Naïve Bayes Classifiers

Anna Monreale
Computer Science Department

Introduction to Data Mining, 2nd Edition
Chapter 5.3

Motivation

- Relationship between attributes and class lables may not be deterministic
- Reasons:
  - Noise in the data
  - Confounding factors affecting the classification and not in the data
- Bayesian Classifier exploit the Bayes Theorem that combines prior knowledge on the class labels with knowledge derivable from data

Bayes Classifier

- A probabilistic framework for solving classification problems.
- Let P be a probability function that assigns a number between 0 and 1 to events.
- X = x an events is happening
- P(X = x) is the probability that events X = x.
- Joint Probability P(X = x, Y = y)
- Conditional Probability P(Y = y | X = x)
- Relationship: P(X,Y) = P(Y|X) P(X) = P(X|Y) P(Y)
- Bayes Theorem: P(Y|X) = P(X|Y)P(Y) / P(X)
- Another Useful Property: P(X =x) = P(X=x, Y=0) + P(X=x, Y=1)

Bayes Theorem

- Consider a football game. Team 0 wins 65% of the time, Team 1 the remaining 35%. Among the game won by Team 1, 75% of them are won playing at home. Among the games won by Team 0, 30% of them are won at Team 1's field.
- **If Team 1 is hosting the next match, which team will most likely win?**
- Team 0 wins: $P(Y = 0) = 0.65$
- Team 1 wins: $P(Y = 1) = 0.35$
- Team 1 hosted the match won by Team 1: $P(X = 1|Y = 1) = 0.75$
- Team 1 hosted the match won by Team 0: $P(X = 1|Y = 0) = 0.30$
- Objective $P(Y = 1|X = 1)$

Bayes Theorem

- $P(Y = 1|X = 1) = P(X = 1|Y = 1)P(Y = 1) / P(X = 1) =$

  $= 0.75 \times 0.35 / (P(X = 1, Y = 1) + P(X = 1, Y = 0))$

  $= 0.75 \times 0.35 / (P(X = 1|Y = 1)P(Y=1) + P(X = 1|Y = 0)P(Y=0))$

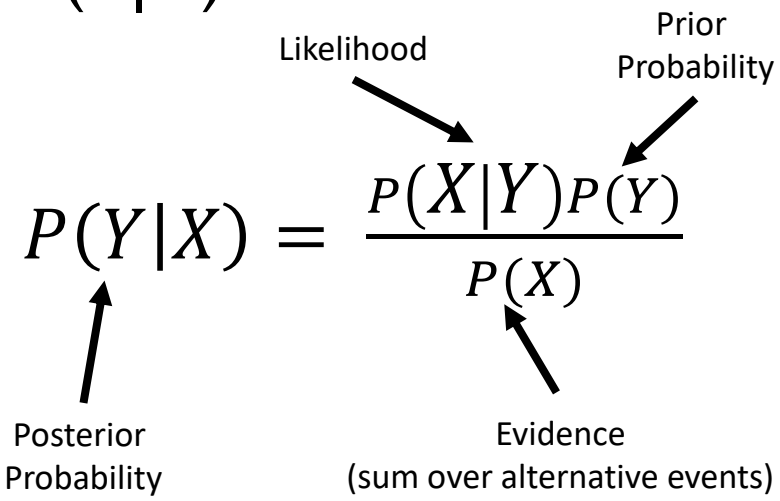
  $= 0.75 \times 0.35 / (0.75 \times 0.35 + 0.30 \times 0.65)$

  $= 0.5738$

- Therefore Team 1 has a better chance to win the match

Bayes Theorem for Classification

- X denotes the attribute sets, $X = \{X_1, X_2, \ldots X_d\}$
- Y denotes the class variable
- We treat the relationship probabilistically using P(Y|X)



Bayes Theorem for Classification

- Learn the posterior P(Y | X) for every combination of X and Y.
- By knowing these probabilities, a test record X’ can be classified by finding the class Y’ that maximizes the posterior probability P(Y’|X’).
- This is equivalent of choosing the value of Y’ that maximizes P(X’|Y’)P(Y’).
- How to estimate it?

Naïve Bayes Classifier

- It estimates the class-conditional probability by **assuming that the attributes are conditionally independent** given the class label y.
- The conditional independence is stated as:

$$P(X|Y = y) = \prod_{i=1}^{d} P(X_i|Y = y)$$

where each attribute set $X = \{X_1, X_2, \ldots X_d\}$

Conditional Independence

- Given three variables Y, X_1, X_2 we can say that Y is independent from X_1 given X_2 if the following condition holds:

$$P(Y \mid X_1, X_2) = P(Y|X_2)$$

- With the conditional independence assumption, instead of computing the class-conditional probability for every combination of *X* we only need to estimate the conditional probability of each X_i given *Y*.

- Thus, to classify a record the naive Bayes classifier computes the posterior for each class *Y* and takes the maximum class as result

$$P(Y|X) = P(Y) \prod_{i=1}^{d} P(X_i|Y = y) \, / P(X)$$

How to estimate ?

How to Estimate Probability From Data

- Class $P(Y) = N_y / N$
- N_y number of records with outcome y
- N number of records
- Categorical attributes

  $$P(X = x \mid Y = y) = N_{xy} / N_y$$

- N_{xy} records with value x and outcome y
- P(Evade = Yes) = 3/10
- P(Marital Status = Single|Yes) = 2/3

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

How to Estimate Probability From Data

Continuous attributes

- Discretize the range into bins
  - Continuous vs nominal
  - Estimation: count records with class y and falling in the range
- 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(X|y)

How to Estimate Probability From Data

- Normal distribution

$$P(X_i = x_i \mid Y = y) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

- μ_{ij} can be estimated as the mean of X_i for the records that belongs to class y_j.
- Similarly, σ_{ij} as the standard deviation.
- P(Income = 120|No) = 0.0072
  - mean = 110
  - std dev = 54.54

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

M-estimate of Conditional Probability

- If one of the conditional probability is zero, then the entire expression becomes zero.
- For example, given X = {Refund = Yes, Divorced, Income = 120k}, if P(Divorced|No) is zero instead of 1/7, then
  - P(X|No) = 3/7 x 0 x 0.00072 = 0
  - P(X|Yes) = 0 x 1/3 x 10^{-9} = 0
- M-estimate $P(X|Y) = \frac{N_{xy}+mp}{N_y+m}$ (if $P(X|Y) = \frac{N_{xy}+1}{N_y+|Y|}$ is Laplacian estimation)
- m is a parameter, p is a user-specified parameter (e.g. probability of observing x_i among records with class y_j.
- In the example with m = 3 and p = 1/m = 1/3 (i.e., Laplacian estimation) we have

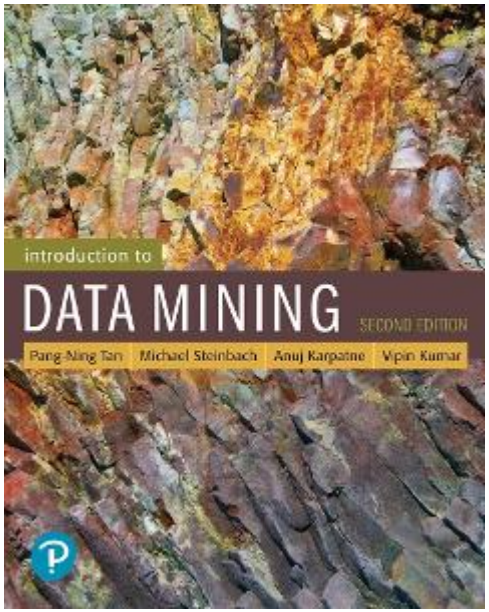
$$P(\text{Married}|\text{Yes}) = (0+3\text{x}1/3)/(3+3) = 1/6$$

Naïve Bayes Classifier

- 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, not treated in this course)

References

- Bayesian Classifiers. Chapter 5.3. Introduction to Data Mining.



EXERCISE - NBC

Play-tennis example. estimating $P(x_i|C)$

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

P(p) = 9/14
P(n) = 5/14

outlook	
P(sunny|p) =	**P(sunny|n) =**
P(overcast|p) =	**P(overcast|n) =**
P(rain|p) =	**P(rain|n) =**
temperature	
P(hot|p) =	**P(hot|n) =**
P(mild|p) =	**P(mild|n) =**
P(cool|p) =	**P(cool|n) =**
humidity	
P(high|p) =	**P(high|n) =**
P(normal|p) =	**P(normal|n) =**
windy	
P(true|p) =	**P(true|n) =**
P(false|p) =	**P(false|n) =**

Play-tennis example. estimating $P(x_i|C)$

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

P(p) = 9/14
P(n) = 5/14

outlook	
P(sunny|p) = 2/9	**P(sunny|n) = 3/5**
P(overcast|p) = 4/9	**P(overcast|n) = 0**
P(rain|p) = 3/9	**P(rain|n) = 2/5**
temperature	
P(hot|p) = 2/9	**P(hot|n) = 2/5**
P(mild|p) = 4/9	**P(mild|n) = 2/5**
P(cool|p) = 3/9	**P(cool|n) = 1/5**
humidity	
P(high|p) = 3/9	**P(high|n) = 4/5**
P(normal|p) = 6/9	**P(normal|n) = 1/5**
windy	
P(true|p) = 3/9	**P(true|n) = 3/5**
P(false|p) = 6/9	**P(false|n) = 2/5**

Play-tennis example. estimating $P(x_i|C)$

P(p) = 9/14
P(n) = 5/14

Outlook	Temeprature	Humidity	Windy	Class
rain	hot	high	false	?

outlook	
P(sunny|p) = 2/9	P(sunny|n) = 3/5
P(overcast|p) = 4/9	P(overcast|n) = 0
P(rain|p) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|p) = 2/9	P(hot|n) = 2/5
P(mild|p) = 4/9	P(mild|n) = 2/5
P(cool|p) = 3/9	P(cool|n) = 1/5
humidity	
P(high|p) = 3/9	P(high|n) = 4/5
P(normal|p) = 6/9	P(normal|n) = 1/5
windy	
P(true|p) = 3/9	P(true|n) = 3/5
P(false|p) = 6/9	P(false|n) = 2/5

$P(X|p)\cdot P(p) =$

$P(X|n)\cdot P(n) =$

Play-tennis example. estimating $P(x_i|C)$

P(p) = 9/14
P(n) = 5/14

Outlook	Temeprature	Humidity	Windy	Class
rain	hot	high	false	N

outlook	
P(sunny|p) = 2/9	P(sunny|n) = 3/5
P(overcast|p) = 4/9	P(overcast|n) = 0
P(rain|p) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|p) = 2/9	P(hot|n) = 2/5
P(mild|p) = 4/9	P(mild|n) = 2/5
P(cool|p) = 3/9	P(cool|n) = 1/5
humidity	
P(high|p) = 3/9	P(high|n) = 4/5
P(normal|p) = 6/9	P(normal|n) = 1/5
windy	
P(true|p) = 3/9	P(true|n) = 3/5
P(false|p) = 6/9	P(false|n) = 2/5

P(X|p)·P(p) = P(rain|p)·P(hot|p)·P(high|p)·P(false|p)·P(p) = 3/9 · 2/9 · 3/9 · 6/9 · 9/14 = 0.010582

P(X|n)·P(n) = P(rain|n)·P(hot|n)·P(high|n)·P(false|n)·P(n) = 2/5 · 2/5 · 4/5 · 2/5 · 5/14 = 0.018286

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

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

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

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

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

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

=> Mammals

a) Naive Bayes (**3 points**)
Given the training set below, build a Naive Bayes classification model (i.e. the corresponding table of probabilities) using (i) the normal formula and (ii) using Laplace formula. What are the main effects of Laplace on the models?

A	B	class
no	green	N
no	red	Y
yes	green	N
no	red	N
no	red	Y
no	green	Y
yes	green	N

Answer:

Normal

	Y	N		Y	N
	3	4		0.43	0.57
	A | Y	A | N		A | Y	A | N
yes	0	2	yes	0.00	0.50
no	3	2	no	1.00	0.50
	B | Y	B | N		B | Y	B | N
green	1	3	green	0.33	0.75
red	2	1	red	0.67	0.25

Laplace

	Y	N		Y	N
	3	4		0.43	0.57
	A | Y	A | N		A | Y	A | N
yes	0	2	yes	0.20	0.50
no	3	2	no	0.80	0.50
	B | Y	B | N		B | Y	B | N
green	1	3	green	0.40	0.67
red	2	1	red	0.60	0.33

a) Naive Bayes (**3 points**)
Given the training set on the left, build a Naive Bayes classification model and apply it to the test set on the right.

SCORE	FIRST-TRY	FACULTY	class
good	no	science	Y
medium	yes	science	N
bad	yes	science	N
bad	yes	humanities	Y
good	no	humanities	N
good	no	science	Y
medium	no	humanities	Y

SCORE	FIRST-TRY	FACULTY	class
bad	no	humanities	
good	yes	science	
medium	yes	humanities	

Rule-based Classifier

- Classify records by using a collection of "if…then…" rules
- Rule: (*Condition*) $\rightarrow$ *y*
  - where
    - *Condition* is a conjunction of tests on attributes
    - *y* is the class label
  - 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)

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
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

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 => Bird

The rule R3 covers the grizzly bear => 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 the antecedent that also satisfy the 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 a 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 triggers rule R3, so it is classified as a mammal

A turtle triggers both R4 and R5

A dogfish shark triggers none of the rules

Characteristics of Rule Sets: Strategy 1

- **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

Characteristics of Rule Sets: Strategy 2

- **Rules are not mutually exclusive**
  - A record may trigger more than one rule
  - Solution?
    - Ordered rule set
    - Unordered rule set – use voting schemes

- **Rules are not 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 (typically majority 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	?

Rule Ordering Schemes

- Rule-based ordering
  - Individual rules are ranked based on their quality
- Class-based ordering
  - Rules that belong to the same class appear together

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

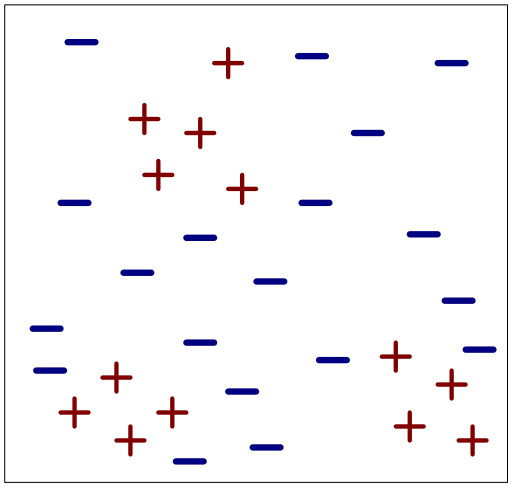
- Direct Method:
  - Extract rules directly from data
  - Examples: RIPPER, CN2, Holte's 1R

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

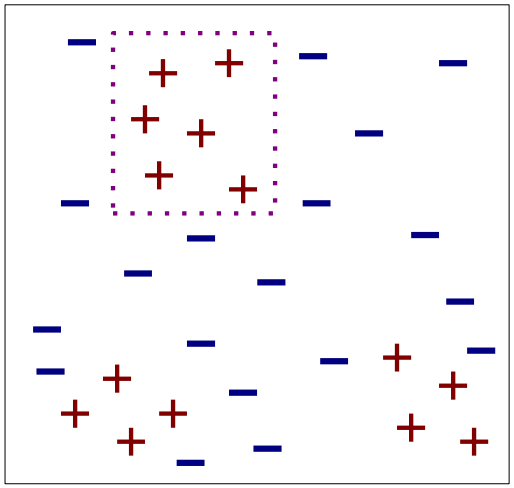
Direct Method: Sequential Covering

1. Start from an empty rule
2. For each class
    1. Grow a rule using the Learn-One-Rule function
    2. Remove training records covered by the rule
    3. Repeat Step (2) and (3) until stopping criterion is met

Example of Sequential Covering

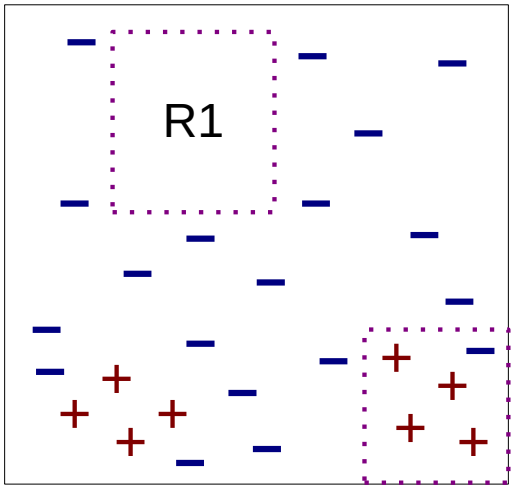


(i) Original Data

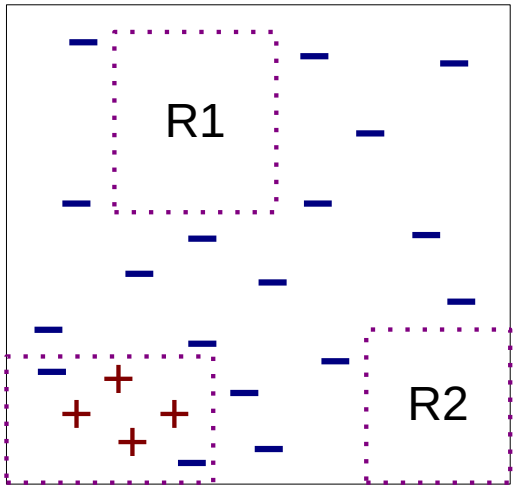


(ii) Step 1

Example of Sequential Covering…



(iii) Step 2



(iv) Step 3

Learn-One-Rule Function

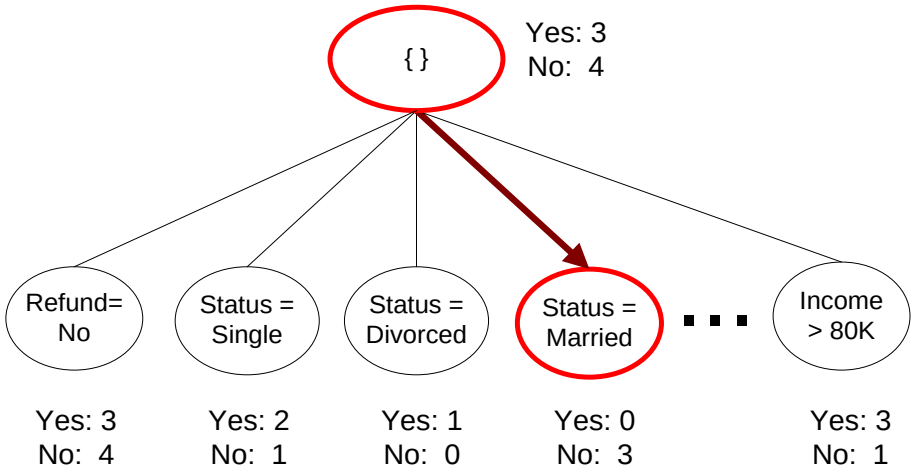
- The goal is to extract a classification rule covering many positive records and none (few) negative ones
- Finding optimal rule requires high computational time
- Greedy strategy by refining an initial rule

Greedy approach

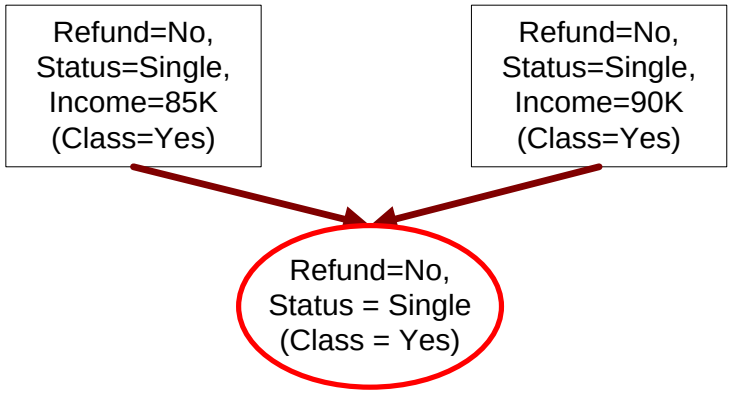
- The approach for growing rules is greedy
  - Based on some evaluation measure
- Rules are extracted one class per time
- The criterion for deciding the order of the class to consider depends on:
  - Class prevalence
  - Miss classification error for a given class

Rule Growing

- Two common strategies



(a) General-to-specific



(b) Specific-to-general

Rule Evaluation for growing rules

- Accuracy $= \frac{n_c}{n}$

- Laplace $= \frac{n_c + 1}{n + k}$

- M-estimate $= \frac{n_c + kp}{n + k}$

n : Number of instances covered by rule

n_c : Number of instances covered by rule with class c

k : Number of classes

p : Prior probability

Rule Evaluation for growing rules

- Foil's Information Gain
  - R0: {} => class (initial rule)
  - R1: {A} => class (rule after adding conjunct)
  - $$Gain(R_0, R_1) = p_1 \times [\, log_2 \left(\frac{p_1}{p_1 + n_1}\right) - log_2 \left(\frac{p_0}{p_0 + n_0}\right)]$$
  - p_0: number of positive instances covered by R0
    n_0: number of negative instances covered by R0
    p_1: number of positive instances covered by R1
    n_1: number of negative instances covered by R1

FOIL: First Order Inductive Learner – an early rule-based learning algorithm

Direct Method: RIPPER

- For 2-class problem, choose one of the classes as positive class, and the other as negative class
  - Learn rules for positive class
  - Negative class will be default class

- For multi-class problem
  - Order the classes according to increasing class prevalence (fraction of instances that belong to a particular class)
  - Learn the rule set for smallest class first, treat the rest as negative class
  - Repeat with next smallest class as positive class

Direct Method: RIPPER

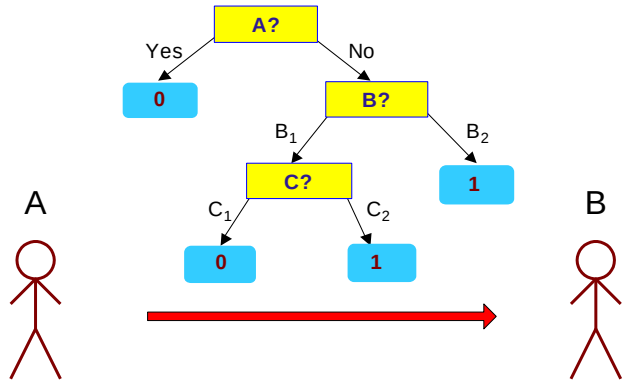
- Growing a rule:
  - Start from empty rule
  - Add conjuncts as long as they improve FOIL's information gain
  - Stop when rule no longer covers negative examples
  - Prune the rule immediately using incremental reduced error pruning
  - Measure for pruning: $v = (p-n)/(p+n)$
    - p: number of positive examples covered by the rule in the validation set
    - n: number of negative examples covered by the rule in the validation set
  - Pruning method: delete any final sequence of conditions that maximizes v

Direct Method: RIPPER

- Building a Rule Set:
  - Use sequential covering algorithm
    - Finds the best rule that covers the current set of positive examples
    - Eliminate both positive and negative examples covered by the rule
  - Each time a rule is added to the rule set, compute the new description length
    - Stop adding new rules when the new description length is *d* bits longer than the smallest description length obtained so far

Minimum Description Length (MDL)

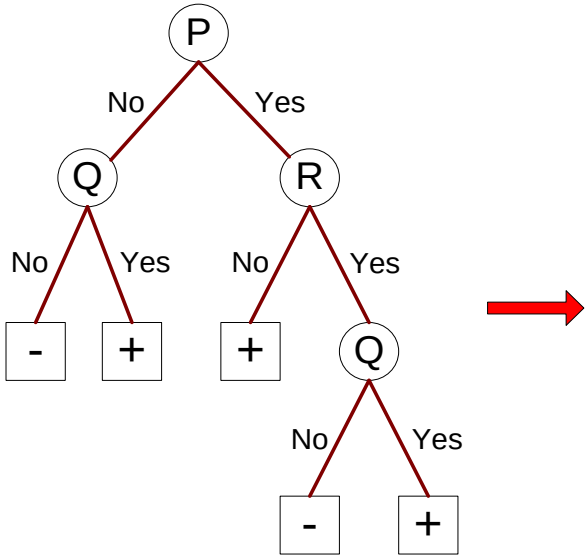
X	y
X_1	1
X_2	0
X_3	0
X_4	1
…	…
X_n	1



X	y
X_1	?
X_2	?
X_3	?
X_4	?
…	…
X_n	?

- Cost(Model,Data) = Cost(Data|Model) + αx Cost(Model)
  - Cost is the number of bits needed for encoding.
  - Search for the least costly model.
- Cost(Data|Model) encodes the misclassification errors.
- Cost(Model) uses node encoding (number of children) plus splitting condition encoding.

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 an alternative rule $r': A' \rightarrow y$ where A' is obtained by removing one of the conjuncts in A
  - Compare the *pessimistic error rate* for r against all r's
  - Prune if one of the alternative rules has lower pessimistic error rate
  - Remove duplicate rules
  - Repeat until we can no longer improve generalization error

Pessimistic Error Estimate

- **Pessimistic Error Estimate** of a rule set T with k rules:

$$err_{gen}(T) = err(T) + \Omega \times \frac{k}{N_{train}}$$

  - *err(T)*: error rate on all training records
  - Ω: trade-off hyper-parameter relative cost of adding a rule
  - *k*: number of rules nodes
  - N_{train}: total number of training records

Indirect Method: C4.5rules

- Instead of ordering the rules, order subsets of rules (class ordering)

- Each subset is a collection of rules with the same rule consequent (class)

- Compute description length of each subset
  - Description length = *L(error) + g L(model)*
  - *g* is a parameter that takes into account the presence of redundant attributes in a rule set (default value = 0.5)

Advantages of Rule-Based Classifiers

- Has characteristics quite similar to decision trees
  - As highly expressive as decision trees
  - Easy to interpret
  - Performance comparable to decision trees
  - Can handle redundant attributes

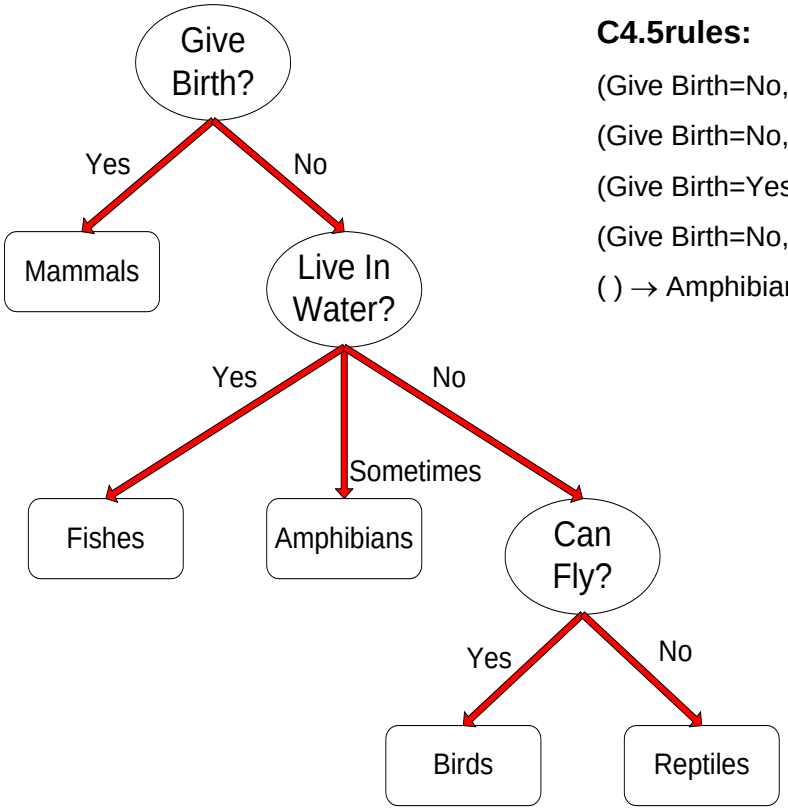
- Better suited for handling imbalanced classes

- Harder to handle missing values in the test set

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

(Have Legs=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

References

- Rule-Based Classifiers. Chapter 5.1. Introduction to Data Mining.