converted to a linear one)

Prediction

- **Linear Regression:** $Y = \alpha + \beta X$
  - **We need to predict Y as a linear function of X**
  - **Two parameters, α and β specify the line and are to be estimated by using the data at hand.**
  - **using <u>the least squares</u> criterion to the known values of $Y_1, Y_2, \ldots, X_1, X_2,$**

X years experience	Y salary (in \$1000)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

Y = 21.7 + 3.7 X

Summary

- **Classification is an extensively studied problem (mainly in statistics, machine learning & neural networks)**

- **Classification is probably one of the most widely used data mining techniques with a lot of extensions**

- **Scalability is still an important issue for database applications: thus combining classification with database techniques should be a promising topic**

- **Research directions: classification of non-relational data, e.g., text, spatial, multimedia, etc..**