

Week 4

Classification (Part II)

Seokho Chi

Professor | Ph.D.

SNU Construction Innovation Lab

Source: Tan, Kumar, Steinback (2006)



Rule-Based Classifier

- Classify records by using a collection of “if...then...” rules
- Rule: $(Condition) \rightarrow y$
 - where
 - *Condition* is a conjunctions of attributes
 - y is the class label
 - *LHS*: rule antecedent or condition
 - *RHS*: rule consequent
 - Examples of classification rules:
 - $(Blood\ Type=Warm) \wedge (Lay\ Eggs=Yes) \rightarrow Birds$
 - $(Taxable\ Income < 50K) \wedge (Refund=Yes) \rightarrow Evade=No$

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) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles (파충류)

R5: (Live in Water = sometimes) \rightarrow Amphibians (양서류)

Application of Rule-Based Classifier

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

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow 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 a hawk \Rightarrow Bird

The rule R3 covers the grizzly bear (회색곰) \Rightarrow Mammal

Rule Coverage and Accuracy

- Coverage of a rule:
 - Fraction of records that satisfy the antecedent of a rule
- Accuracy of a rule:
 - Fraction of records that satisfy both the antecedent and consequent of a rule

<i>Tid</i>	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) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow 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(여우원숭이) triggers rule R3, so it is classified as a mammal

A turtle triggers both R4 and R5

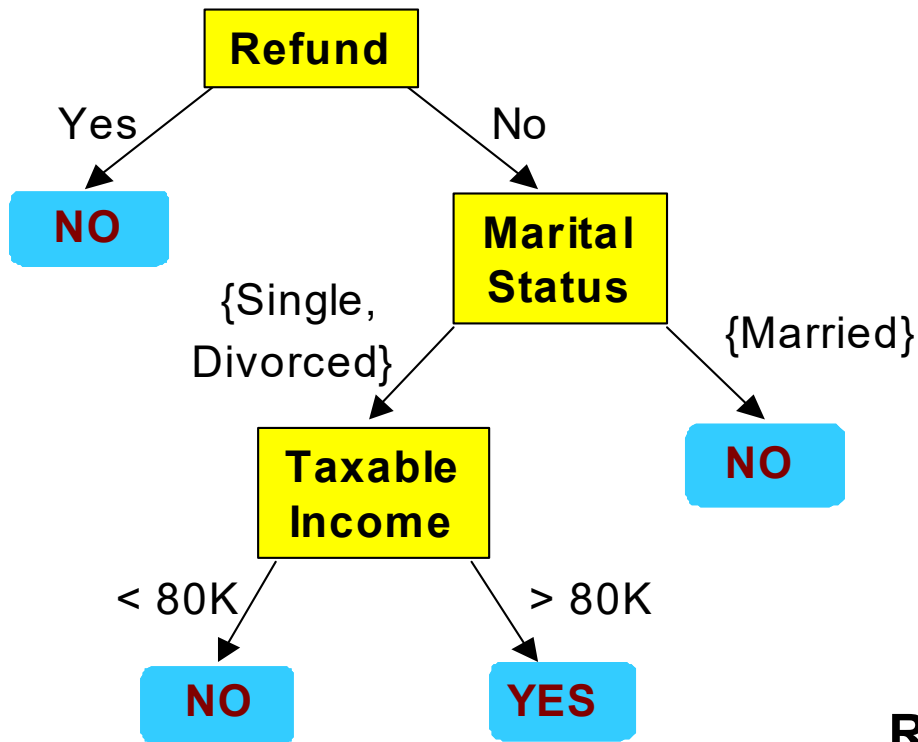
A dogfish shark(곰상어) triggers none of the rules

Characteristics of Rule-Based Classifier

- Mutually exclusive rules (상호배타적)
 - Classifier contains mutually exclusive rules **if the rules are independent of each other**
 - No two rules are triggered by the same record
 - Every record is covered by at most one rule
- Exhaustive rules (포괄적)
 - Classifier has exhaustive coverage **if it accounts for every possible combination** of attribute values
 - Each record is covered by at least one rule

*By combining two characteristics,
every record is exactly matched with one rule.*

From Decision Trees To Rules



Classification Rules

(Refund=Yes) ==> No

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

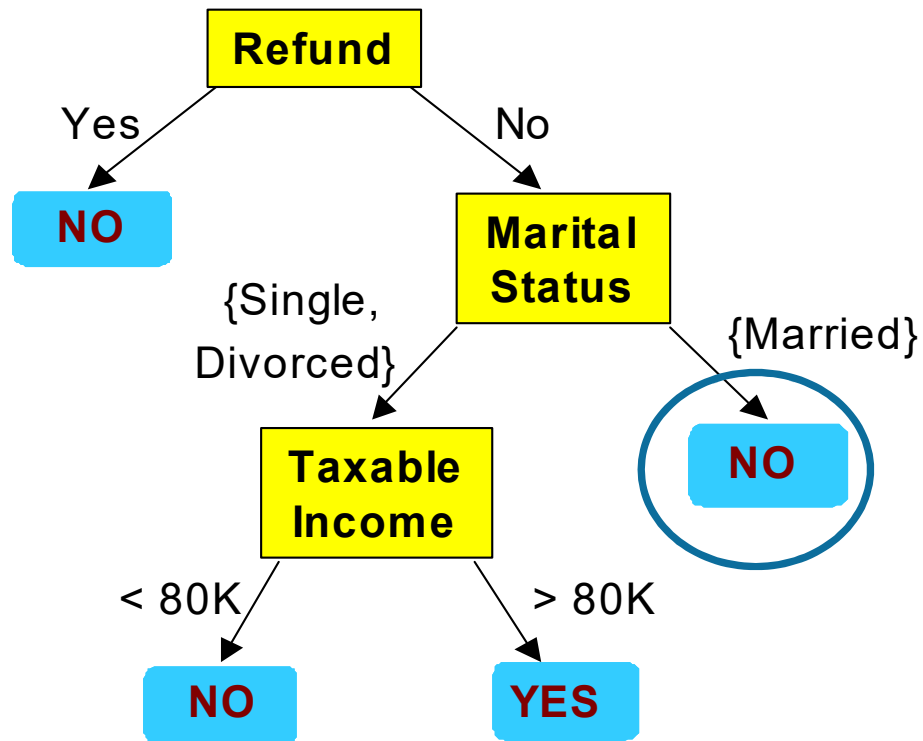
(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(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



<i>Tid</i>	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: $(\text{Refund}=\text{No}) \wedge (\text{Status}=\text{Married}) \rightarrow \text{No}$

Simplified Rule: $(\text{Status}=\text{Married}) \rightarrow \text{No}$

Effect of Rule Simplification

- Rules are no longer exhaustive
 - A record may not trigger any rules
 - Solution?
 - Add a default rule to assign the default class for remainders

$$r_d : () \rightarrow y_d$$

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

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

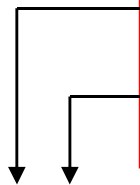
R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow Amphibians



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

Can avoid mutually exclusive problems

However, prioritization requires computing and the results are sensitive to the decision list.

Unordered Rule Set

- Allow one record is classified by multiple rules
- Each prediction result works as voting
- Voting counts determine the class

*No need for prioritization (less sensitive to wrong rules)
Each record needs to be compared with every rule → Take longer time*

Rule Ordering Schemes

■ Rule-based ordering

- Individual rules are ranked based on their quality (coverage & accuracy)
- Lower-ranked rules are much harder to interpret because they assume the negation of the rules preceding them. (no 1, 2, 3, 4,..., 9 → then 10)

■ Class-based ordering

- Rules that belong to the same class appear together
- Poor rules appeared first can be applied instead of good rules appeared late

Rule-based Ordering

(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

Class-based Ordering

(Refund=Yes) ==> No

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

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

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

Building Classification Rules

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

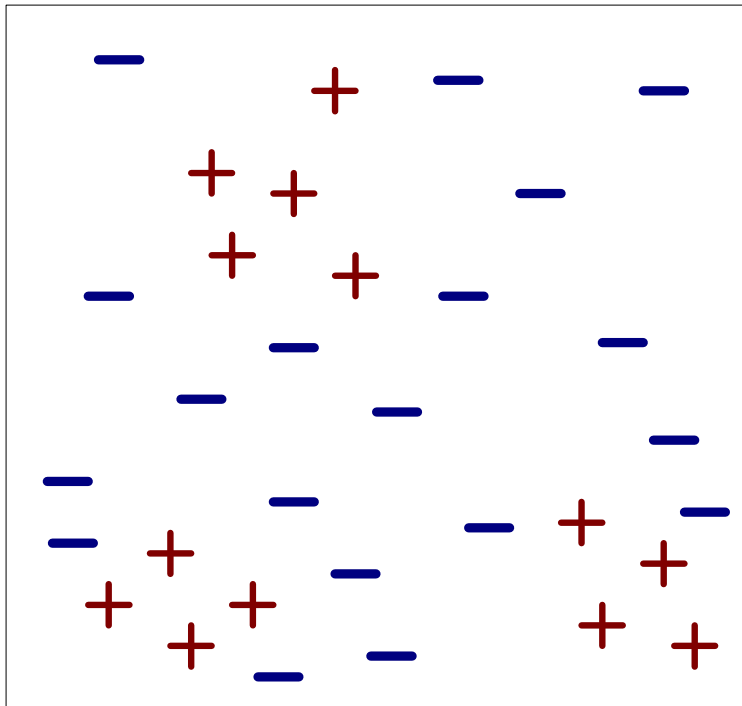
- **Indirect Method:**
 - Extract rules from other classification models (e.g. decision trees, neural networks, etc.).
 - e.g.: C4.5rules

Direct Method: Sequential Covering

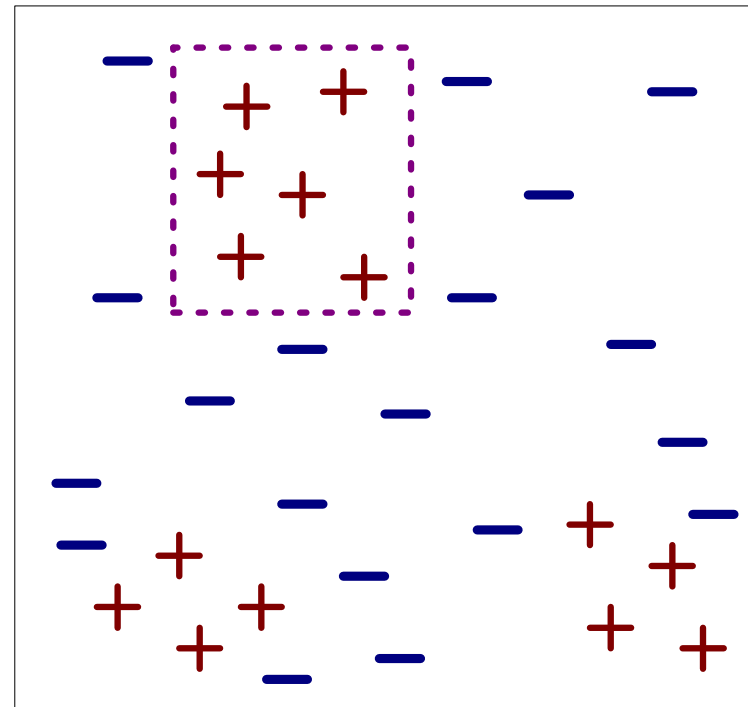
1. Start from an empty decision list
2. The Learn-One-Rule function is then used to extract the best rule for class y (positive examples, other classes: negative examples)
3. Remove training records covered by the rule
4. Repeat Step (2) and (3) until stopping criterion is met
5. The algorithm then proceeds to generate rules for the next class

Example of Sequential Covering

Fraction of positive examples



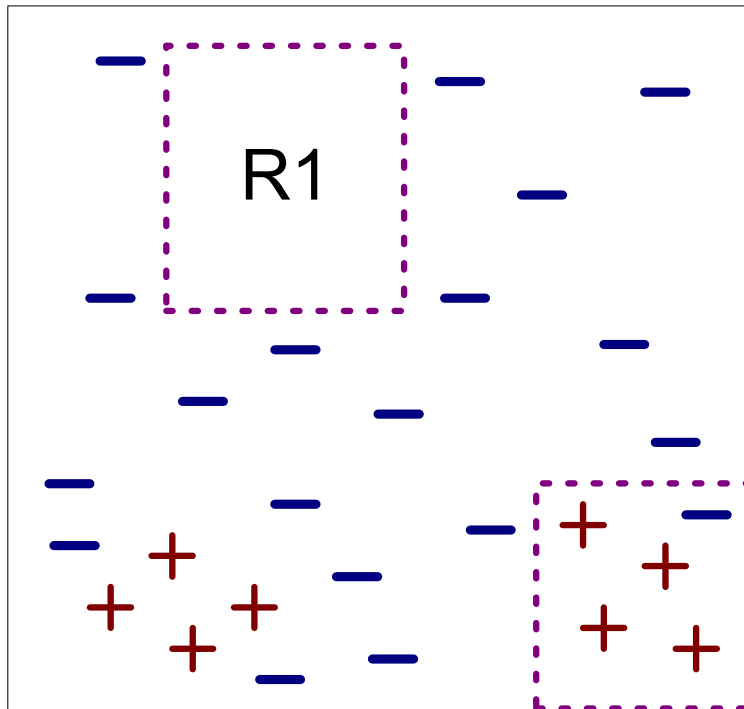
(i) Original Data



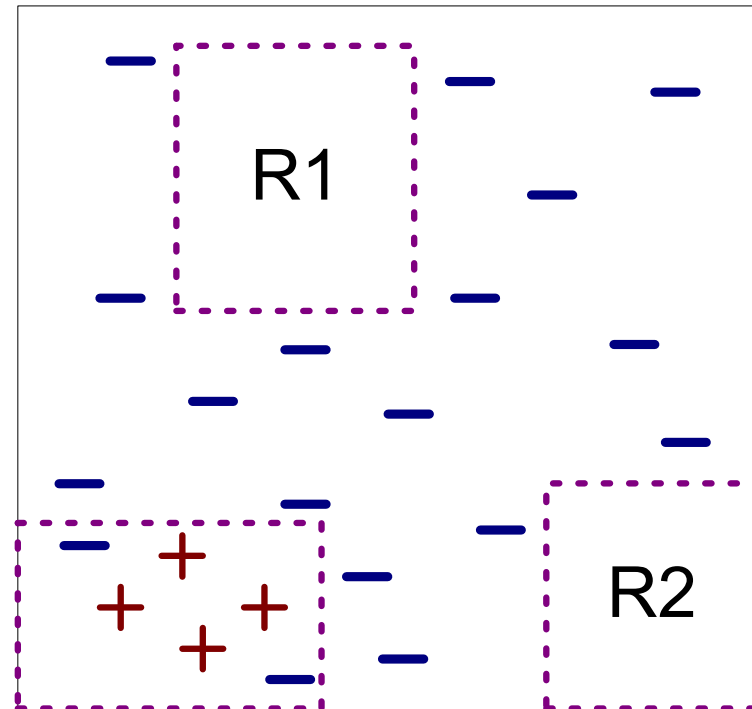
(ii) Step 1

Example of Sequential Covering...

Rule creation and removed examples



(iii) Step 2



(iv) Step 3

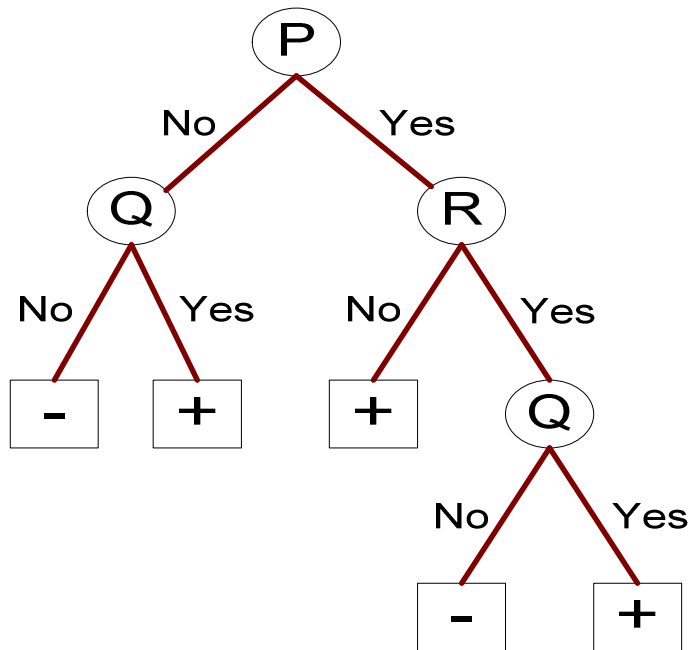
Direct Method: RIPPER

- Work good with imbalanced class distribution
- Work good with noisy data sets since it uses a validation set to prevent model overfitting
- Let (y_1, y_2, \dots, y_c)
 - y_1 : the least frequent class (positive examples) \rightarrow rule for detecting minority class
 - y_c : the most frequent class (negative examples) \rightarrow default
 - Sequential covering to generate rules that discriminate between the positive and negative examples
 - Then, extract rules that distinguish y_2 from other remaining classes

Direct Method: RIPPER

- Building a rule set
 - After generating a rule, all the positive and negative examples covered by the rule are eliminated
 - The rule is then added into the rule set as long as it does not violate the stopping condition
 - Minimum description length principle : if the new rule increase the total description length of the rule set by as least d bits (by default, 64) then RIPPER stops adding rules into its rule set
 - Also, the error rate of the rule on the validation set must not exceed 50%

Indirect Methods



Rule Set

r1: (P=No, Q=No) ==> -

r2: (P=No, Q=Yes) ==> +

r3: (P=Yes, R=No) ==> +

r4: (P=Yes, R=Yes, Q=No) ==> -

r5: (P=Yes, R=Yes, Q=Yes) ==> +

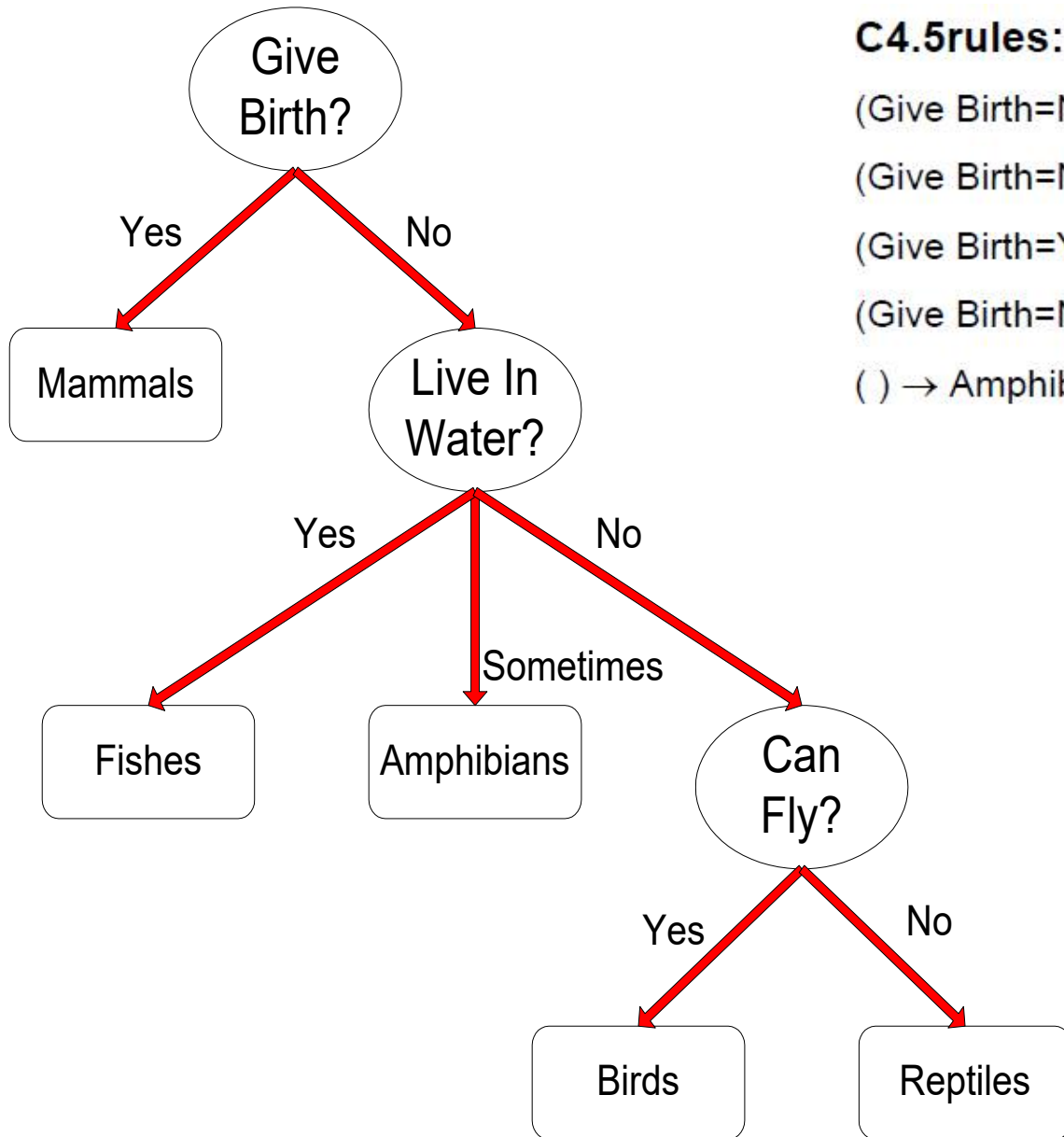
Indirect Method: C4.5rules

- Extract rules from an unpruned decision tree
- For each rule, $r: A \rightarrow y$,
 - Consider a simplified rule $r': A' \rightarrow y$ where A' is obtained by removing one of the conjuncts(결합항) in A
 - Compare the pessimistic error rate for r against all r' 's
 - Prune A if one of the r' 's has lower pessimistic error rate than the original rule
 - Repeat until we can no longer improve generalization error

Example

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

C4.5 versus C4.5rules versus RIPPER



C4.5rules:

(Give Birth=No, Can Fly=Yes) → Birds

(Give Birth=No, Live in Water=Yes) → Fishes

(Give Birth=Yes) → Mammals

(Give Birth=No, Can Fly=No, Live in Water=No) → Reptiles

() → Amphibians

RIPPER:

(Live in Water=Yes) → Fishes

(Can Fly=No) → Reptiles

(Give Birth=No, Can Fly=No, Live In Water=No)
→ Reptiles

(Can Fly=Yes, Give Birth=No) → Birds

() → Mammals

C4.5 versus C4.5rules versus RIPPER

C4.5 and C4.5rules:

		PREDICTED CLASS				
		Amphibians	Fishes	Reptiles	Birds	Mammals
ACTUAL CLASS	Amphibians	2	0	0	0	0
	Fishes	0	2	0	0	1
	Reptiles	1	0	3	0	0
	Birds	1	0	0	3	0
	Mammals	0	0	1	0	6

RIPPER:

		PREDICTED CLASS				
		Amphibians	Fishes	Reptiles	Birds	Mammals
ACTUAL CLASS	Amphibians	0	0	0	0	2
	Fishes	0	3	0	0	0
	Reptiles	0	0	3	0	1
	Birds	0	0	1	2	1
	Mammals	0	2	1	0	4

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

- Memory-based learning → lazy learning
- Store the training records
- Use training records to predict the class label of unseen cases

Unseen Case

Atr1	AtrN

It does not build models explicitly

**Eager learning: training data → model development → predict testing data (e.g., decision tree)*

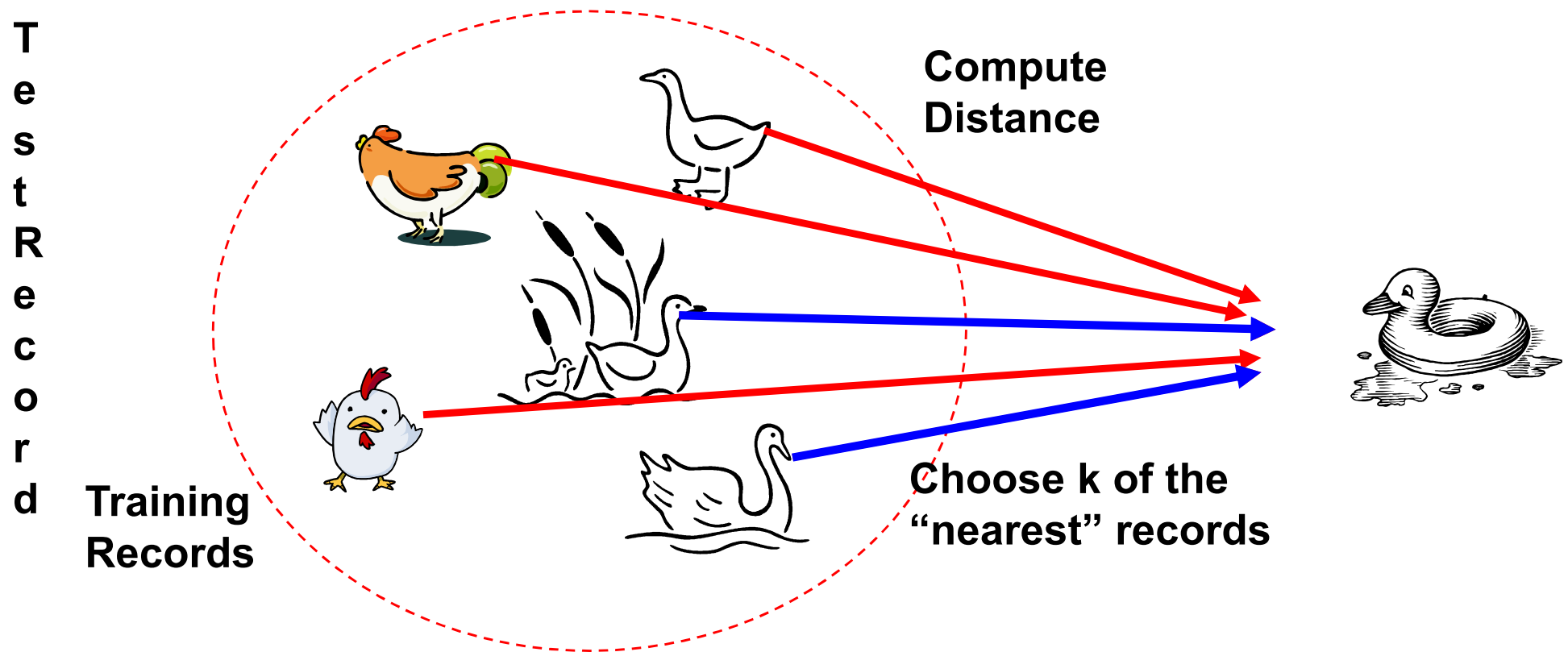
**Lazy learning: comparing similarities b/w training and testing records (store the entire training dataset → no abstraction is made) Instance-Based Classifier*

Instance Based Classifiers

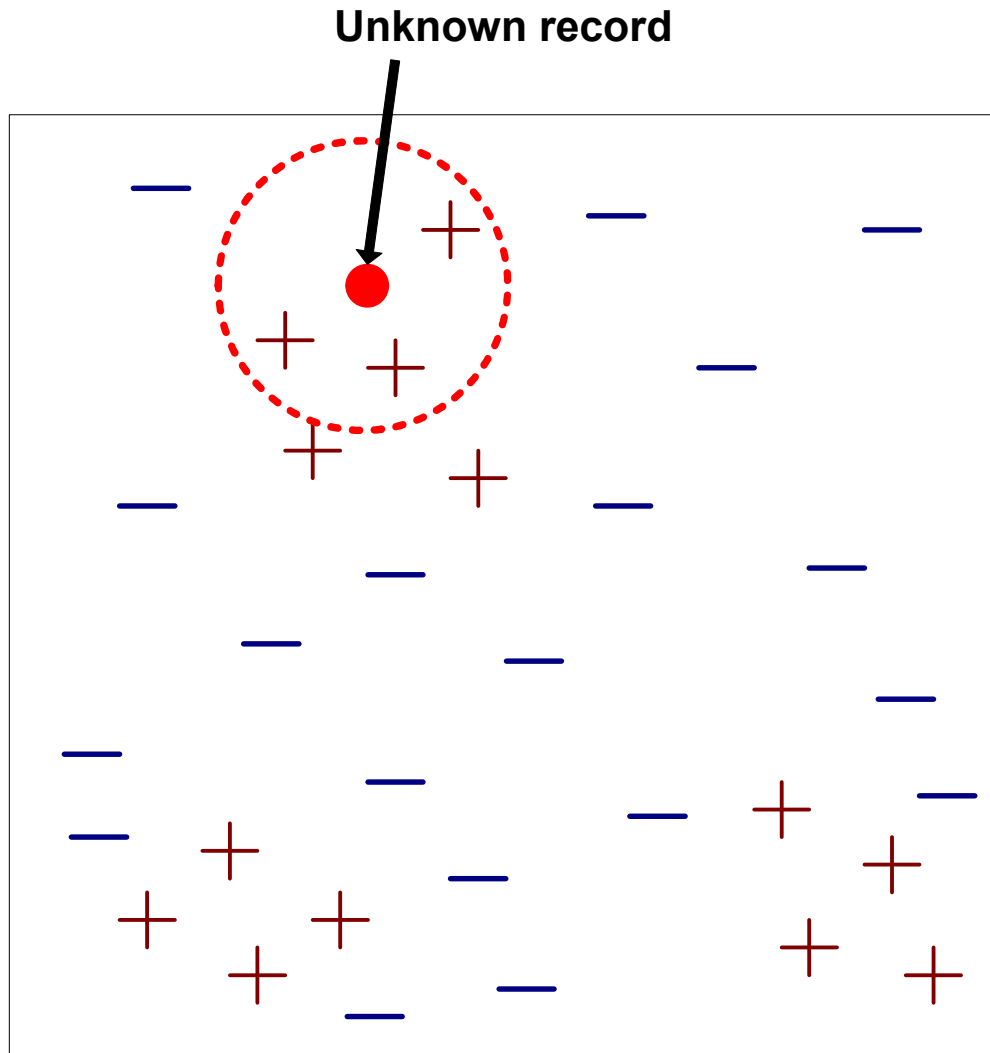
- Examples:
 - Rote-learner
 - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly
 - Nearest neighbor
 - Uses k "closest" points (nearest neighbors) for performing classification

Nearest Neighbor Classifiers

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

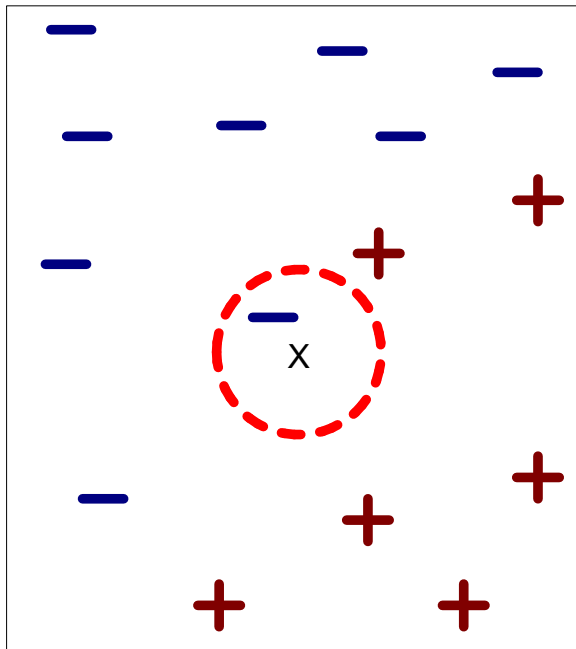


Nearest Neighbor Classifiers

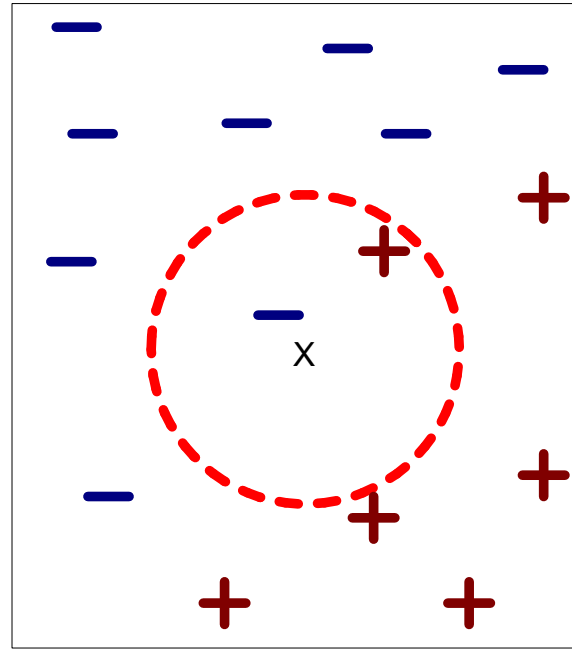


- Requires three things
 - The set of stored records
 - Distance Metric to compute distance between records
 - The value of k , the number of nearest neighbors to retrieve
- To classify an unknown record
 - Compute distance to other training records
 - Identify k nearest neighbors
 - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

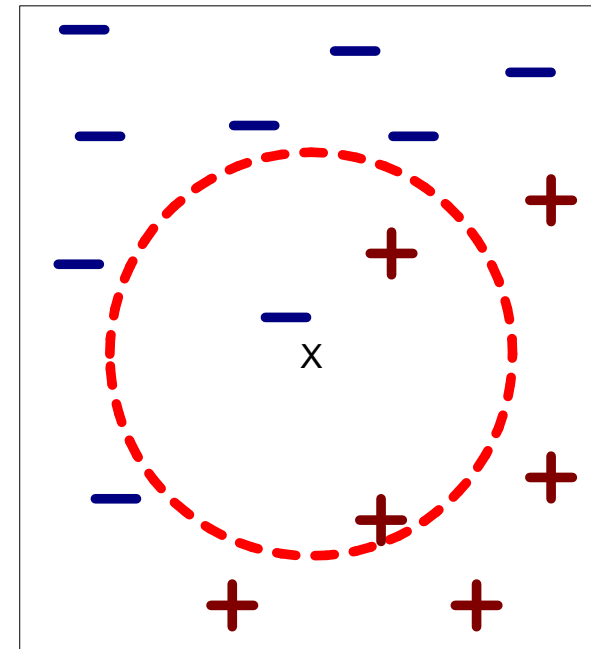
Definition of Nearest Neighbor



(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 distance to x

Nearest Neighbor Classification

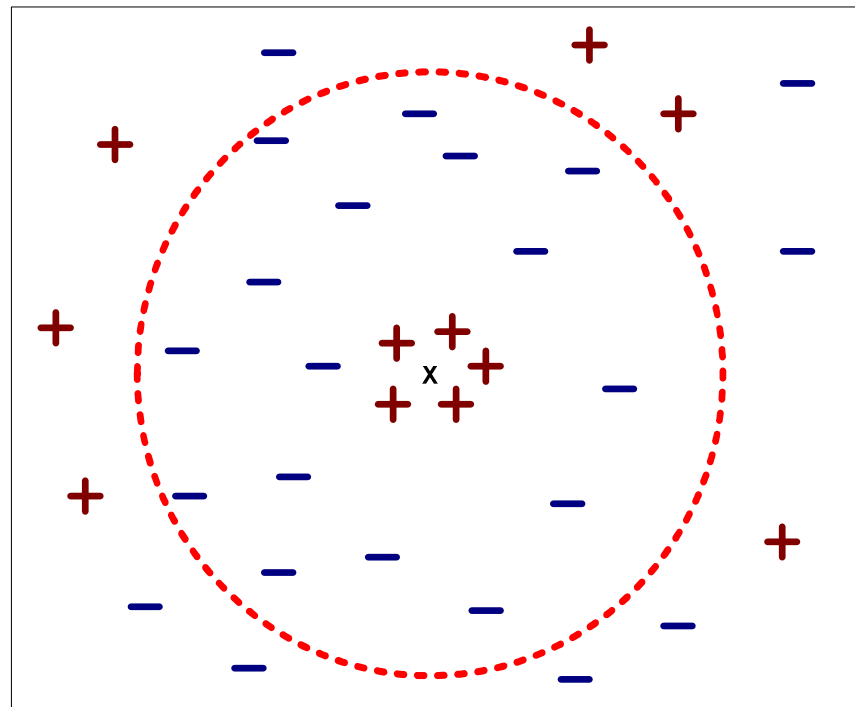
- Compute distance between two points:
 - Euclidean distance

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

- 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 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 40kg to 150kg
 - income of a person may vary from \$10K to \$1M

Nearest Neighbor (Summary)

- k-NN classifiers are lazy learners
 - It does not build models explicitly
 - Unlike eager learners such as decision tree induction and rule-based systems
 - Classifying unknown records are relatively expensive

Bayesian Classifier

- A probabilistic framework for solving classification problems

- Conditional Probability: $P(C | A) = \frac{P(A, C)}{P(A)}$

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

- Bayes theorem:

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

Example of Bayes Theorem

- Given:

뇌수막염

- 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?

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

Bayesian Classifiers

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

Bayesian Classifiers

- Approach:

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

$$P(C | A_1 A_2 \dots A_n) = \frac{P(A_1 A_2 \dots A_n | C) P(C)}{P(A_1 A_2 \dots A_n)}$$

- Choose value of C that maximizes

$$P(C | A_1, A_2, \dots, A_n)$$

- Equivalent to choosing value of C that maximizes

$$P(A_1, A_2, \dots, A_n | C) P(C)$$

- How to estimate $P(A_1, A_2, \dots, A_n | C)$?

Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given (attributes are independent each other):
 - $P(A_1, A_2, \dots, A_n | C) = P(A_1 | C_j) P(A_2 | C_j) \dots P(A_n | C_j)$
 - Can estimate $P(A_i | C_j)$ for all A_i and C_j .
 - New point is classified to C_j if $P(C_j) \prod P(A_i | C_j)$ is maximal.

How to Estimate Probabilities from Data?

categorical
categorical
continuous
class

<i>Tid</i>	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., $P(\text{No}) = 7/10$,
 $P(\text{Yes}) = 3/10$

- For discrete attributes:

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

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

$$P(\text{Status}=\text{Married}|\text{No}) = 4/7$$

$$P(\text{Refund}=\text{Yes}|\text{Yes})=0/3$$

How to Estimate Probabilities from Data?

- For continuous attributes:
 - **Discretize** the range into bins
 - one ordinal attribute per bin
 - **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?

categorical
categorical
continuous
class

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:

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

- One for each (A_i, c_i) pair

- For (Income, Class=No):

- If Class=No

- sample mean = 110

- sample variance = 2975

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

Example of Naïve Bayes Classifier

Given a Test Record:

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

naive Bayes Classifier:

$$P(\text{Refund}=\text{Yes}|\text{No}) = 3/7$$

$$P(\text{Refund}=\text{No}|\text{No}) = 4/7$$

$$P(\text{Refund}=\text{Yes}|\text{Yes}) = 0$$

$$P(\text{Refund}=\text{No}|\text{Yes}) = 1$$

$$P(\text{Marital Status}=\text{Single}|\text{No}) = 2/7$$

$$P(\text{Marital Status}=\text{Divorced}|\text{No})=1/7$$

$$P(\text{Marital Status}=\text{Married}|\text{No}) = 4/7$$

$$P(\text{Marital Status}=\text{Single}|\text{Yes}) = 2/7$$

$$P(\text{Marital Status}=\text{Divorced}|\text{Yes})=1/7$$

$$P(\text{Marital Status}=\text{Married}|\text{Yes}) = 0$$

For taxable income:

If class=No: sample mean=110

sample variance=2975

If class=Yes: sample mean=90

sample variance=25

- $P(X|\text{Class}=\text{No}) = P(\text{Refund}=\text{No} | \text{Class}=\text{No})$
 $\times P(\text{Married} | \text{Class}=\text{No})$
 $\times P(\text{Income}=120\text{K} | \text{Class}=\text{No})$
 $= 4/7 \times 4/7 \times 0.0072 = 0.0024$
- $P(X|\text{Class}=\text{Yes}) = P(\text{Refund}=\text{No} | \text{Class}=\text{Yes})$
 $\times P(\text{Married} | \text{Class}=\text{Yes})$
 $\times P(\text{Income}=120\text{K} | \text{Class}=\text{Yes})$
 $= 1 \times 0 \times 1.2 \times 10^{-9} = 0$

Since $P(X|\text{No})P(\text{No}) > P(X|\text{Yes})P(\text{Yes})$

Therefore $P(\text{No}|X) > P(\text{Yes}|X)$

$\Rightarrow \text{Class} = \text{No}$

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

A: attributes

M: mammals

N: non-mammals

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

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

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

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

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

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

=> Mammals

Naïve Bayesian Classifier: Example

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

Classes:

C1:buys_computer= 'yes'; C2:buys_computer= 'no'

Unknown sample:

X =(age<=30, Income=medium, Student=yes, Credit rating= Fair)

Naïve Bayes (Summary)

- Advantages
 - Easy to implement
 - Good results obtained in most of the cases
- Disadvantages
 - Assumption: class conditional independence, therefore loss of accuracy
 - Practically, dependencies exist among variables
 - e.g., hospitals: patients: Profile: age, family history, etc.
Symptoms: fever, cough, etc., Disease: lung cancer, diabetes, etc.
 - Dependencies among these cannot be modeled by Naïve Bayesian Classifier

Summary

- **Decision tree**
 - Easy modeling building from training data
 - Fast prediction of testing data
 - Major issues like overfitting
- **K-NN**
 - Prediction based on majority voting (better representativeness)
 - Time and energy consumption due to every possible comparison
 - Weak to noise and outliers
- **Naïve Bayes Classifier**
 - Accurate prediction based on probability
 - Less sensitive to noise
 - Do not consider dependencies between attributes