Data Mining

Classification – Alternative Techniques (1)

Topics

- **Rule-Based Classifier**
- Nearest Neighbor Classifier
- Naive Bayes Classifier

Rule-Based Classifier

- Classify records by using a collection of "if…then…" rules
- Rule: $(Condition) \rightarrow y$
  - —Condition is a conjunctions of attributes called LHS, antecedent or condition
  - — y is the class label called RHS or consequent
- Examples of classification rules for an animal dataset:
  - $(Blood\ Type = Warm) \wedge (Lay\ Eggs = Yes) \rightarrow Birds$
  - $(Taxable\ Income < 50K) \wedge (Refund = Yes) \rightarrow Evade = No$

Using a Rule-Based Classifier

A rule R **covers** an instance x if the attributes of the instance satisfy the condition of the rule. Such a rule can be used for classification.

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

Rule base

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: $grizzly\ bear \rightarrow Mammal$

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 (R3 is selected below -> Amphibians)
  - —If none of the rules fired, it is assigned to the default class

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

R2: (Give Birth = yes) ∧ (Blood Type = warm) → Mammals

R3: (Live in Water = sometimes) → Amphibians

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

R5: (Give Birth = no) → Amphibians

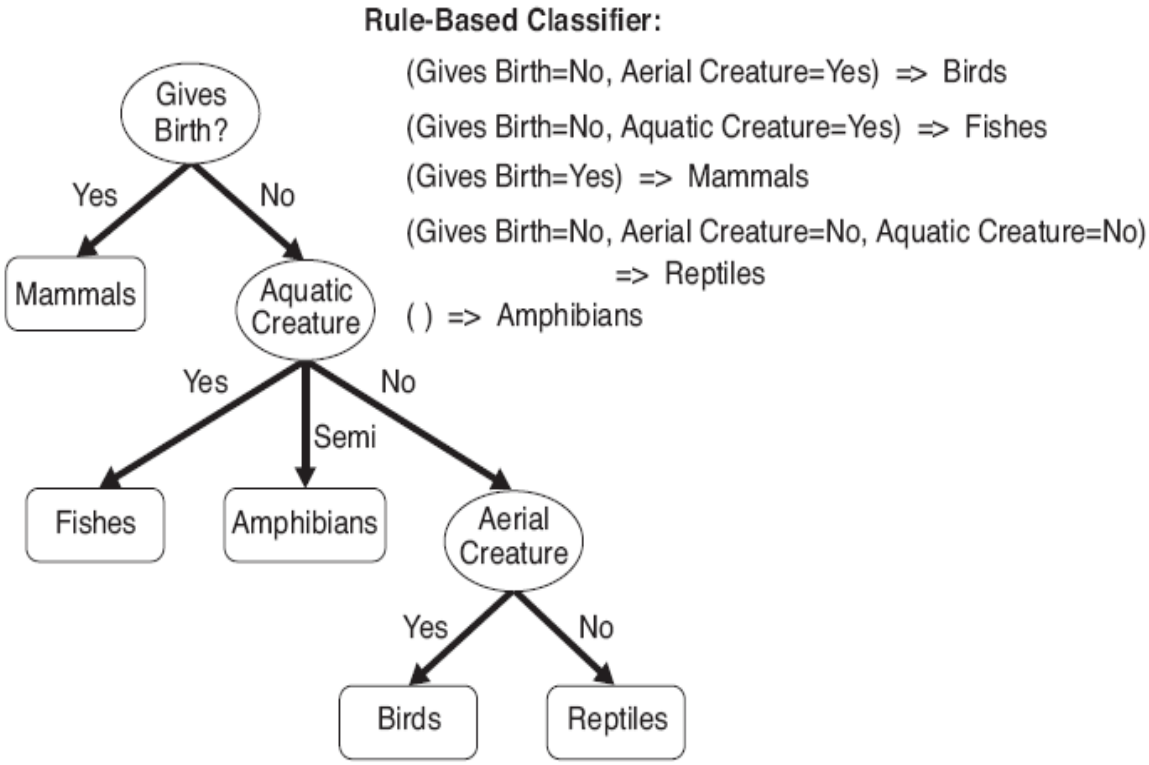
Rule base

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

R3, 4 and 5 cover the observation

- Alternative: (weighted) voting by all matching rules (-> Amphibians

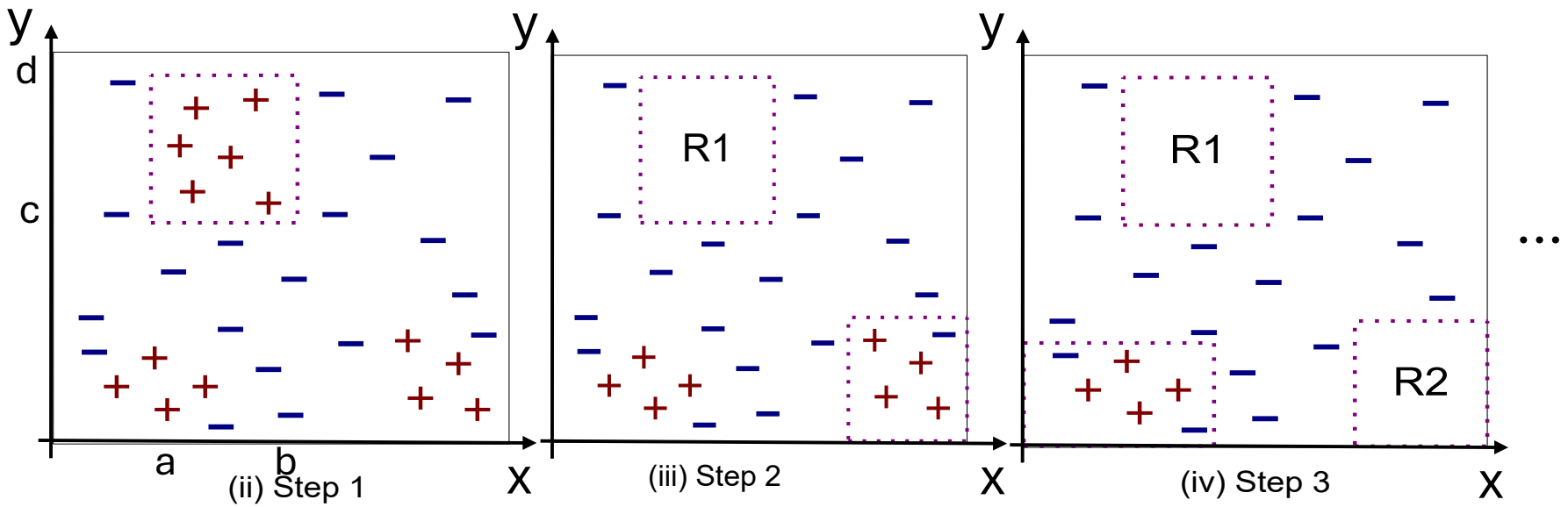
Rules From Decision Trees



- Rules are created by reading the decisions in tree branches from the root to a final node.
- 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 +)



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

Advantages of Rule-Based Classifiers

As expressive as decision trees

Easy to interpret

Easy to generate

Can classify new instances rapidly

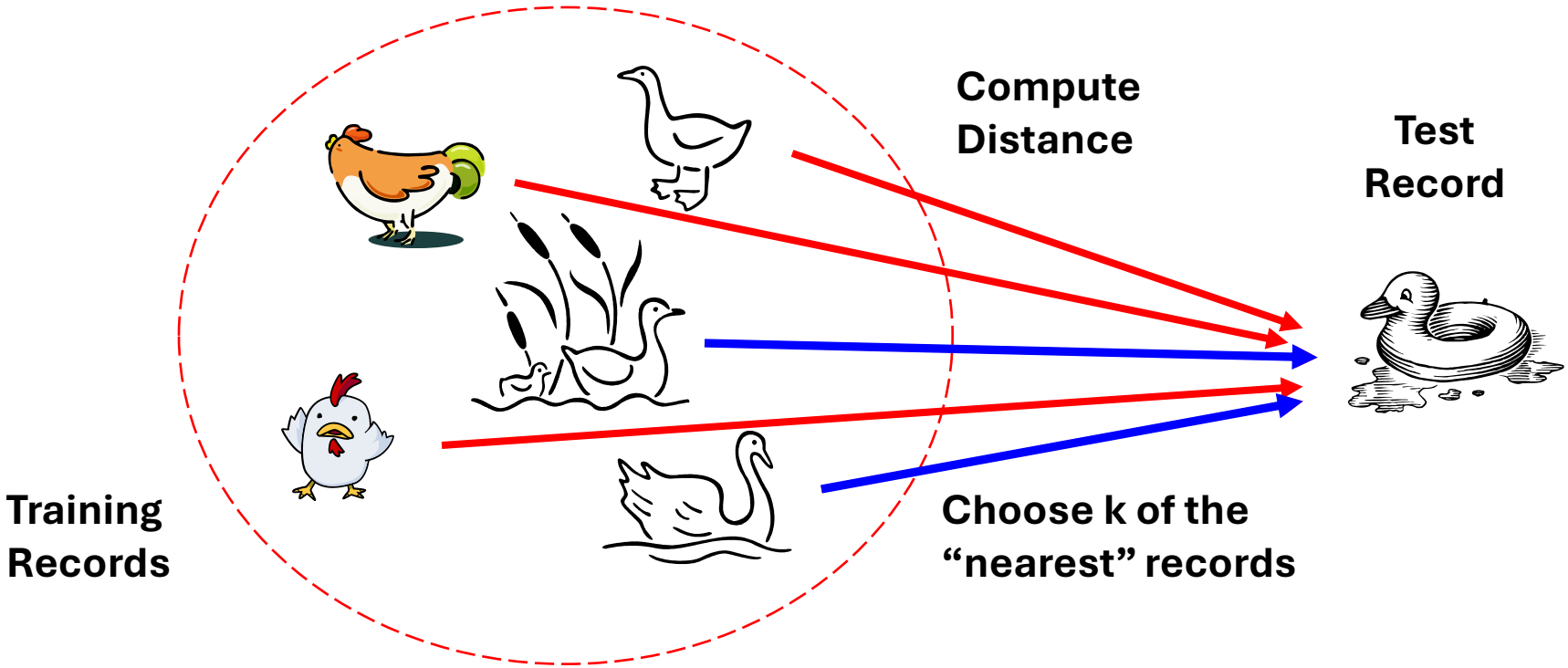
Performance comparable to decision trees

Topics

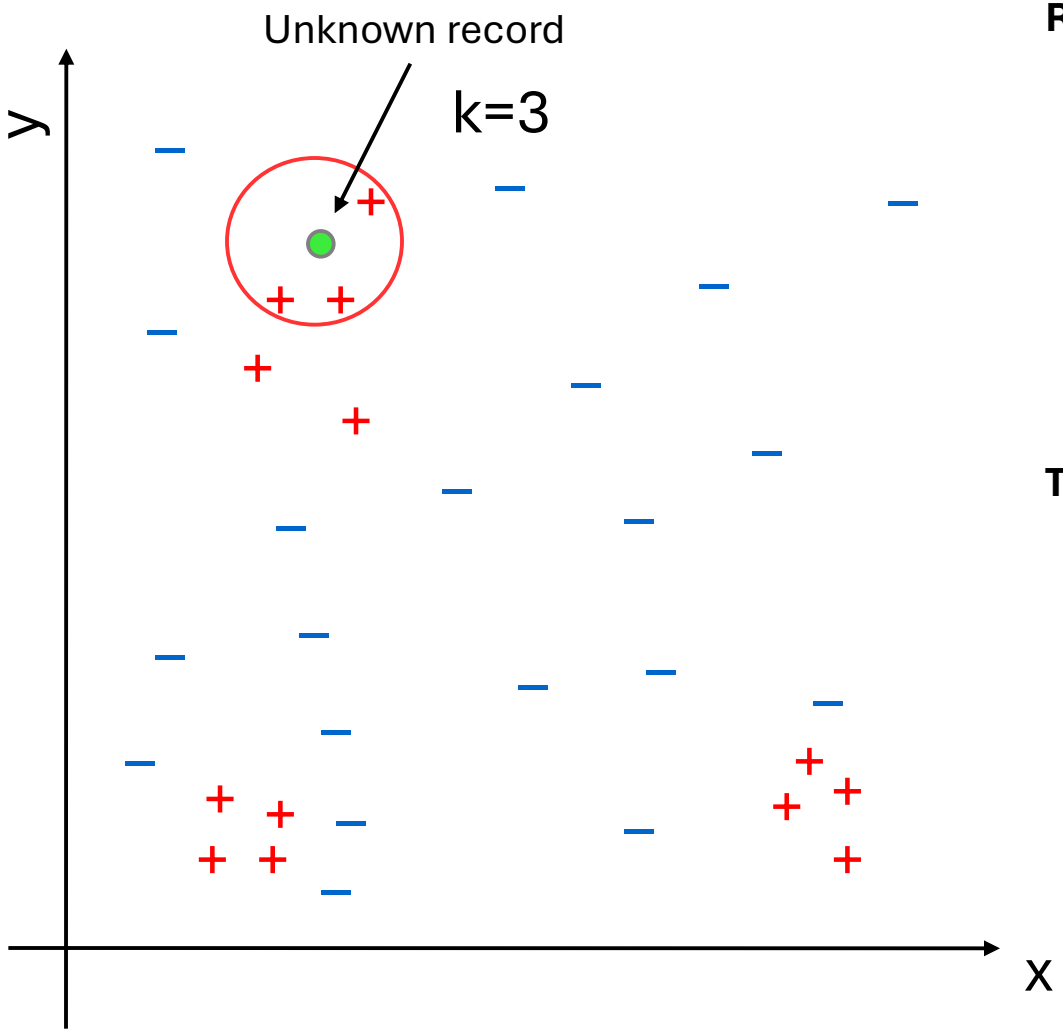
- Rule-Based Classifier
- **Nearest Neighbor Classifier**
- Naive Bayes Classifier

Nearest-Neighbor Classifiers

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



Nearest-Neighbor Classifiers



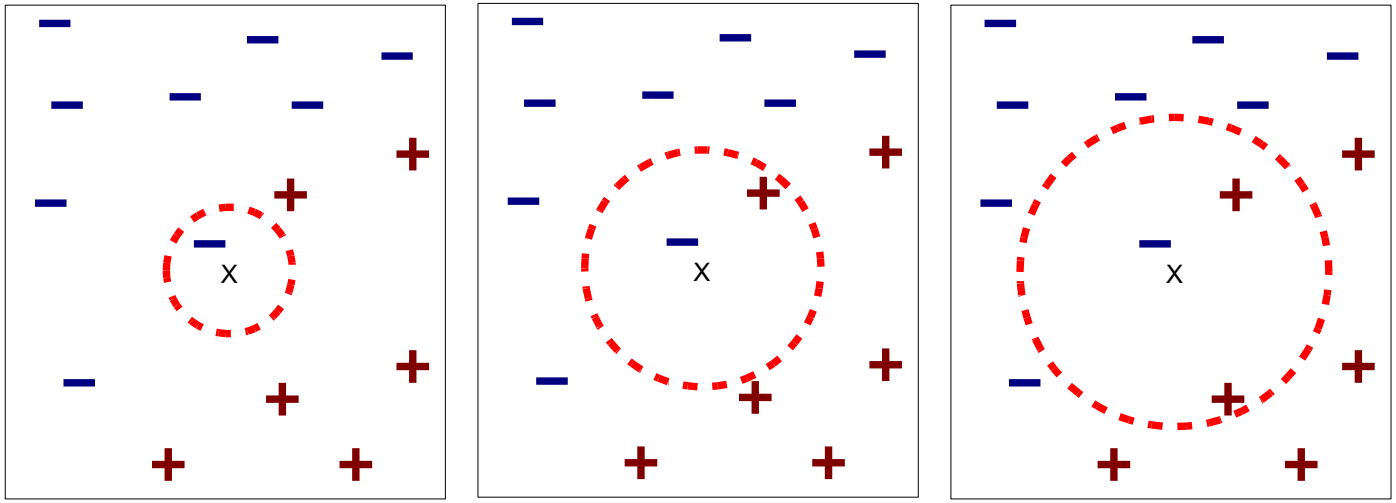
Requires three things

- The set of stored records.
- Distance Metric to compute the 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 an 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 with the k smallest distances to x.
- k is a hyperparameter.
- Odd numbers are preferable for k.

Distance Computation for Nearest-Neighbor Classification

- Compute distance between two points:
  - —Typically uses Euclidean distance

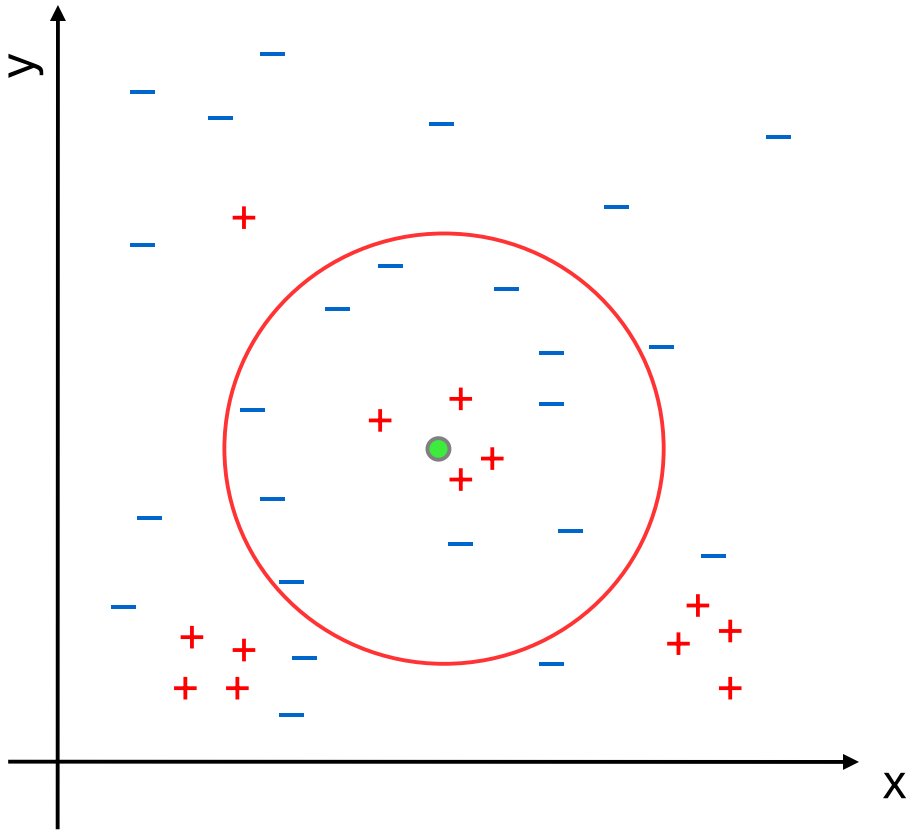
$$d(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{\sum_i (p_i - q_i)^2}$$

Note: This means that the data needs to be **scaled**!

- Determine the class from nearest neighbor list. Options
  a. Take the majority vote of class labels among the k-nearest neighbors.
  b. Weigh the vote according to distance (e.g., weight factor $w = 1/d^2$).

Choosing k

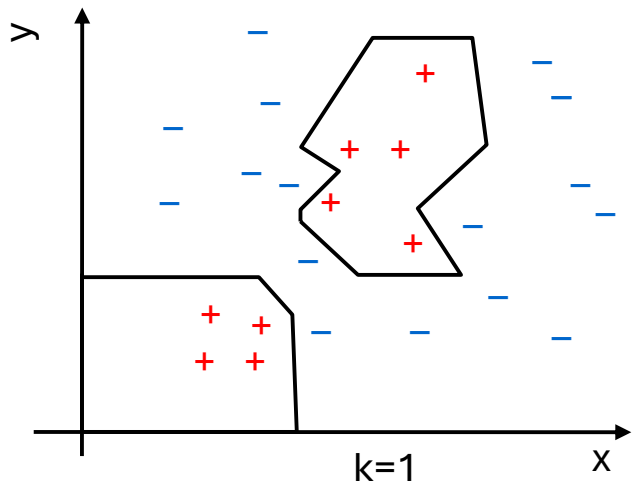
- If k is too small, sensitive to noise points
- If k is too large, neighborhood may include points from other classes



k is too large!

Advantages and Disadvantages

Advantage: Can create arbitrary non-linear decision boundaries.



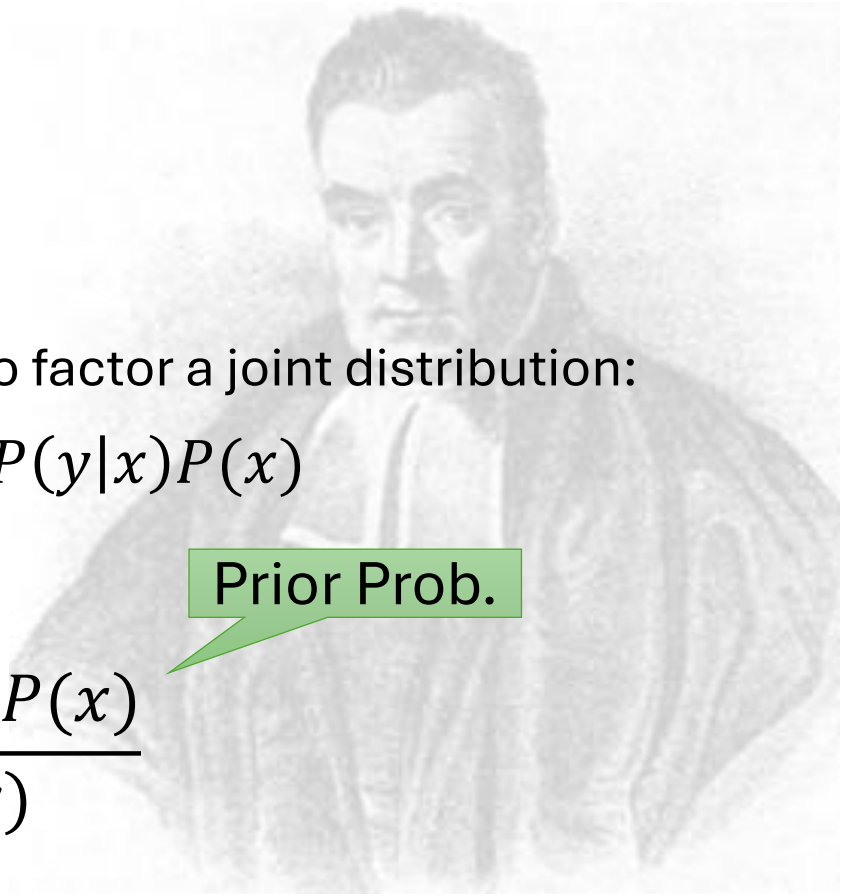
Disadvantages: k-NN classifiers are lazy learners

- 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). Space partitioning data structures like k-d trees can help.

Topics

- Rule-Based Classifier
- Nearest Neighbor Classifier
- **Naive Bayes Classifier**

Bayes' Rule



- The product rule gives us two ways to factor a joint distribution:

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x)$$

- Therefore,

Posterior Prob. Prior Prob.

$$P(y|x) = \frac{P(x|y)\ P(x)}{P(y)}$$

- Why is this useful?
  - —Can get diagnostic probability $P(\text{cavity} \mid \text{toothache})$ from causal probability $P(\text{toothache} \mid \text{cavity})$
  - —We can update our beliefs based on evidence.
  - —Important tool for probabilistic inference .

Example of Bayes Theorem

- A doctor knows that meningitis causes a stiff neck 50% of the time

$$P(s|m) = .5$$

- The probability of any patient having meningitis is

$$P(m) = 1/50{,}000 = \mathbf{0.00002}$$

- The probability of any patient having stiff neck is

$$P(s) = 1/20 = 0.05$$

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

$$P(m \mid s) = \frac{P(s \mid m)\, P(m)}{P(s)} = \frac{.5 \times 0.00002}{0.05} = \mathbf{0.0002}$$

Increases the probability by x10!

Bayesian Classification Rule

- Consider each attribute and class label as a random variable X_i taking the value x_i and Y taking y.
- Classification problem: Given a record with attributes $(x_1, x_2, \ldots, x_n)$ predict class y.
- This can be done by finding the most likely class that has the largest

$$\text{argmax}_y \, P(y \,|x_1, x_2, \ldots, x_n)$$

- This classification rule is guaranteed **optimal** for the accuracy measure!

Bayesian Classifiers

- Compute the posterior probability $P(y \mid x_1, x_2 \ \ldots, x_n)$ for all values of y using the Bayes theorem

$$\underset{y}{\text{argmax}}\, P(y \mid x_1, x_2, \ldots, x_n) = \text{argmax}_C \frac{P(x_1, x_2, \ldots, x_n \mid y)\, P(y)}{P(x_1, x_2, \ldots, x_n)}$$

This is a constant! We don't need it for the max.

- This is equivalent to choosing value of C that maximizes

$$\underset{y}{\text{argmax}}\, P(x_1, x_2, \ldots, x_n \mid y)\, P(y)$$

- Estimating the probability distribution $\boldsymbol{P}(Y)$ is easy, but how do we estimate

$$\boldsymbol{P}(X_1, X_2, \ldots, X_n \mid Y)?$$

Unfortunately, the table for this probability distribution is very large and can only be estimated for a small number of attributes n.

Approximation of Bayesian Classifiers

- Decision trees use
$$\underset{y}{\operatorname{argmax}}\ P(y \mid \text{leafNodeMatching}(x_1, x_2, \ldots, x_n))$$

- Rule-based classifiers use
$$\underset{y}{\operatorname{argmax}}\ P(y \mid \text{rulesMatching}(x_1, x_2, \ldots, x_n))$$

- K-NN classifiers use
$$\underset{y}{\operatorname{argmax}}\ P(y \mid \text{neighborhood}(x_1, x_2, \ldots, x_n))$$

- ANN classifiers with a final Softmax layer use
$$\underset{y}{\operatorname{argmax}}\ P(y \mid \text{activationBeforeSoftmaxLayer}(x_1, x_2, \ldots, x_n))$$

Naïve Bayes Classifier

Approximates a Bayes Classifier by assuming independence among attributes X given the class. Now we can factor the probability distribution into the product of a few independent probabilities.

$$P(x_1, x_2, \dots, x_n \,|y) \;=\; P(x_1|\, y)\, P(x_2|\, y) \dots\; P(x_n|\, y) = \prod_i P(x_i \mid y)$$

We can estimate $P(x_i \mid y)$ for all x_i and y.

A new observation is classified as y such that:

$$\underset{y}{\operatorname{argmax}}\; P(y) \prod_i P(x_i \,|y)$$

How to Estimate Probabilities from Data? Nominal Features

- Use the maximum likelihood estimate for probabilities.
- Class: $P(y) = \frac{N_y}{N}$

  e.g., $P(Y = \text{No}) = 7/10$,
  $P(Y = \text{Yes}) = 3/10$
- For discrete attributes:

$$P(x_i|y) = \frac{N_{x_i \wedge y}}{N_y}$$

where $N_{x_i \wedge y}$ is number of instances having attribute x_i and belongs to class y.

e.g.,
$P(\text{Status=Married} \mid y = \text{No}) = 4/7$
$P(\text{Refund=Yes} \mid y = \text{Yes}) = 0$

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**

How to Estimate Probabilities from Data? Continuous Features

Several options:

- Discretize the range into bins
  - one binary variable per bin (one-hot encoding).
  - violates the independence assumption.

- Two-way split: $(x_i < v)$ or $(x_i > v)$
  - Encode with one binary variable.

- Probability density estimation.
  - Assume the attribute follows a normal distribution.
  - Use data to estimate the parameters of the distribution (e.g., mean and standard deviation).
  - Once the probability distribution is known, we can use it to estimate the conditional probability $P(x_i \mid y)$.
  - Most implementations will do this automatically. This is called a Gaussian Naïve Bayes Classifier.

Example of Naïve Bayes Classifier

Given a Test Record what is the most likely class?

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

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

0s are an issue!

P(No|***x***) = P(***x***|No)P(No) > P(Yes|***x***) = P(***x***|Yes)P(Yes)

Predicted Class is No

Naïve Bayes Classifier: Dealing With Low Counts

Probability estimation:

Original: $P(x_i \mid y) = \frac{N_{x_i \wedge y}}{N_y}$

Issue: If one of the conditional probabilities is zero, then the entire expression becomes zero.

Laplace: $P(x_i \mid y) = \frac{N_{x_i \wedge y} + 1}{N_y + c}$

c: number of classes
m: parameter

m-estimate: $P(x_i \mid y) = \frac{N_{x_i \wedge y} + mP(y)}{N_y + m}$

Summary of Naïve Bayes Classifiers

- **Robust to outliers** and isolated noise points since it is not based on distances.
- **Can handle missing value** during prediction: Ignore the attribute during probability estimate calculations.
- **Robust to irrelevant attributes**: Features are estimated independently. Irrelevant features will produce a likelihood that is a uniform distribution given the class.
- **Independence assumption** may not hold for some attributes
  - Typically, the classifiers still work well when the assumption is slightly violated.
  - You can remove highly correlated attributes.
  - Use other techniques such as Bayesian Belief Networks (BBN) that explicitly model dependence.