Rule Based Classifieer

Rule-Based Classifier

- Classify records by using a collection of "if…then…" rules
- Rule: $(Condition) \rightarrow y$

  where
  - Condition is a conjunctions of attributes (calles LHS, antecedent or condition)
  - y is the class label (called RHS or 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

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

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: $hawk \rightarrow Bird$

The rule R3 covers: $grizzly\ bear \rightarrow Mammal$

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

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) $\rightarrow$ No

Coverage = 40%, Accuracy = 50%

Ordered Rule Set vs. Voting

- 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	?

- Alternative: (weighted) voting by all matching rules.

Ordered Rule Set vs. Voting

- 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	?

- Alternative: (weighted) voting by all matching rules.

Rules From Decision Trees

Rule-Based Classifier:

(Gives Birth=No, Aerial Creature=Yes) => Birds

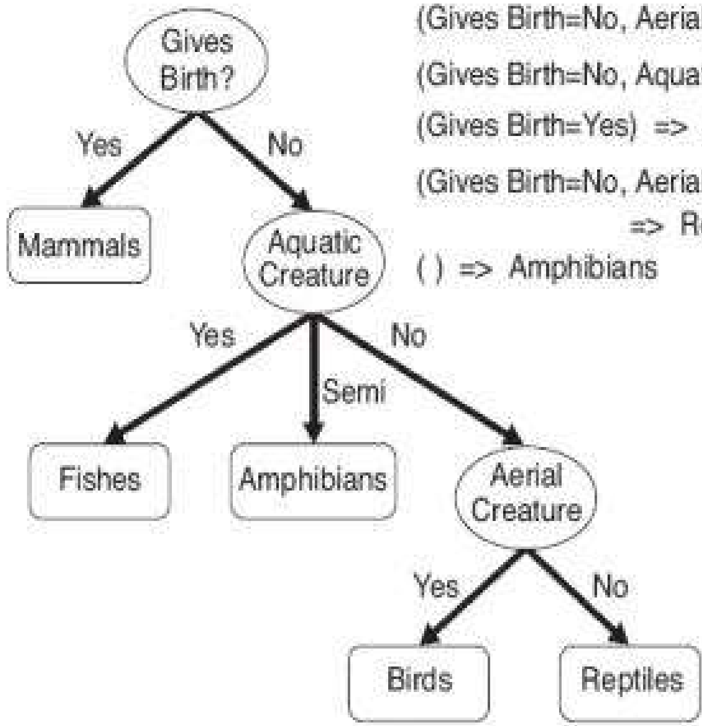
(Gives Birth=No, Aquatic Creature=Yes) => Fishes

(Gives Birth=Yes) => Mammals

(Gives Birth=No, Aerial Creature=No, Aquatic Creature=No) => Reptiles

( ) => Amphibians

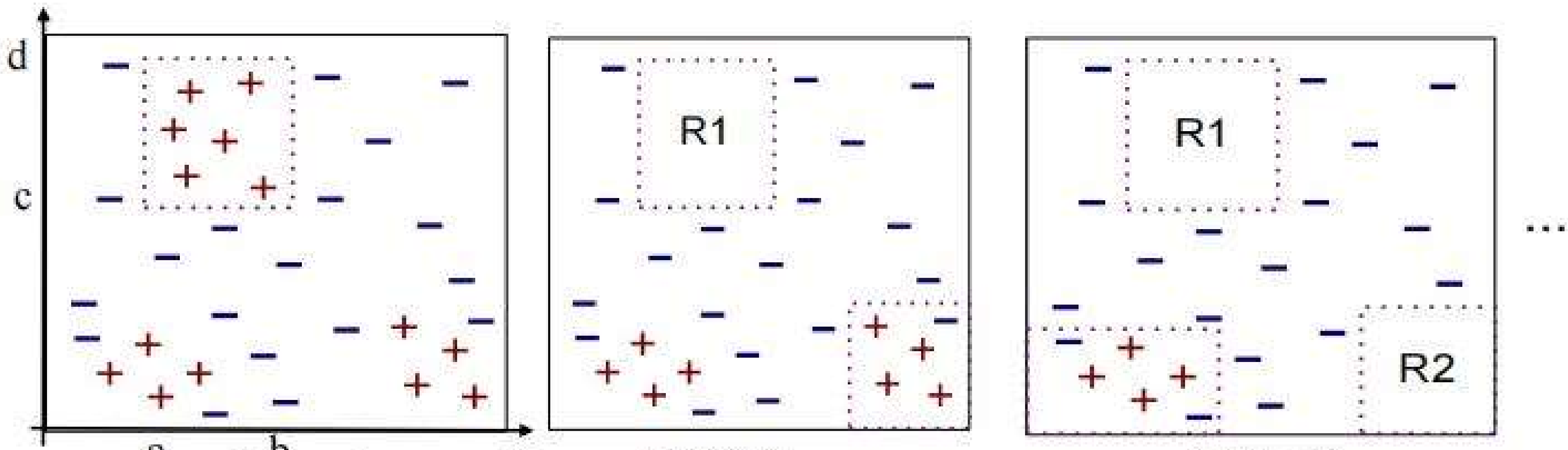
Aquatic Creature = No was pruned



- Rules are mutually exclusive and exhaustive (cover all training cases)
- Rule set contains as much information as the tree
- Rules can be simplified (similar to pruning of the tree)
- Example: C4.5rules

Direct Methods of Rule Generation

- Extract rules directly from the data
- Sequential Covering (Example: try to cover class +)



(ii) Step 1 (iii) Step 2 (iv) Step 3

R1: $a>x>b \wedge c>y>d \longrightarrow class +$

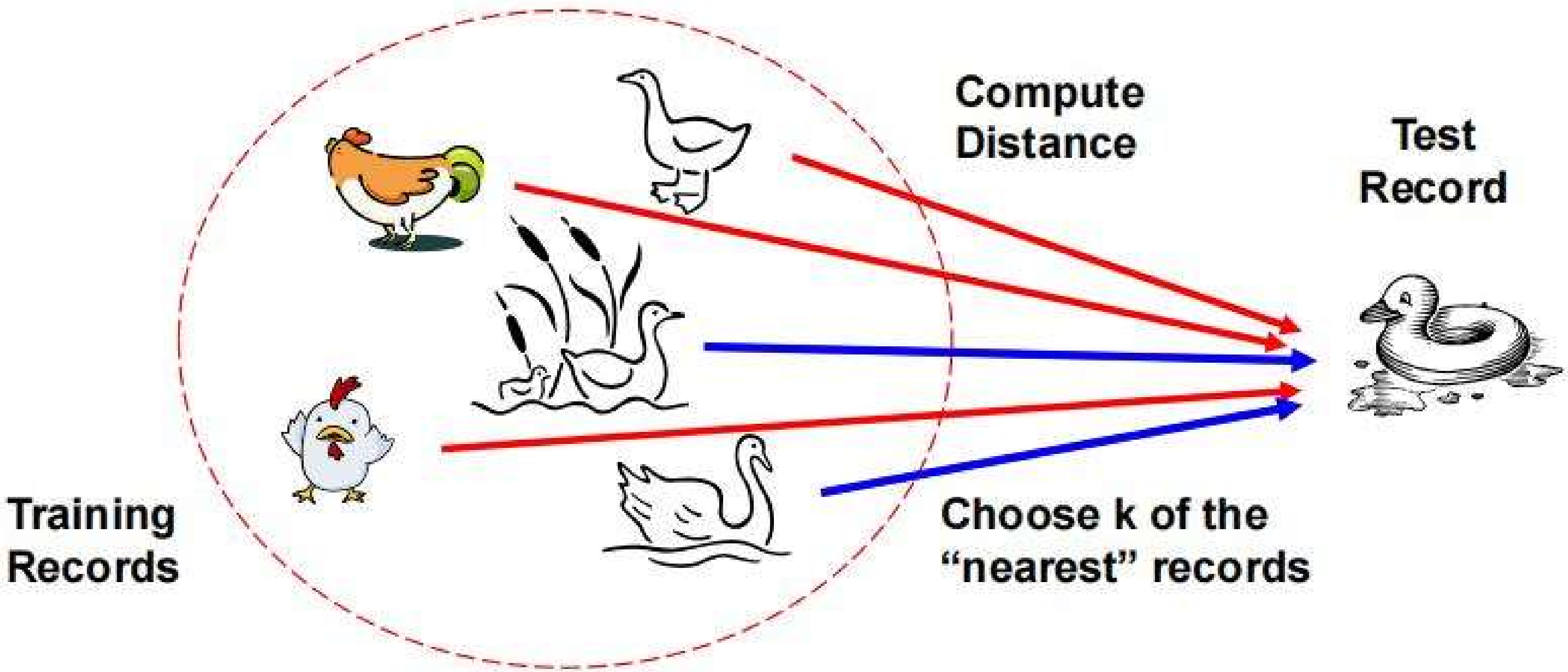
Advantages of Rule-Based Classifiers

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

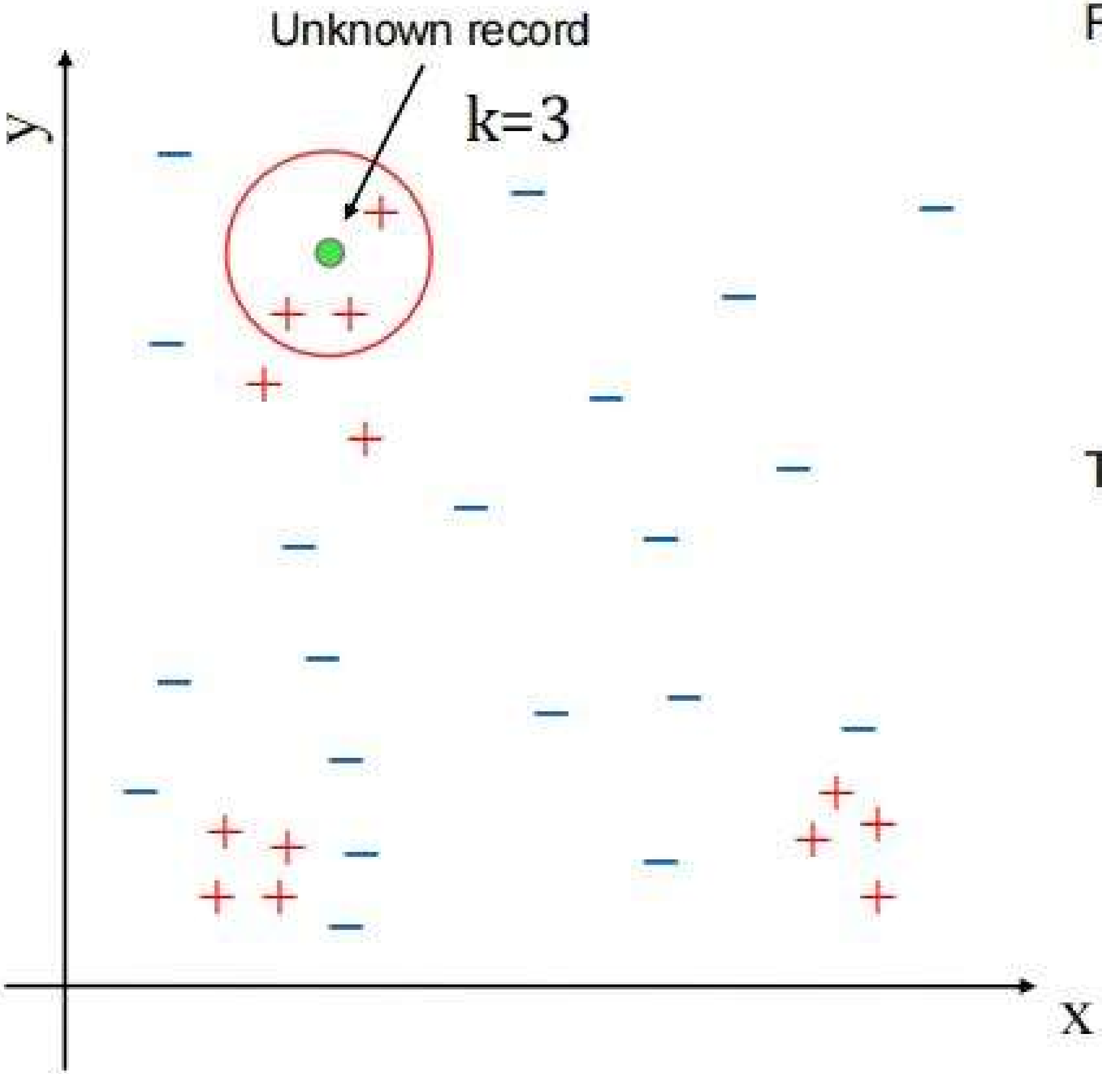
Nearest Neighbor Classifier

Nearest Neighbor Classifiers

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



Nearest-Neighbor Classifiers



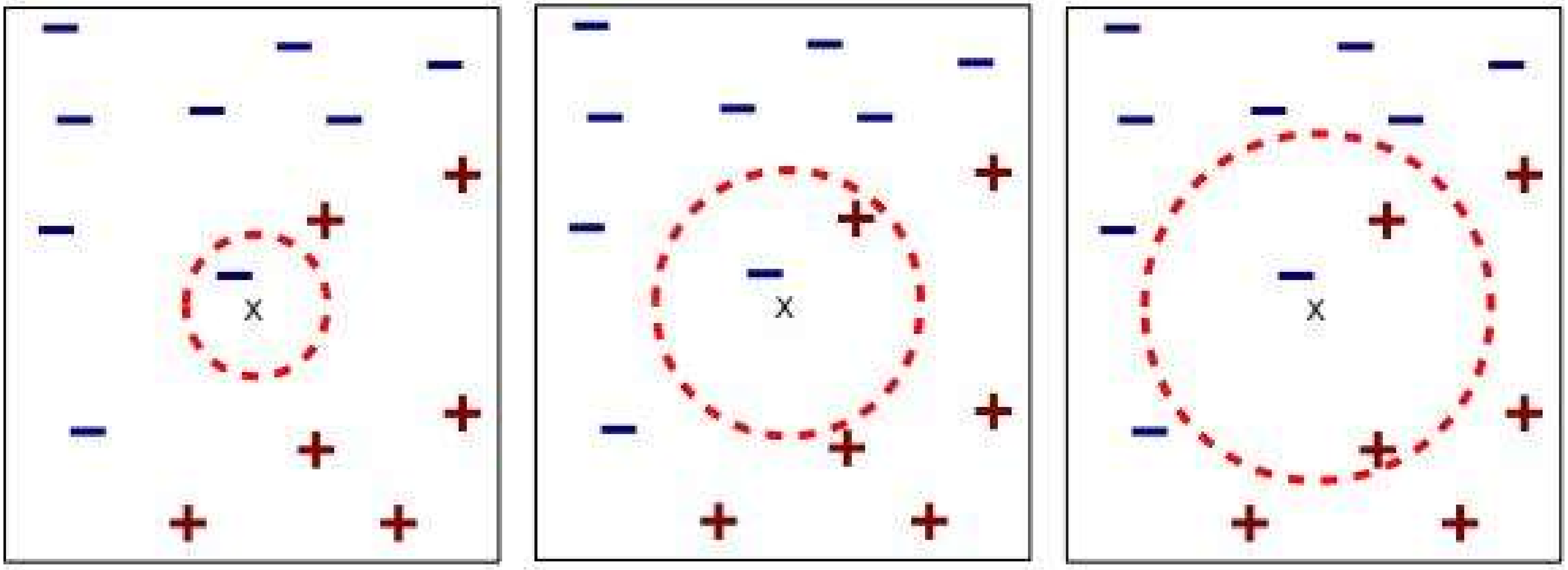
Requires three things

- 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

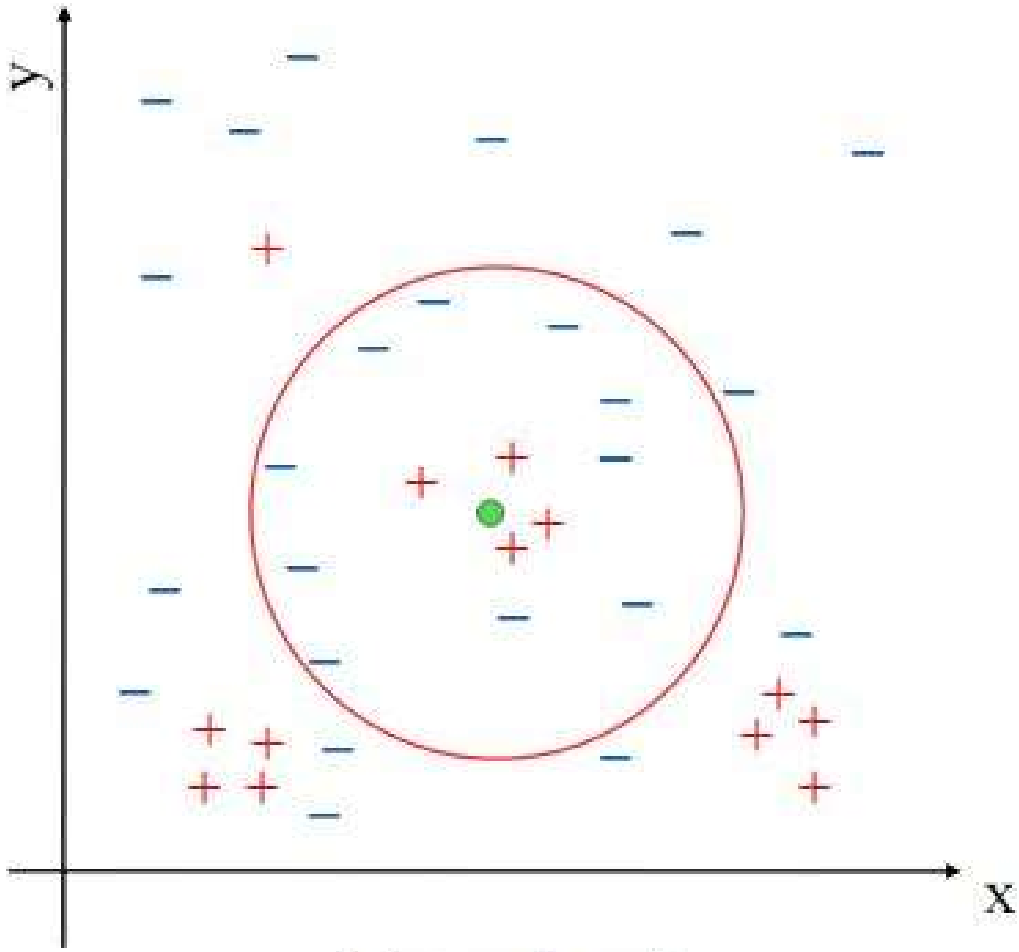
- Compute distance between two points:
  - Euclidean distance

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

- 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 (e.g., 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



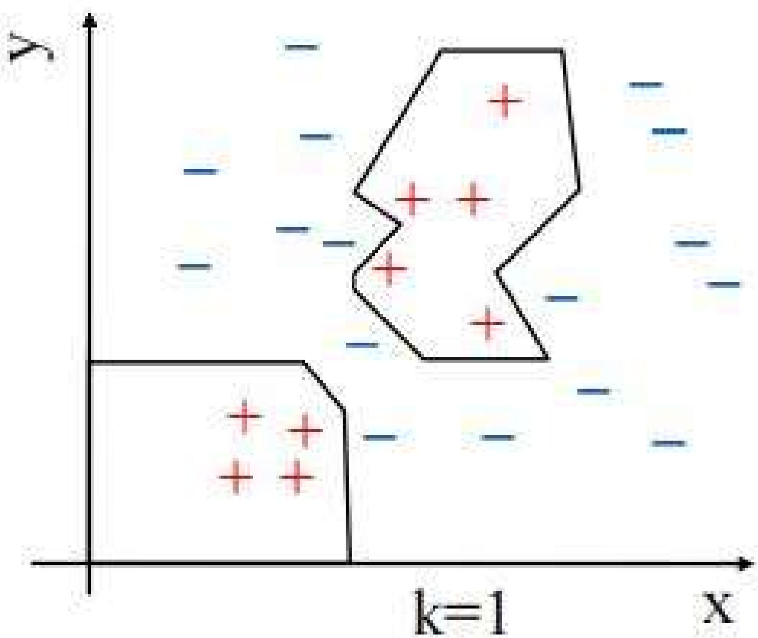
k is too large!

Nearest neighbor Classification…

k-NN classifiers are lazy learners

- —It does not build models explicitly (unlike eager learners such as decision trees)
- —Needs to store all the training data
- —Classifying unknown records are relatively expensive (find the k-nearest neighbors)

Advantage: Can create non-linear decision boundaries



Naive Bayes Classifier

Naive Bayes classifier

- Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class.
- Bayesian classification is based on Bayes' Theorem.
- It is based on simplifying assumpations that the attribute values are *conditionally independent,*
- A naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable.

Naive Bayes classifier

- For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. *A naive Bayes classifier considers all these features to contribute independently to the probability that this fruit is an apple*, whether or not they're in fact related to each other or to the existence of the other features.

- This reduces significantly computation cost since calculating each one of the $P(a_i|v_j)$ requires only a frequency count over the tuples in the training data with class value equal to v_j.

Conditional Probability

- For example, suppose you go out for lunch at the same place and time every Friday and you are served lunch within 15 minutes with probability 0.9. However, given that you notice that the restaurant is exceptionally busy, the probability of being served lunch within 15 minutes may reduce to 0.7. This is the conditional probability of being served lunch within 15 minutes given that the restaurant is exceptionally busy.
- The usual notation for "event A occurs given that event B has occurred" is "A | B" (A given B). The symbol | is a vertical line and does not imply division.
- P(A | B) denotes the probability that event A will occur given that event B has occurred already.

Conditional Probability

- A rule that can be used to determine a conditional probability from unconditional probabilities is:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

- where:
- P(A | B) = the (conditional) probability that event A will occur given that event B has occurred already.
- P(A∩B) = the (unconditional) probability that event A and event B both occur.
- P(B) = the (unconditional) probability that event B occurs.

Bayes Theorem : Basics

- Let **X** be a data sample : class label is unknown
- Let H be a *hypothesis* that X belongs to a specified class C
- For classification problems, we want to determine P(H|**X**), the probability that the hypothesis holds given the observed data sample **X**
- P(H) (*prior probability*), the initial probability
  - E.g., **X** will buy computer, regardless of age, income or any other information, for that matter.
- P(H|X) (*posteriori probability*), the probability of observing the sample **X**, given that the hypothesis holds
  - Suppose that *H* is the hypothesis that our customer will buy a computer.
  - Then P(H|X) reflects the probability that customer ***X*** will buy a computer given that we know the customer's age and income.

Bayesian Theorem

- Given data **X**, *posteriori probability of a hypothesis* H, P(H|**X**), follows the Bayes theorem

$$P(H\,|\,\mathbf{X}) = \frac{P(\mathbf{X}\,|\,H)P(H)}{P(\mathbf{X})}$$

- *P*(*X*|*H*) is the posterior probability of *X* conditioned on *H*. That is, it is the probability that a customer, *X*, is 35 years old and earns $40,000, given that we know the customer will buy a computer.

- Predicts **X** belongs to Ci if the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|X)$ for all the *k* classes.
- Practical difficulty: require initial knowledge of many probabilities.

Towards Naïve Bayesian Classifier

- Let D be a training set of tuples and their associated class labels, and each tuple is represented by an n- dimensional attribute vector $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, showing n measurements made on the tuple from n attributes.
- Suppose there are m classes $C_1, C_2, \ldots, C_m$.
- Classification is to derive the maximum posteriori, i.e., the maximal $P(C_i|\mathbf{X})$
- This can be derived from Bayes' theorem

$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

- Since P(X) is constant for all classes, only

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

needs to be maximized

Bayesian Classifier – Basic Equation

Class Prior Probability

Descriptor Posterior Probability

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

Class Posterior Probability

Descriptor Prior Probability

Example 1 -Naïve Bayesian Classifier

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

Test Data:
X = (age <=30,
Income = medium,
Student = yes
Credit_rating = Fair)
Class : ??

Training Dataset

age	income	student	credit_rating	_comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…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

Example 1 -Naïve Bayesian Classifier

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357

- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4

- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019
$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

Therefore, X belongs to class ("buys_computer = yes")

Example 2 -Naïve Bayesian Classifier

Outlook	Temp	Humidity	Windy
sunny	hot	high	FALSE
sunny	hot	high	TRUE
overcast	hot	high	FALSE
rainy	mild	high	FALSE
rainy	cool	normal	FALSE
rainy	cool	Normal	TRUE
overcast	cool	Normal	TRUE
sunny	mild	High	FALSE
sunny	cool	Normal	FALSE
rainy	mild	Normal	FALSE
sunny	mild	normal	TRUE
overcast	mild	High	TRUE
overcast	hot	Normal	FALSE
rainy	mild	high	TRUE

Play?
No
No
Yes
Yes
Yes
No
Yes
No
Yes
Yes
Yes
Yes
Yes
No

P(yes) = 9/14
P(no) = 5/14

Outlook	Temp.	Humidity	Windy	Play
sunny	cool	high	true	?

Example 2 -Naïve Bayesian Classifier

Bayesian Classifier – Probabilities for the weather data

Frequency Tables

Outlook	No	Yes
Sunny	3	2
Overcast	0	4
Rainy	2	3

Temp.	No	Yes
Hot	2	2
Mild	2	4
Cool	1	3

Humidity	No	Yes
High	4	3
Normal	1	6

Windy	No	Yes
False	2	6
True	3	3

Outlook	No	Yes
Sunny	3/5	2/9
Overcast	0/5	4/9
Rainy	2/5	3/9

Temp.	No	Yes
Hot	2/5	2/9
Mild	2/5	4/9
Cool	1/5	3/9

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

Windy	No	Yes
False	2/5	6/9
True	3/5	3/9

Example 2 -Naïve Bayesian Classifier

Bayesian Classifier – Predicting a new day

	Outlook	Temp.	Humidity	Windy	Play
X →	sunny	cool	high	true	NO

Class?

P(**yes**|**X**) = p(sunny|yes) x p(cool|yes) x p(high|yes) x p(true|yes) x **p(yes)**

= 2/9 x 3/9 x 3/9 x 3/9 x **9/14** = 0.0053 => 0.0053/(0.0053+0.0206) = 0.205

P(**no**|**X**) = p(sunny|no) x p(cool|no) x p(high|no) x p(true|no) x **p(no)**

= 3/5 x 1/5 x 4/5 x 3/5 x **5/14** = 0.0206=0.0206/(0.0053+0.0206) = 0.795

Outlook	No	Yes
Sunny	3/5	2/9
Overcast	0/5	4/9
Rainy	2/5	3/9

Temp.	No	Yes
Hot	2/5	2/9
Mild	2/5	4/9
Cool	1/5	3/9

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

Windy	No	Yes
False	2/5	6/9
True	3/5	3/9

Metrics for Performance Evaluation of Classifier : Confusion Matrix

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes (Positive)	Class=No (Negative)
	Class=Yes (Positive)	a	b
	Class=No (Negative)	c	d

- The entries in the confusion matrix have the following meaning :
  - a is the number of **correct** predictions that an instance is **positive,**
  - *b* is the number of **incorrect** of predictions that an instance **negative**,
  - *c* is the number of **incorrect** predictions that an instance is **positive**, and
  - d is the number of **correct** predictions that an instance is **negative.**

Metrics for Performance Evaluation of Classifier

- The *accuracy* (*AC*)- is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

- Consider a 2-class problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - ➢ Accuracy is misleading because model does not detect any class 1 example

Metrics for Performance Evaluation of Classifier

- The *recall* or *true positive rate* (*TP*) is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$\text{TP} = \frac{a}{a+b} = \frac{TP}{TP+FN}$$

- The *false positive rate* (*FP*) is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation:

$$\text{FP} = \frac{c}{c+d} = \frac{FP}{FP+TN}$$

Metrics for Performance Evaluation of Classifier

- The *true negative rate* (*TN*) is defined as the proportion of negatives cases that were classified correctly, as calculated using the equation:

$$\mathrm{TN} = \frac{d}{d+c} = \frac{TN}{TN+FP}$$

- The *false negative rate* (*FN*) is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation:

$$\mathrm{FN} = \frac{b}{b+a} = \frac{FN}{FN+TP}$$

Metrics for Performance Evaluation of Classifier

- *The precision* (*P*) is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$\mathrm{P} = \frac{a}{c+a} = \frac{TP}{FP+TP}$$

Confusion Matrix : Example

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Confusion Matrix : Example

- **Accuracy:** Overall, how often is the classifier correct?
  - (TP+TN)/total = (100+50)/165 = 0.91
- **Misclassification Rate:** Overall, how often is it wrong?
  - (FP+FN)/total = (10+5)/165 = 0.09
  - equivalent to 1 minus Accuracy
  - also known as "Error Rate"

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Confusion Matrix : Example

- **True Positive Rate:** When it's actually yes, how often does it predict yes?
  - TP/actual yes = 100/105 = 0.95
  - also known as "Sensitivity" or "Recall"
- **False Positive Rate:** When it's actually no, how often does it predict yes?
  - FP/actual no = 10/60 = 0.17
- **True Negative Rate:** When it's actually no, how often does it predict no?
  - TN/actual no = 50/60 = 0.83
  - equivalent to 1 minus False Positive Rate
  - also known as "Specificity"

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Confusion Matrix : Example

- **Precision:** When it predicts yes, how often is it correct?
  - TP/predicted yes = 100/110 = 0.91
- **Prevalence:** How often does the yes condition actually occur in our sample?
  - actual yes/total = 105/165 = 0.64

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Confusion Matrix : Example

n = 100	Actual: No	Actual: Yes	
Predicted: No	TN: 65	FP: 3	68
Predicted: Yes	FN: 8	TP: 24	32
	73	27	