Similarity & Machine Learning

M. Anthony Kapolka III
CS 340 - Fall 2011 - Wilkes University

What is this?

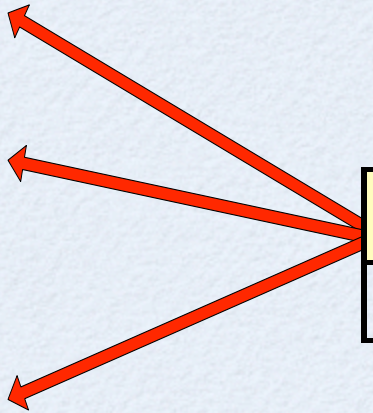
Instance Based Classifiers

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

INSTANCE-BASED CLASSIFIERS

Set of Stored Cases

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

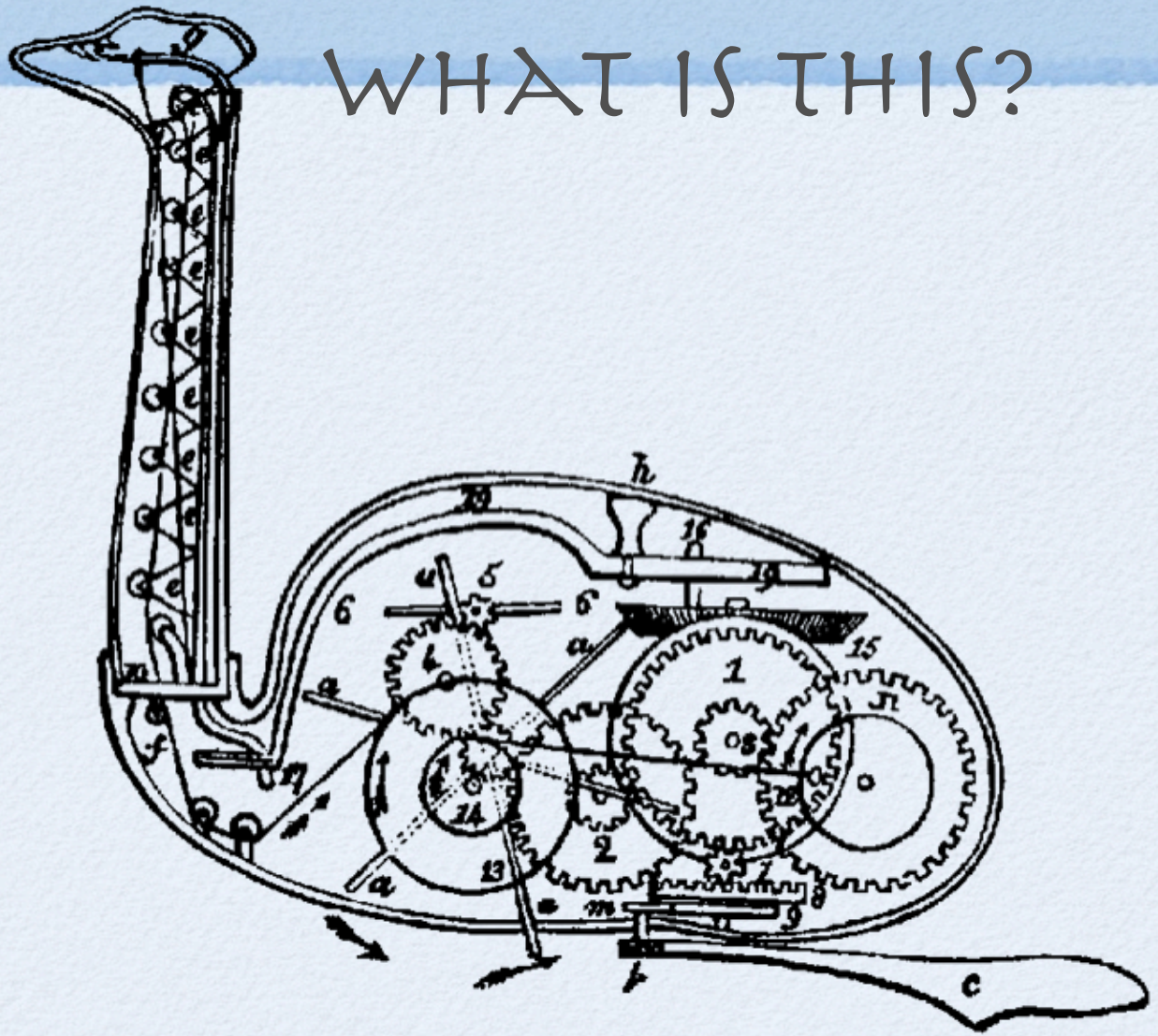


Unseen Case

Atr1	……	AtrN

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

What is this?



[http://www.personeel.unimaas.nl/h.schotel/Eendjes/Vaucanson-eendTransp.gif]

What is this?

http://www.gombergkites.com/

What is this?

http://www.crystal-fox.com

A DUCK?

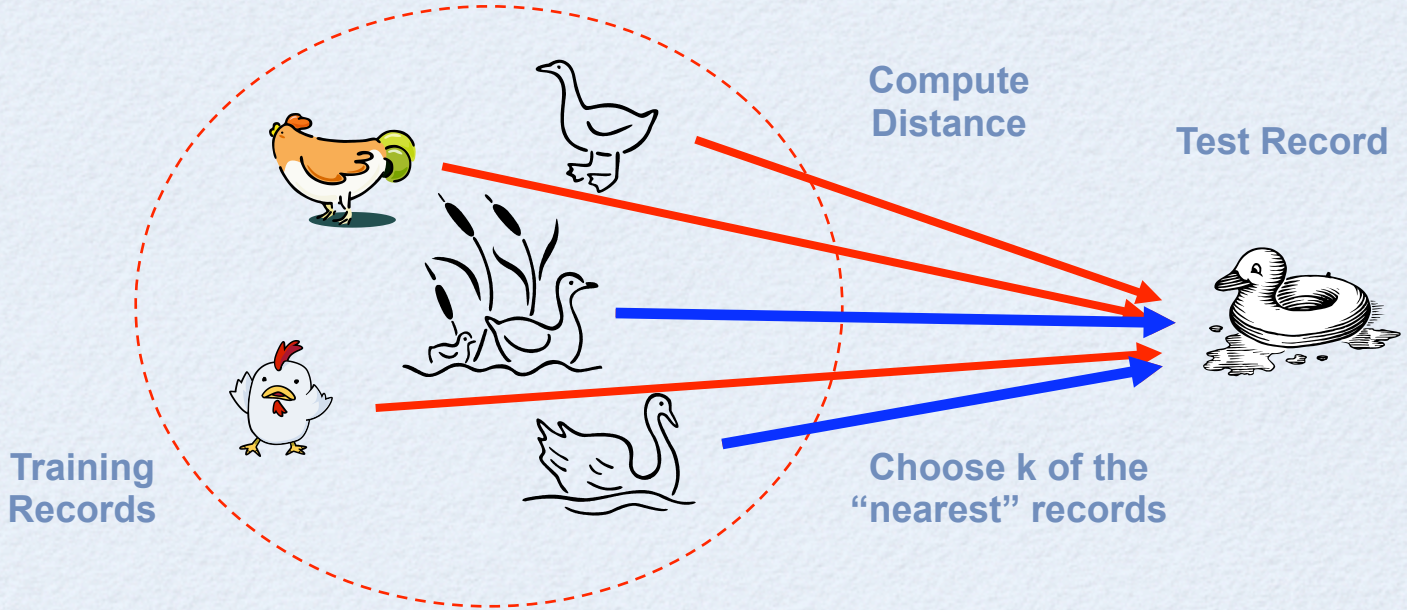
- if it walks like a duck
- and looks like a duck
- and quacks like a duck...

then it must be a duck!

Instance Based Classifiers

- Nearest neighbor
  - Uses k “closest” points (nearest neighbors) for performing classification

DISTANCE CLASSIFIER



What is this?

http://www.treorchy.net/

Similarity Metrics

Geometric Similarity

Define a distance function $\partial(a,b)$

∂ assigns pairs of points (a,b)
a non negative number such that three axioms hold:

Minimality	$\partial(a,b) \geq \partial(a,a) = 0$
Symmetry	$\partial(a,b) = \partial(b,a)$
Triangle Inequality	$\partial(a,b) + \partial(b,c) \geq \partial(a,c)$

Example measure

Euclidean distance

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

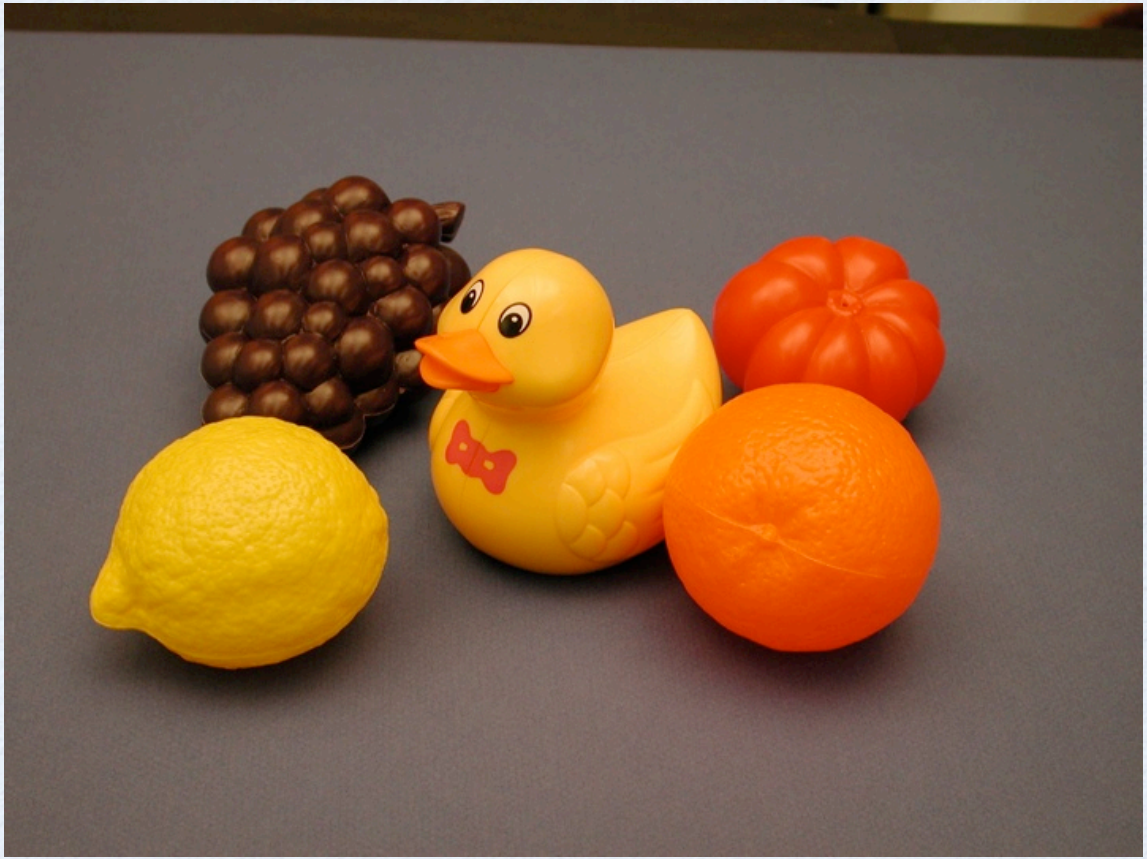
Similarity Metrics

- According to Amos Tversky, similarity between objects a and b is a function of
  - C, the common (shared) features (attributes)
  - A, the features unique in a
  - B, the features unique in b

$$S = \theta C - \alpha A - \beta B$$

where $\theta, \alpha, \beta \geq 0$

ONE OF THESE THINGS...



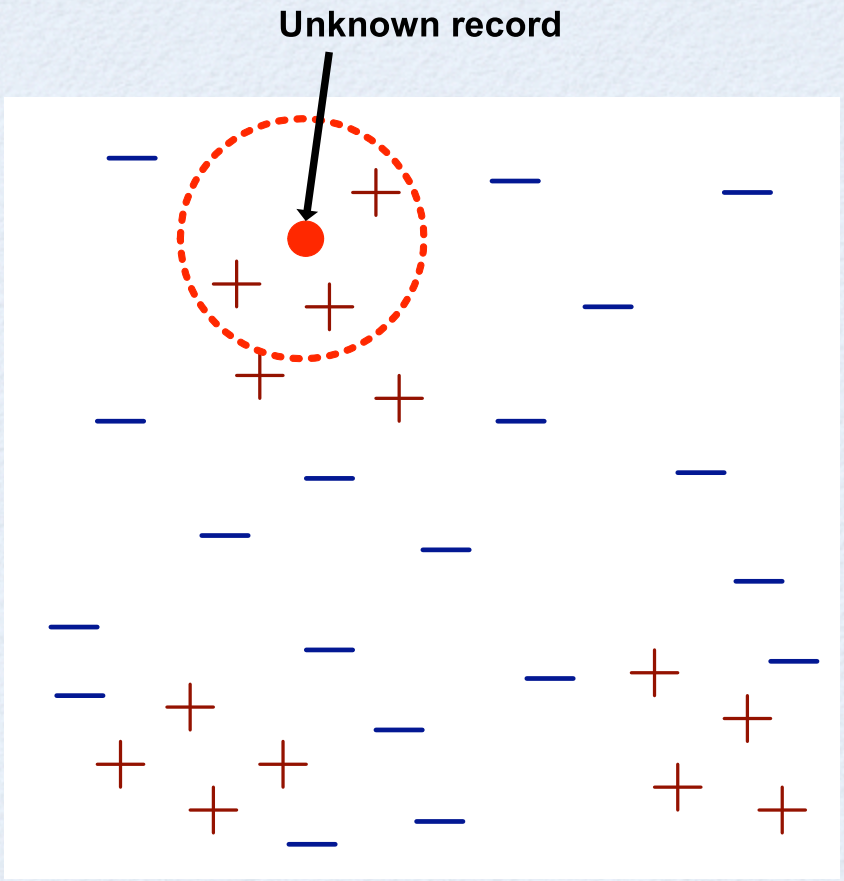
http://www.vision.ethz.ch/projects/cogvis/CogVis-images/

Issues to Resolve

- Categorization of Features
- Interpretation of Features
- Independence of Features

ONE OF THESE THINGS...

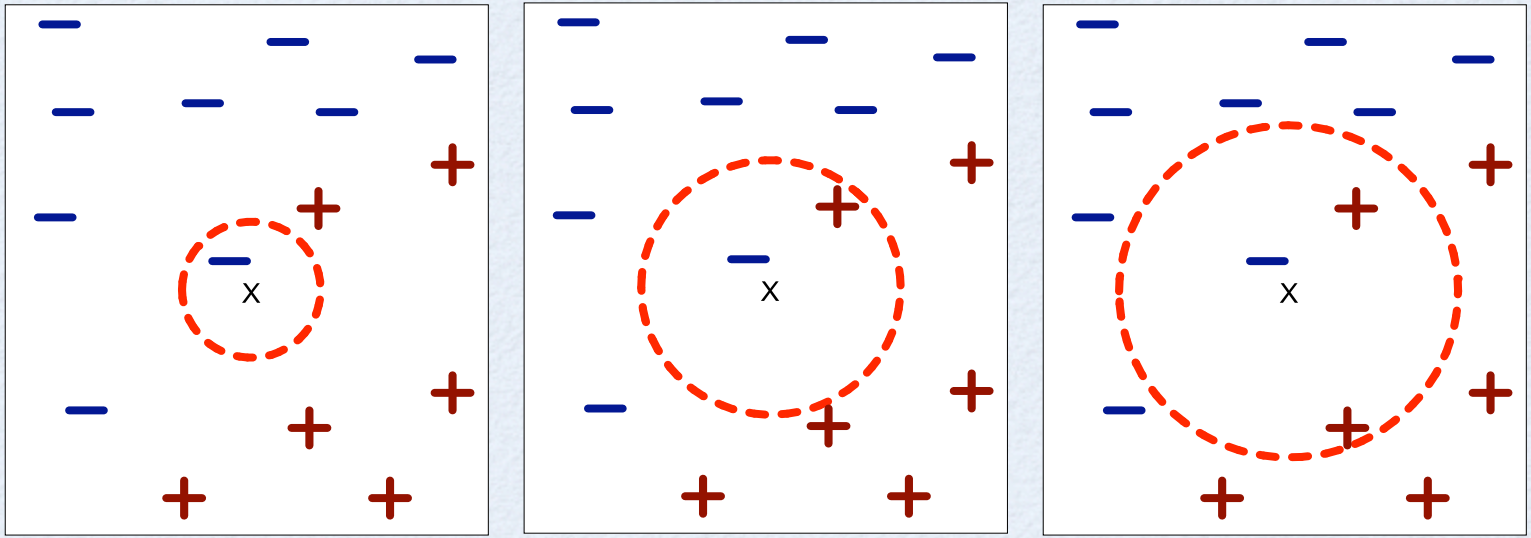
Nearest-Neighbor Classifiers



- 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

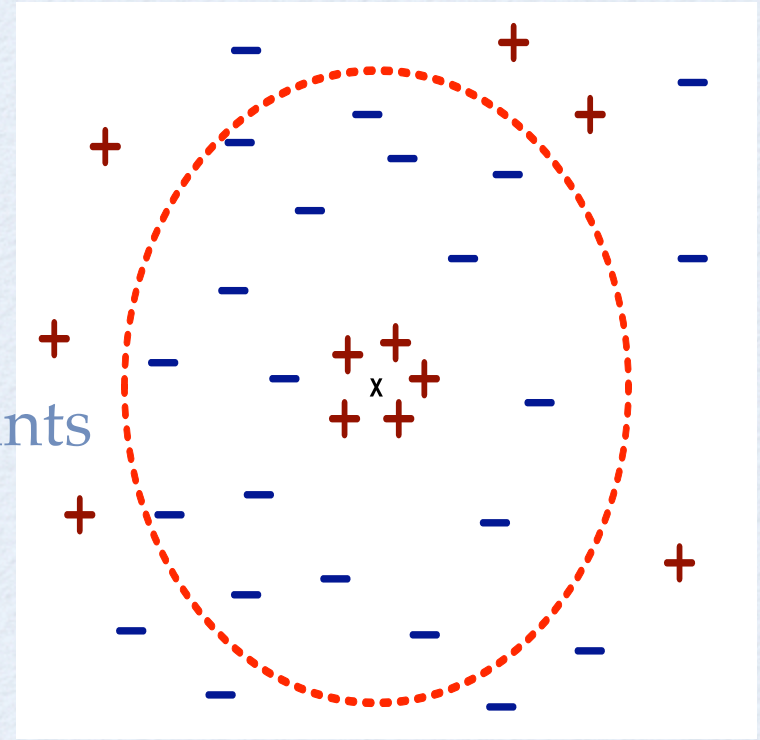


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

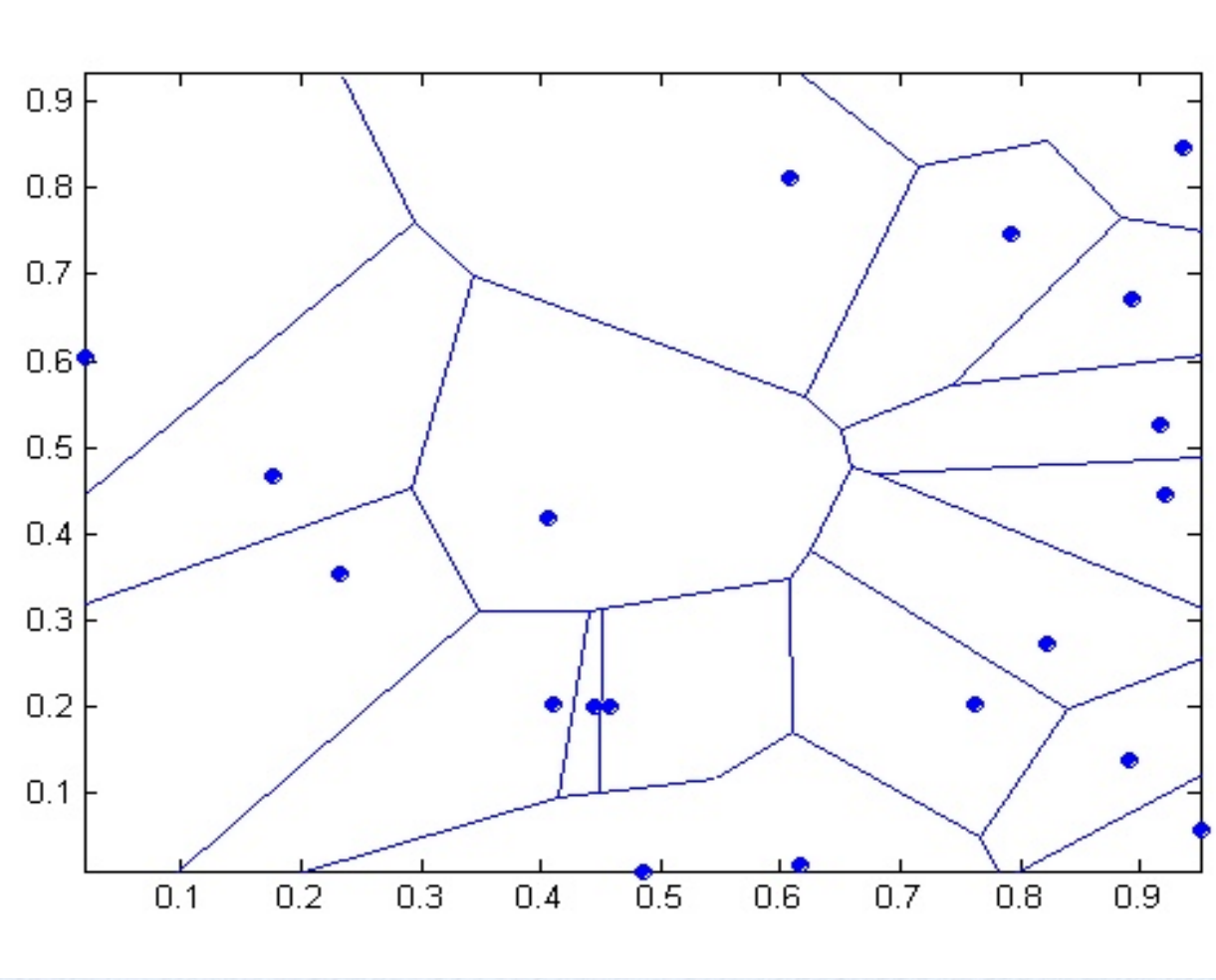
K-nearest neighbors of a record x are data points that have the k smallest distance to x

Nearest Neighbor Classification

- 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



1 NEAREST-NEIGHBOR



Voronoi Diagram

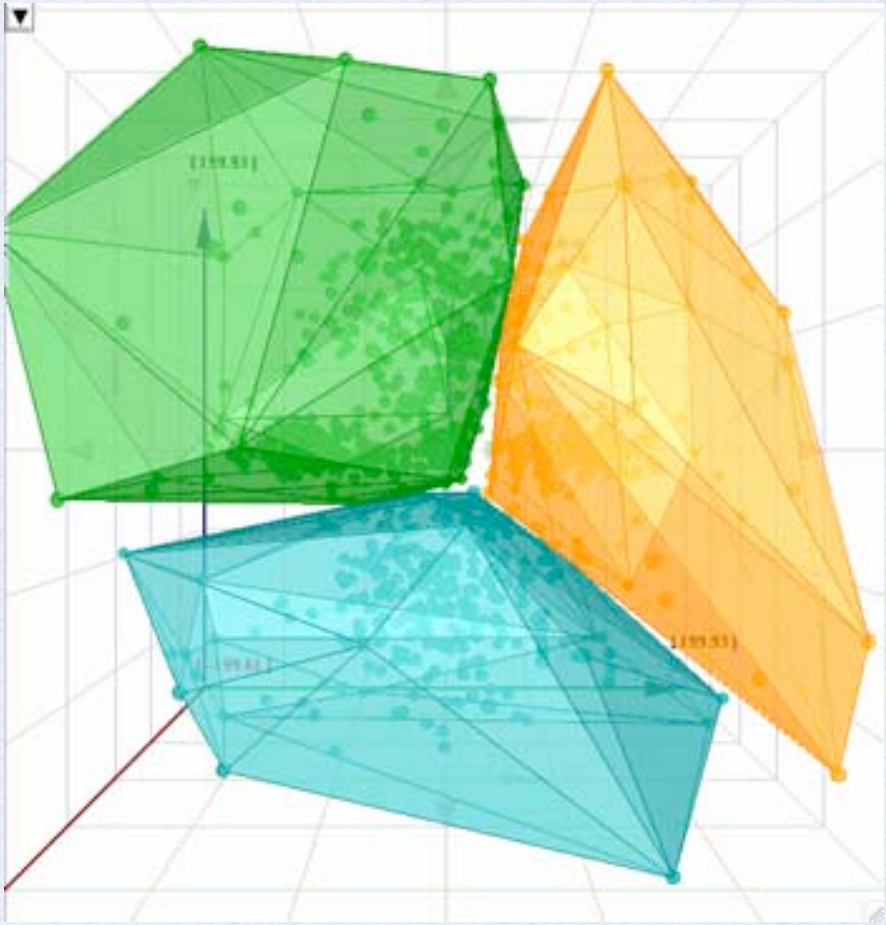
NN Classification

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

Nearest Neighbor Classification

- Attributes may have to be normalized to prevent distance measures from being dominated by one of the attributes
  - height of a person varies from 5 to 6 feet
  - weight of a person varies from 90lb to 300lb
  - income of a person varies from $10K to $1M

Multi-dimensional



Nearest Neighbor Classification

- Problem with Euclidean measure:
- High dimensional data
  - curse of dimensionality
  - Can produce counter-intuitive results

1 1 1 1 1 1 1 1 1 1 1 0		1 0 0 0 0 0 0 0 0 0 0 0	
0 1 1 1 1 1 1 1 1 1 1 1	vs	0 0 0 0 0 0 0 0 0 0 0 1	
d = 1.4142		d = 1.4142	

- Solution: Normalize the vectors to unit length
- Solution: Use different similarity metric

Nearest Neighbor Classification

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

Bayes Classifier

- Uses probabilistic framework for classification
- Conditional Probability:

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

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

- Bayes theorem:

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

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 of 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,…,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 the C with maximal $P(C \mid A_1, A_2, \ldots, A_n)$
  - 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:
  - $P(A_1, A_2, \ldots, A_n \mid C) = P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$
  - 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	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

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

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

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

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

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

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 $P(A_i \mid c)$

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

- If a normal probability 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)
  × 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

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

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

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

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

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

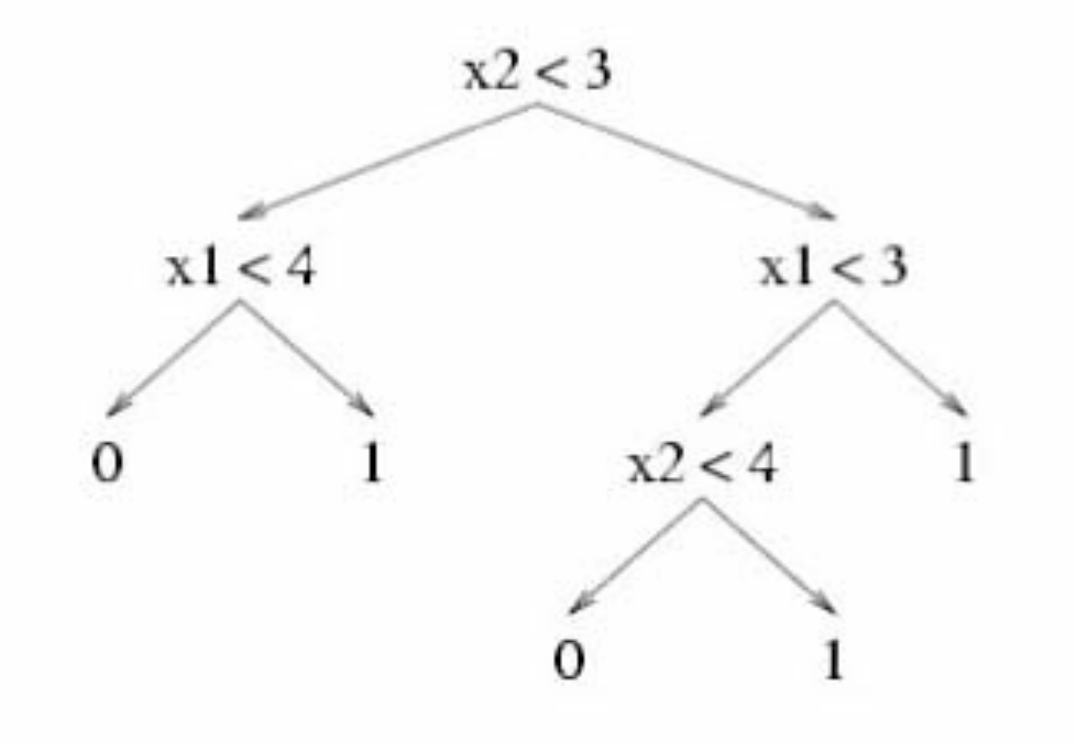
P(A|M)P(M) > P(A|N)P(N)

=> Mammals

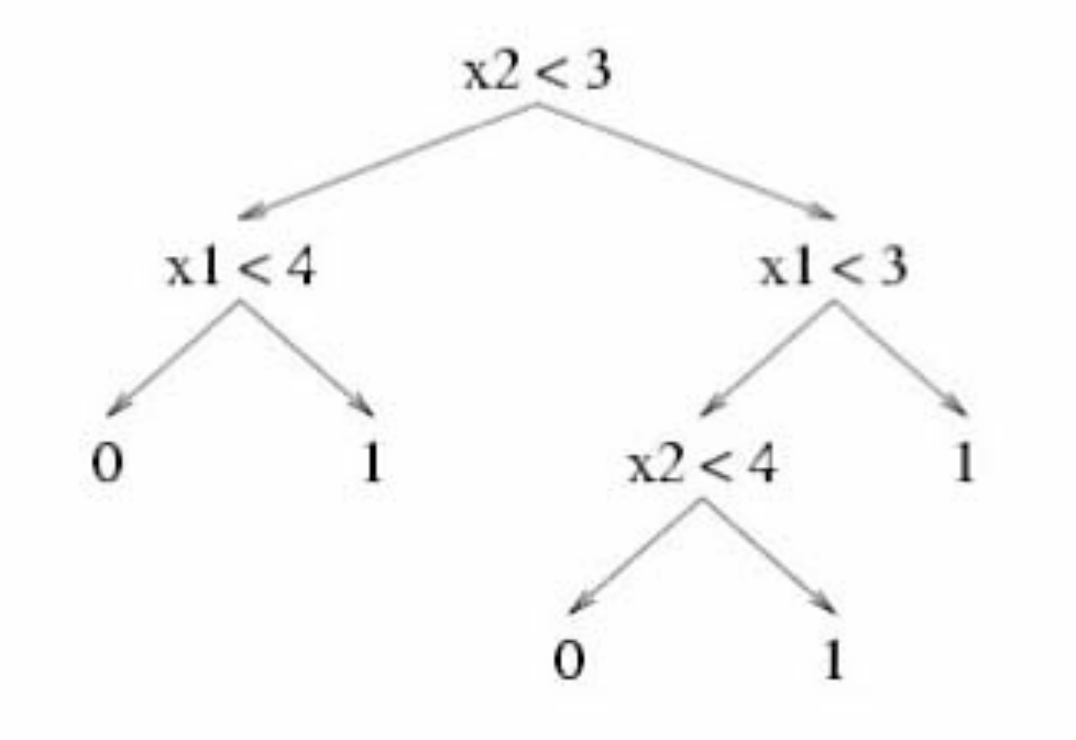
Naïve Bayes (Summary)

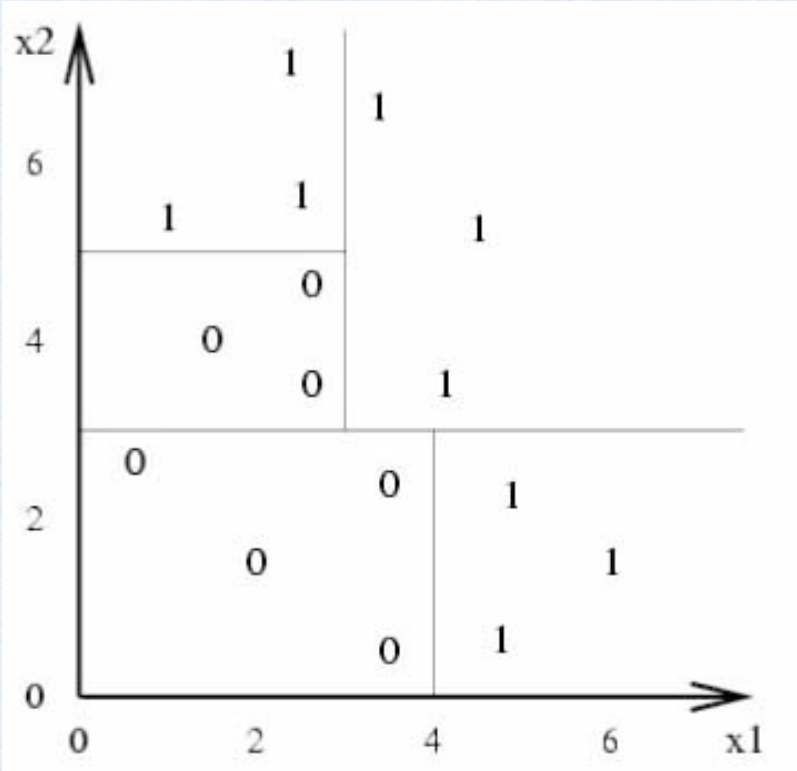
- Robust to isolated noise points (averaged out by conditional probabilitly estimates)
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
  - If X_i is irrelevant then $P(X_i|Y)$ is almost uniformly distributed and thus $P(X_i|Y)$ has little impact on posterior probability
- Independence assumption may not hold for some attributes
  - Correlated attributes can degrade the performance
  - Use other techniques such as Bayesian Belief Networks (BBN)

VS. DECISION TREES



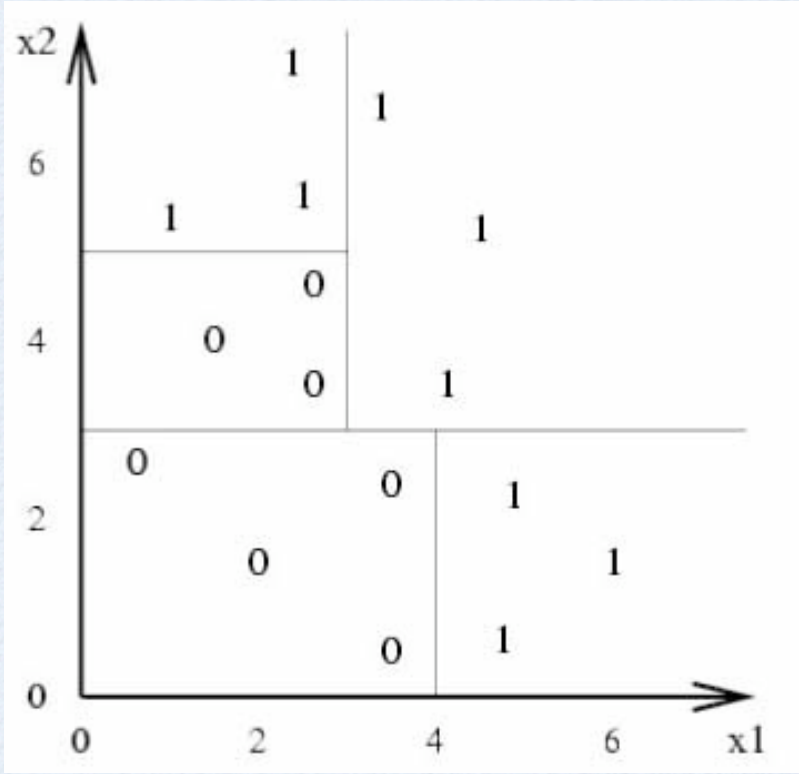
VS. DECISION TREES





VS. DECISION TREES

Decision trees divide the feature space into axis parallel rectangles.



Constructing a tree

GROWTREE(S)

if (y=0 for all $\langle \mathbf{x}, y \rangle \in S$) return new leaf(0)
else if (y=1 for all $\langle \mathbf{x}, y \rangle \in S$) return new leaf(1)
else
  choose best attribute x_i
  let S_0 = all$\langle \mathbf{x}, y \rangle \in S$ with $x_i = 0$
  let S_1 = all$\langle \mathbf{x}, y \rangle \in S$ with $x_i = 1$
  return new node (x_i, GROWTREE(S_0), GROWTREE(S_1))

COMPLICATIONS

- Again, algorithm assuming boolean features
- Multiple discrete values, handle as discussed
  - multiway split or test for one value
  - group into two disjoint subsets
- Real-valued
  - threshold or use density function

Choosing Best Attribute

- Before, we used Naïve Bayes Probability
- Another approach is based on Occam's razor

The world is simple. Therefore the smallest decision tree that works with the training set is most likely to identify unknown objects correctly.

- select to minimize disorder

DISORDER FORMULA

Average Disorder

$$= \sum_{b} \left(n_b / n_t \right) \times \left(\sum_{c} - \left(n_{bc} / n_b \right) \times \log_2 \left(n_{bc} / n_b \right) \right)$$

where

n_b = # of instances in branch b

n_t = total # of instances

n_{bc} = total # of instances in branch b of class c

DISORDER FORMULA

Average Disorder

$$= \sum_{b} \left(n_b / n_t \right) \times \left(\sum_{c} - \left(n_{bc} / n_b \right) \times \log_2 \left(n_{bc} / n_b \right) \right)$$

where

n_b = # of instances in branch b

n_t = total # of instances

n_{bc} = total # of instances in branch b of class c

DISORDER FORMULA

Average Disorder

$$= \sum_{b} \left(n_b / n_t \right) \times \left(\sum_{c} - \left(n_{bc} / n_b \right) \times \log_2 \left(n_{bc} / n_b \right) \right)$$

when $n_{bc} = 1$ and $n_b = 1$ disorder = 0 (min value)

where

n_b = # of instances in branch b

n_t = total # of instances

n_{bc} = total # of instances in branch b of class c

DISORDER FORMULA

Average Disorder

$$= \sum_{b}(n_b/n_t) \times (\sum_{c} - (n_{bc}/n_b) \times \log_2(n_{bc}/n_b))$$

when $n_{bc} = 1$ and $n_b = 2$ disorder = 1 (max value)

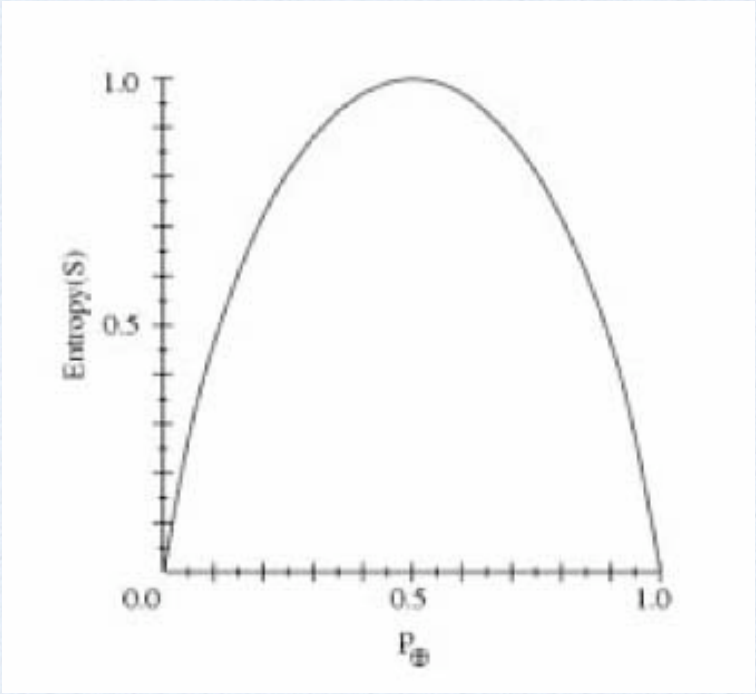
where

n_b = # of instances in branch b

n_t = total # of instances

n_{bc} = total # of instances in branch b of class c

DISORDER AKA ENTROPY



Entropy(S) = Disorder

$P_{\oplus}$ = Proportion of + instances

DISORDER FORMULA

Average Disorder

$$= \sum_{b} \left(n_b / n_t \right) \times \left(\sum_{c} - \left(n_{bc} / n_b \right) \times \log_2 \left(n_{bc} / n_b \right) \right)$$

computes weighted average of all branches

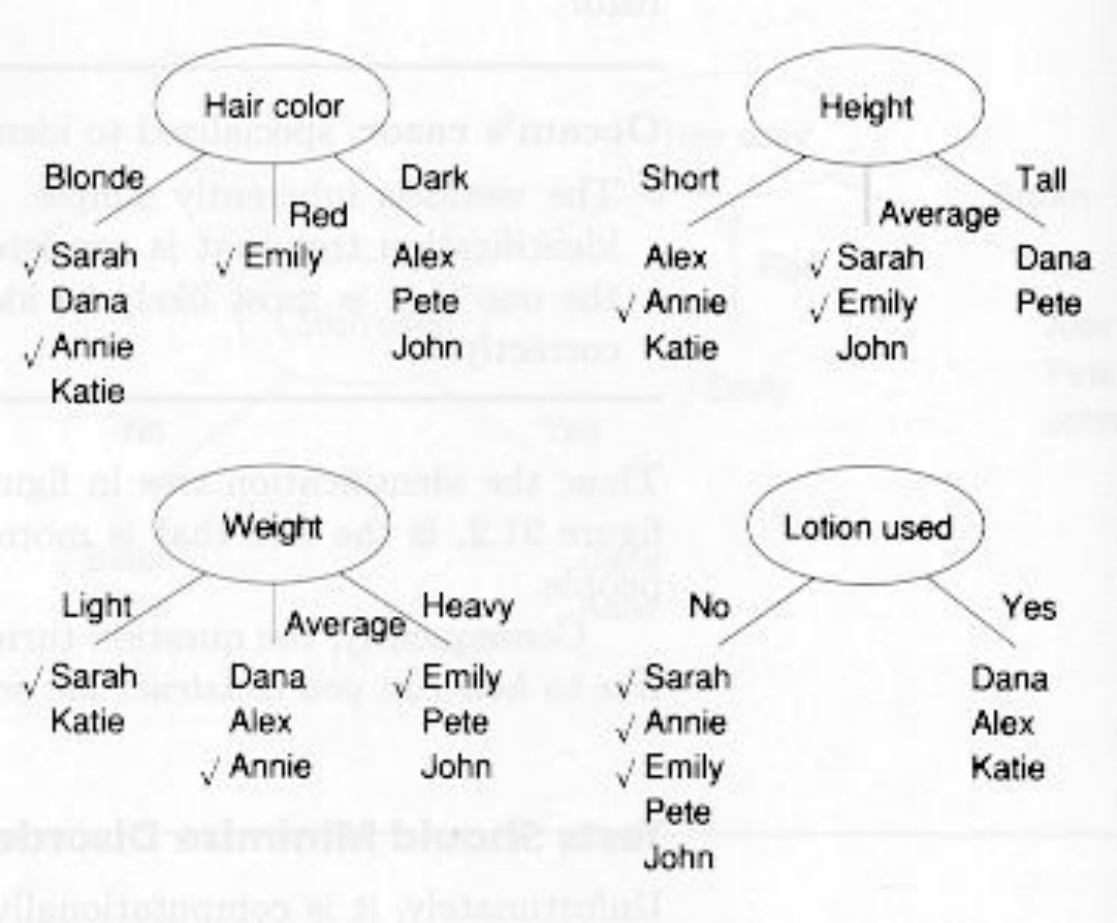
where

n_b = # of instances in branch b

n_t = total # of instances

n_{bc} = total # of instances in branch b of class c

POSSIBLE TREES



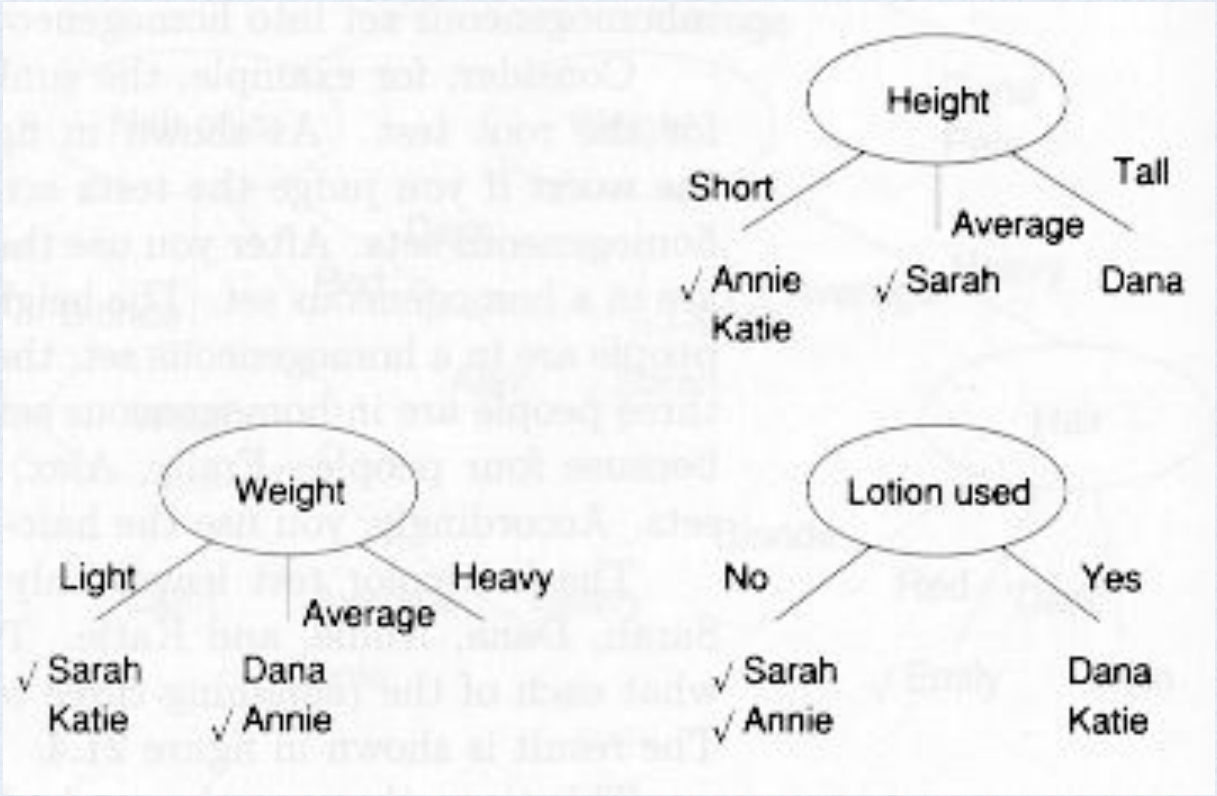
[winston 93]

Minimizing Disorder

Test	Ave. Disorder
Hair	0.5
Height	0.69
Weight	0.94
Lotion	0.61

[winston 93]

JUST THE BLONDS



[winston 93]

Minimizing Disorder

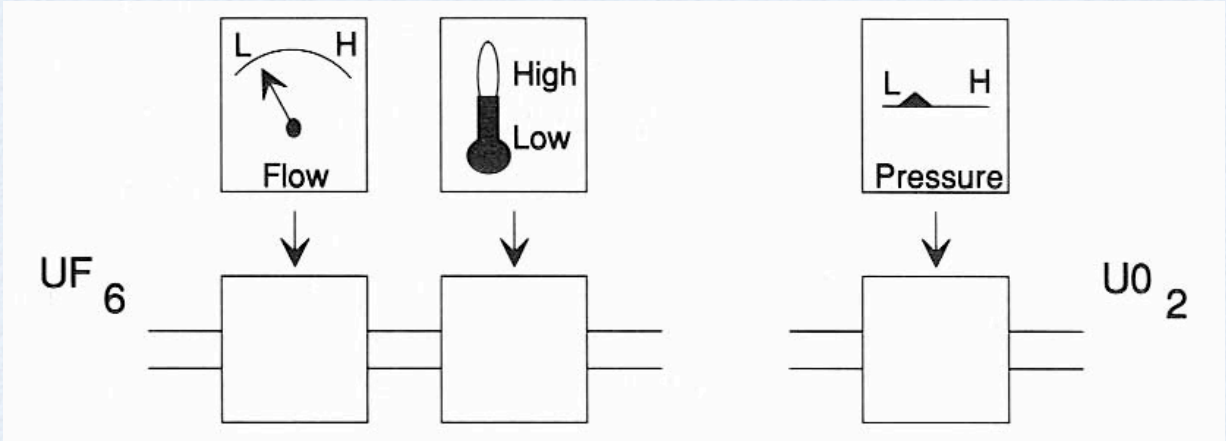
Test	Ave. Disorder
Height	0.5
Weight	1
Lotion	0

[winston 93]

APPLICATION

[winston 93]

- @ Westinghouse chemical plant converting uranium hexafluoride gas into uranium-dioxide fuel
- process takes six steps, about 30 temperatures, pressures, and flow rates (parameters)

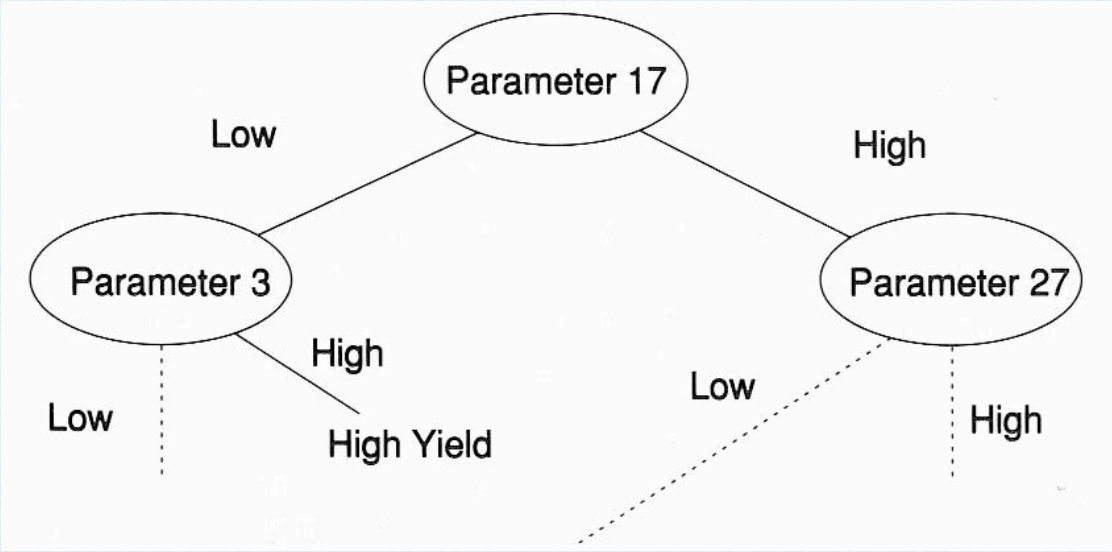


- sometimes yield high, sometimes low. WHY?

APPLICATION

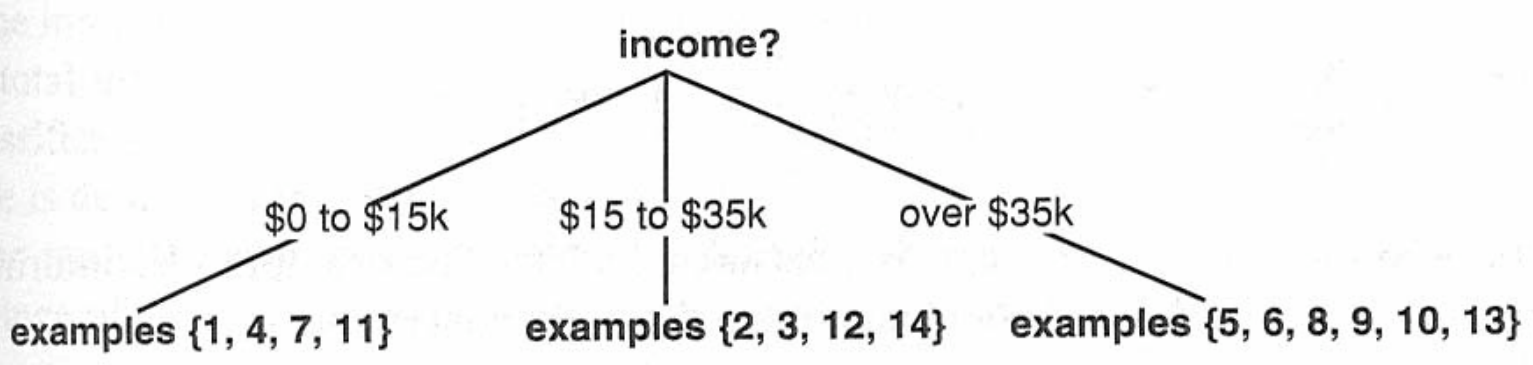
[winston 93]

Decision tree analysis, using existing observations classified yields as high or low. Making the simplest tree, and then finding the shortest path to a high yield result indicated what parameter settings should be used.

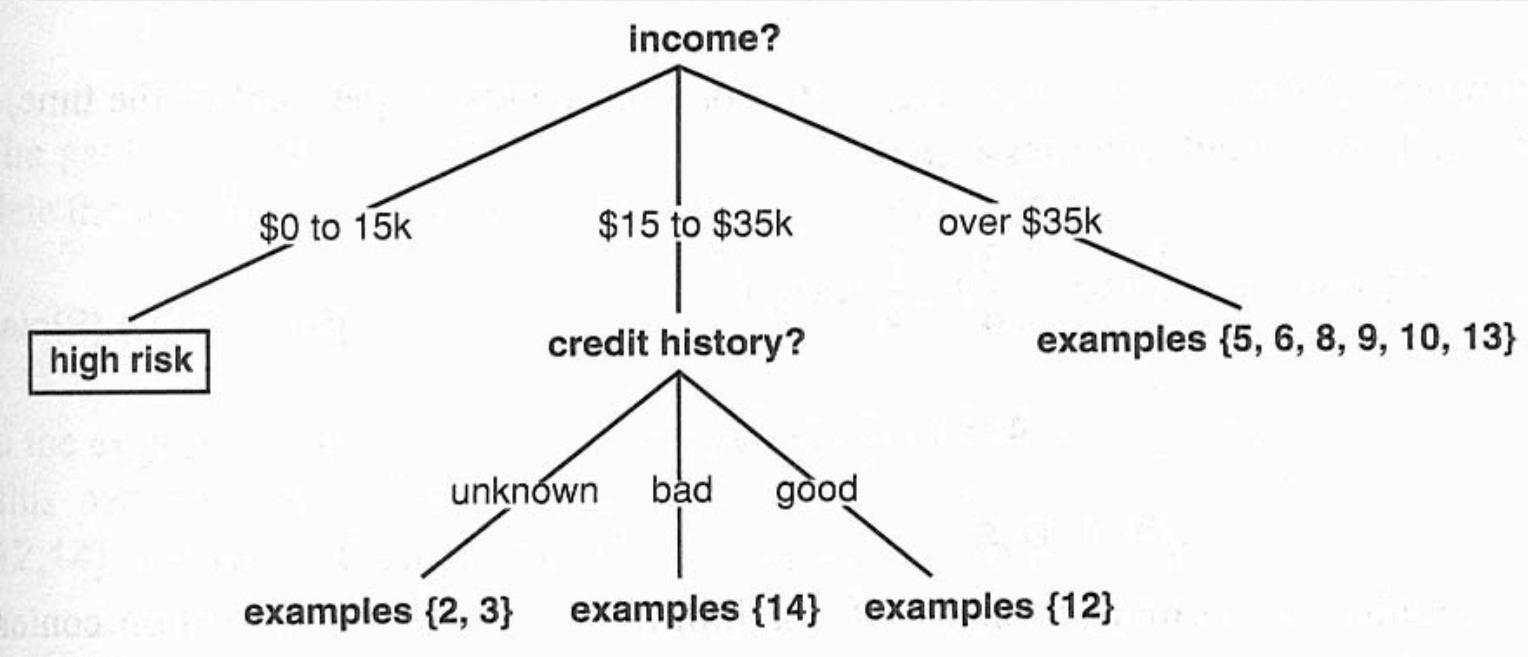


NO.	RISK	CREDIT HISTORY	DEBT	COLLATERAL	INCOME
1.	high	bad	high	none	$0 to $15k
2.	high	unknown	high	none	$15 to $35k
3.	moderate	unknown	low	none	$15 to $35k
4.	high	unknown	low	none	$0 to $15k
5.	low	unknown	low	none	over $35k
6.	low	unknown	low	adequate	over $35k
7.	high	bad	low	none	$0 to $15k
8.	moderate	bad	low	adequate	over $35k
9.	low	good	low	none	over $35k
10.	low	good	high	adequate	over $35k
11.	high	good	high	none	$0 to $15k
12.	moderate	good	high	none	$15 to $35k
13.	low	good	high	none	over $35k
14.	high	bad	high	none	$15 to $35k

POSSIBLE TREE

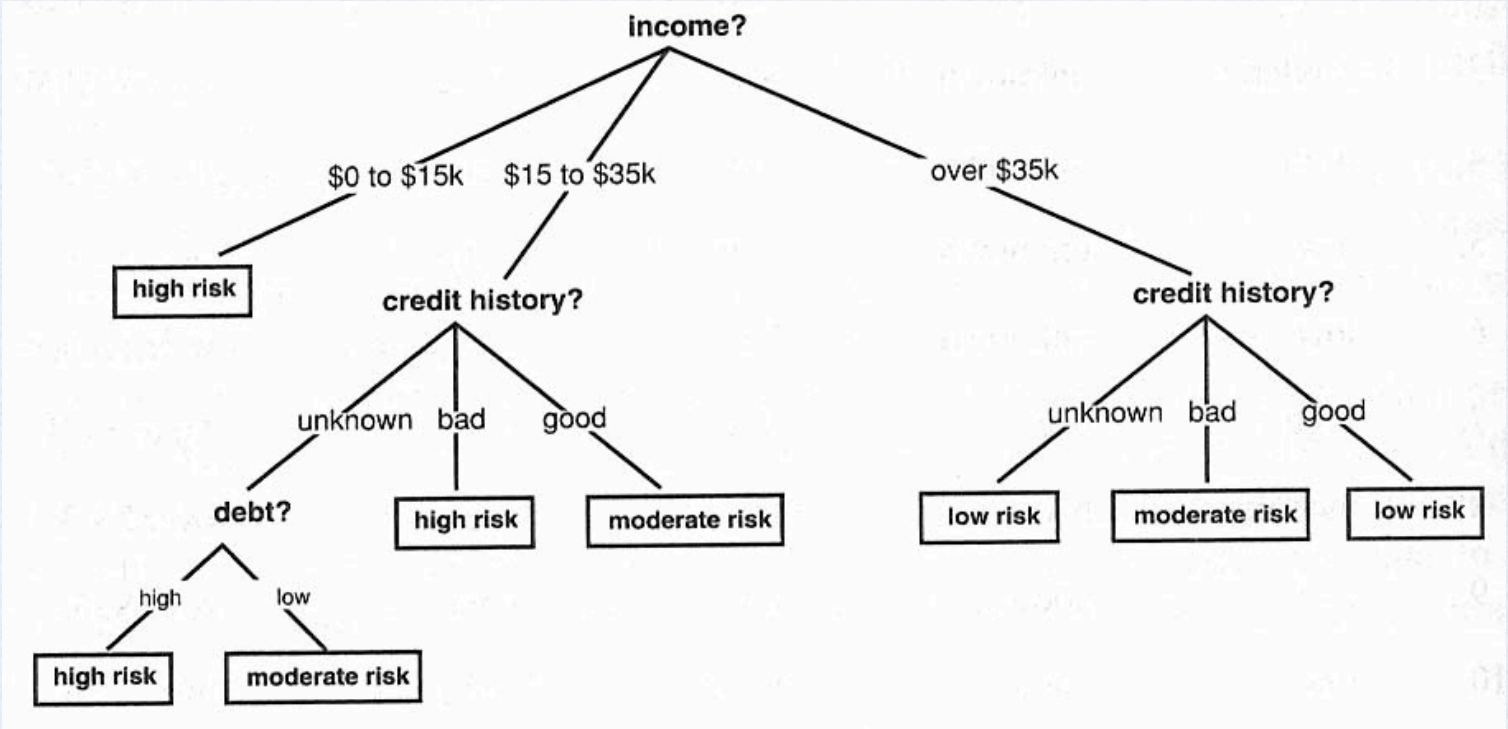


ANOTHER POSSIBLE TREE



DO THE MATH

ID3's Decision Tree



ID3 (Quinlan)

- induces concepts from examples
- top-down construction (like GROWTREE)
- Chooses "best" attribute for information **gain**.
- information gain = total information in the tree minus the weighted average of all the information in all its subtrees.

ID3 EVALUATION

Results on Chess endgame (Black to lose in 3 moves)

Size of Training Set	Percentage of Whole Universe	Errors in 10,000 Trials	Predicted Maximum Errors
200	0.01	199	728
1,000	0.07	33	146
5,000	0.36	8	29
25,000	1.79	6	7
125,000	8.93	2	1

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

Rule-based classifier (example)

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

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

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

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 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
A dogfish shark triggers none of the rules

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

BUILDING CLASSIFICATION RULES

Indirect Method:

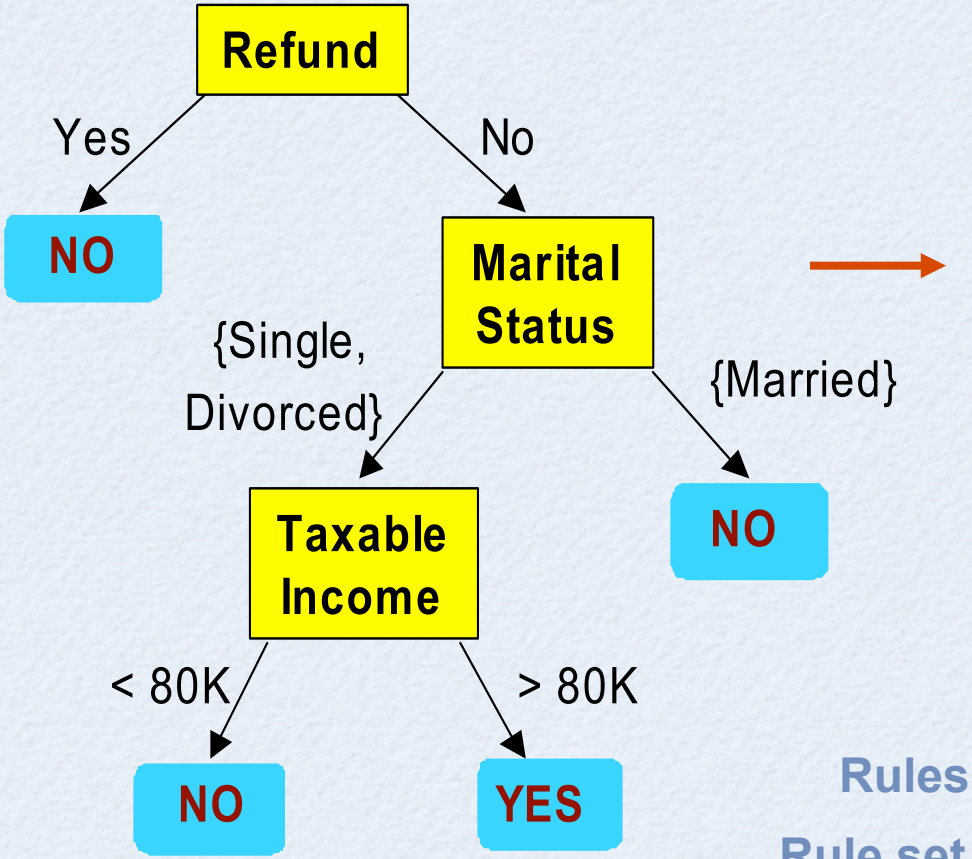
Extract rules from other classification models (e.g. decision trees, neural networks, etc).

Direct Method:

Extract rules directly from data

e.g.: RIPPER

FROM DECISION TREES TO RULES



Classification Rules

(Refund=Yes) ==> No

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

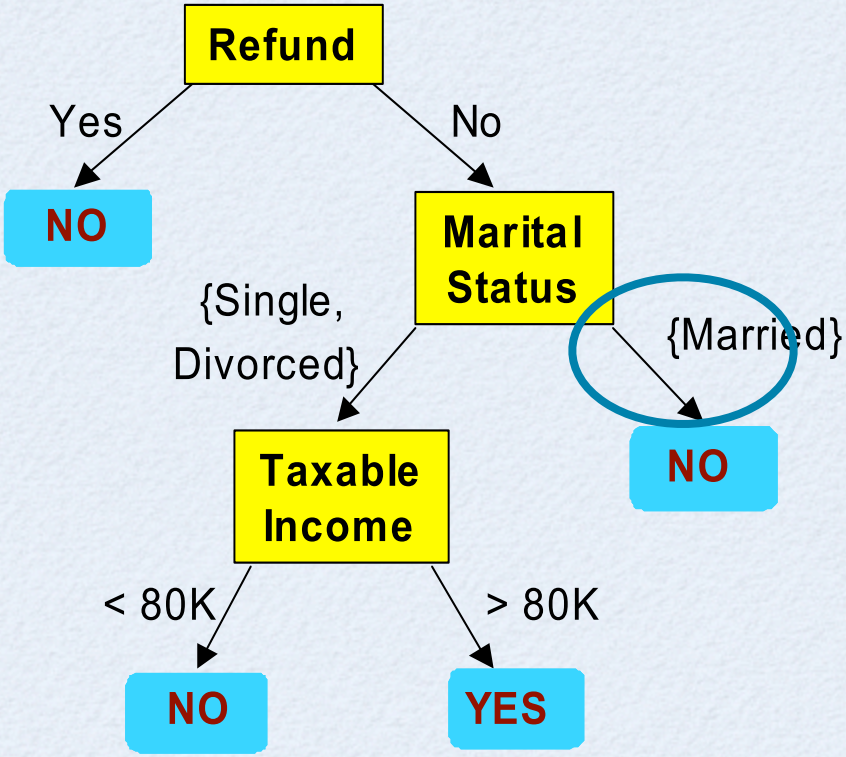
(Refund=No, Marital Status={Single,Divorced}, Taxable Income>80K) ==> Yes

(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	?

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

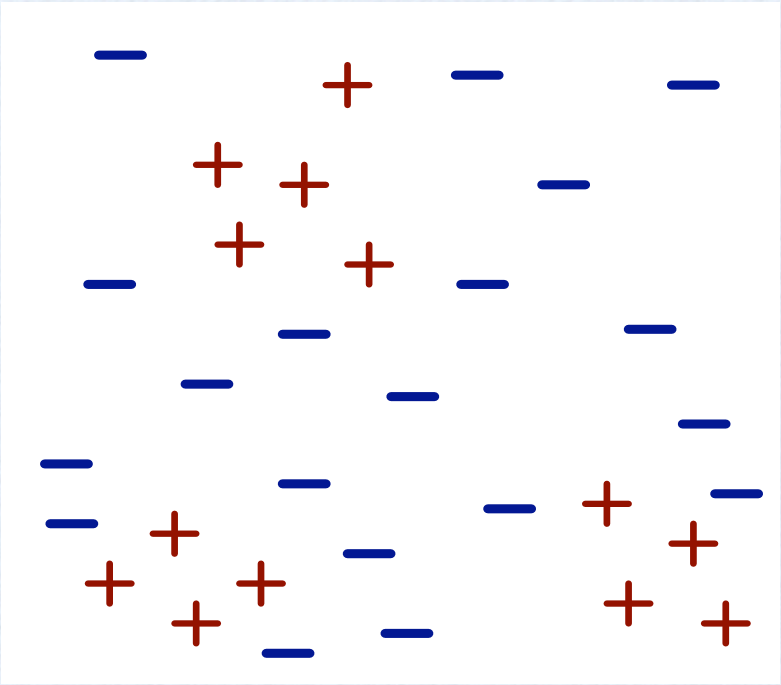
(Refund=No, Marital Status={Married}) ==> No

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

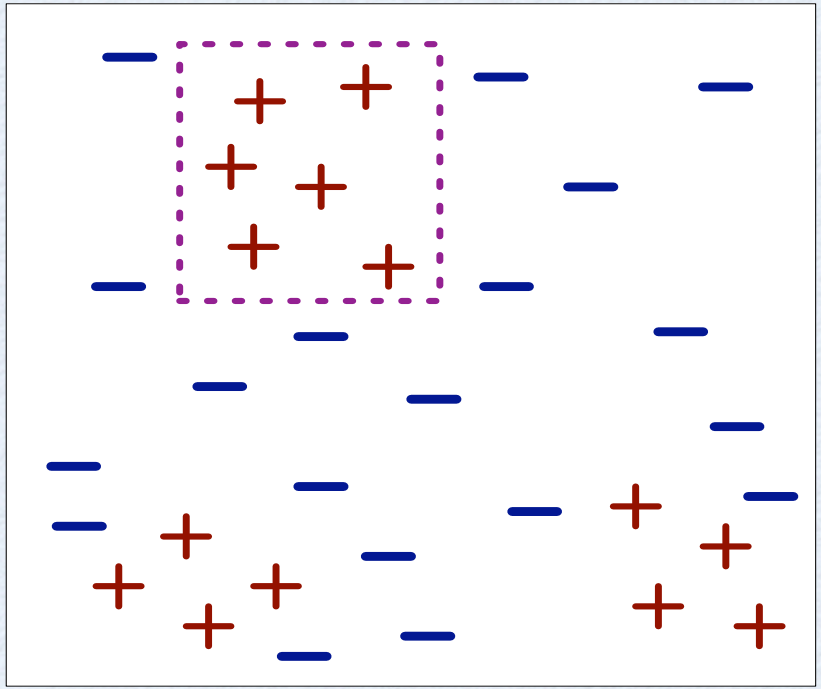
Direct Method: Sequential Covering

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

EXAMPLE OF SEQUENTIAL COVERING

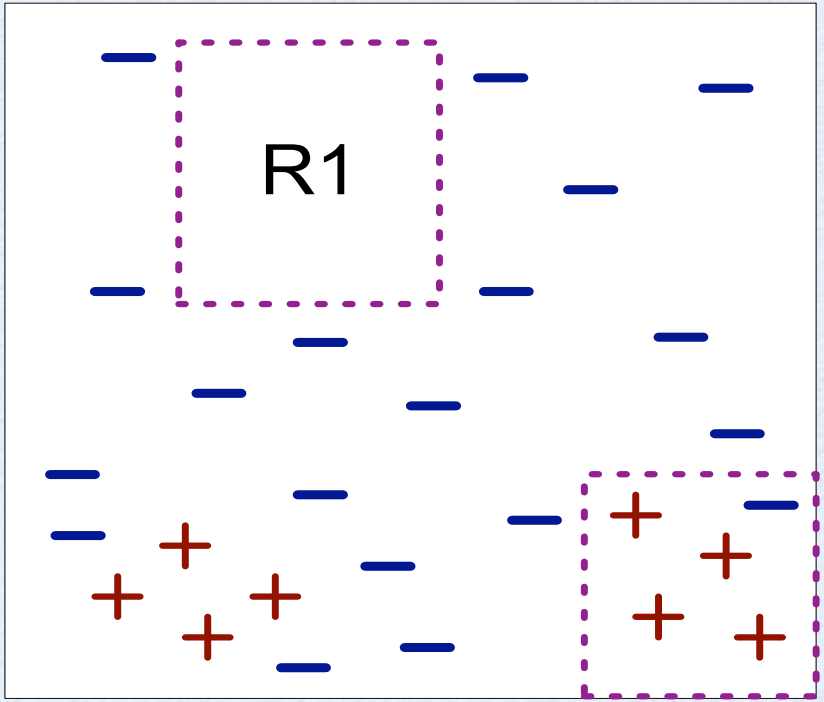


(i) Original Data

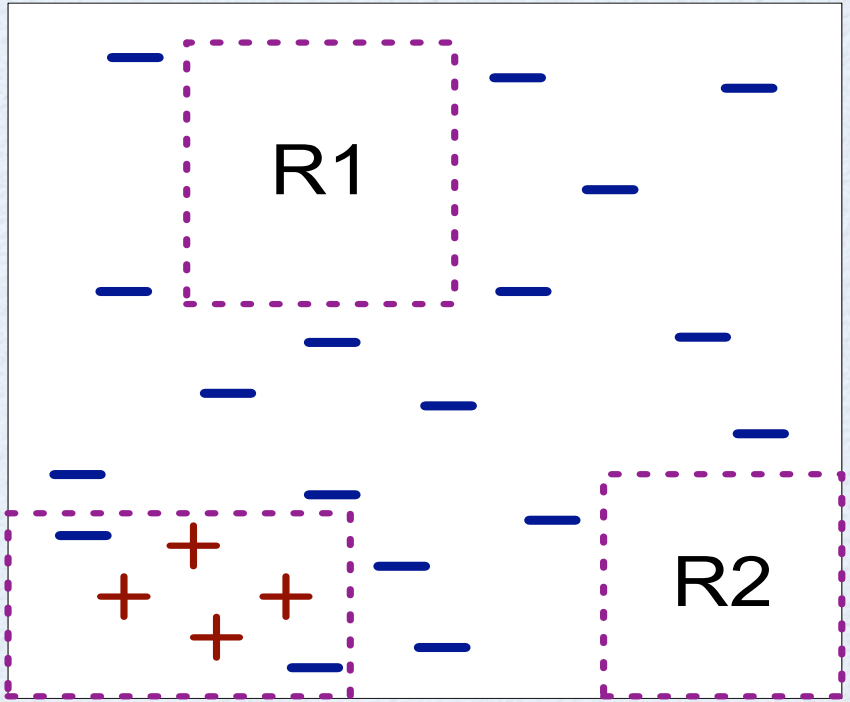


(ii) Step 1

EXAMPLE OF SEQUENTIAL COVERING...



(iii) Step 2



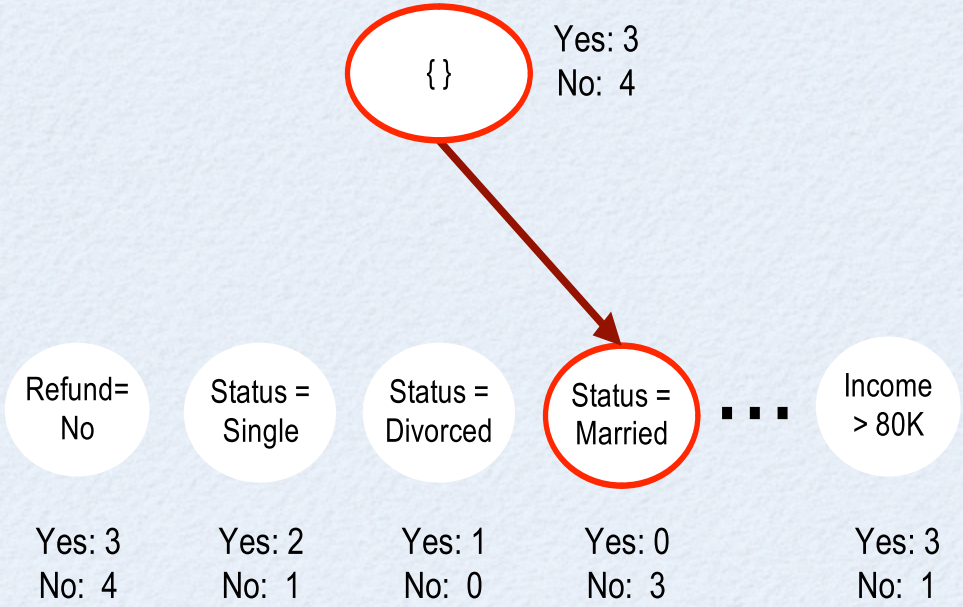
(iv) Step 3

Aspects of Sequential Covering

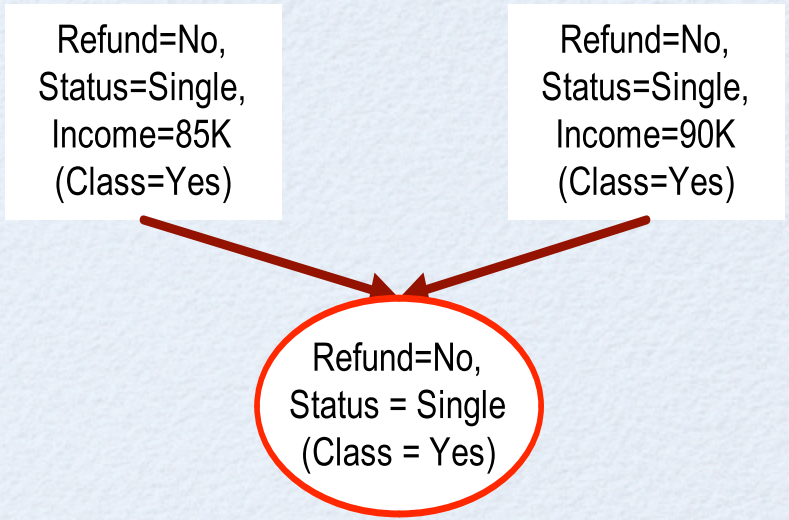
- Rule Growing
- Instance Elimination
- Rule Evaluation
- Stopping Criterion
- Rule Pruning

RULE GROWING

- Two common strategies



(a) General-to-specific



(b) Specific-to-general

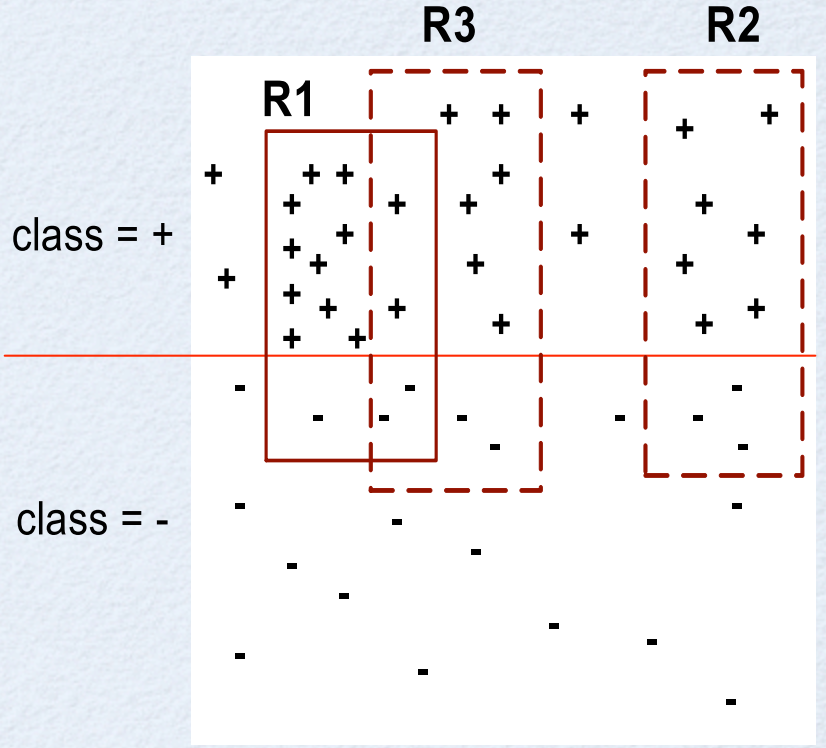
RULE GROWING (EXAMPLES)

- CN2 Algorithm:
  - Start from an empty conjunct: {}
  - Add conjuncts that minimizes the entropy measure: {A}, {A,B}, …
  - Determine the rule consequent by taking majority class of instances covered by rule

- RIPPER Algorithm:
  - Start from an empty rule: {} => class
  - Add conjuncts that maximizes FOIL's information gain measure:
    - R0: {} => class (initial rule)
    - R1: {A} => class (rule after adding conjunct)
    - Gain(R0, R1) = t [log (p1/(p1+n1)) – log (p0/(p0 + n0))]
    - where t: number of positive instances covered by both R0 and R1
      p0: number of positive instances covered by R0
      n0: number of negative instances covered by R0
      p1: number of positive instances covered by R1
      n1: number of negative instances covered by R1

INSTANCE ELIMINATION

- Why do we need to eliminate instances?
  - Otherwise, generate the same rule
- Why do we remove positive instances?
  - We've already "learned" them
- Why do we remove negative instances?
  - Prevent underestimating accuracy of rule
  - Compare rules R2 and R3 in the diagram



Rule Evaluation

- Metrics:
  - 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

k : Number of classes

p : Prior probability

Stopping criterion and rule pruning

- Stopping criterion
  - Compute the gain
  - If gain is not significant, discard the new rule

- Rule Pruning
  - Similar to post-pruning of decision trees
  - Reduced Error Pruning:
    - Remove one of the conjuncts in the rule
    - Compare error rate on validation set with/without it
    - If error decreases, prune the conjunct

Summary of Direct Method

- Grow a single rule
- Remove Instances from rule
- Prune the rule (if necessary)
- Add rule to Current Rule Set
- Repeat

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, 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

DIRECT METHOD: RIPPER

- Optimize the rule set:
  - For each rule *r* in the rule set ***R***
    - Consider 2 alternative rules:
      - Replacement rule (r*): grow new rule from scratch
      - Revised rule(r'): add conjuncts to extend the rule *r*
    - Compare the rule set for *r* against the rule set for r* and r'
    - Choose rule set that minimizes MDL principle
  - Repeat rule generation and rule optimization for the remaining positive examples

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

EXAMPLE APPLICATION

Naïve-Bayes vs. Rule-Learning in Classification of Email

Jefferson Provost
Department of Computer Sciences
The University of Texas at Austin
jp@cs.utexas.edu

Abstract

Recent growth in the use of email for communication and the corresponding growth in the volume of email received has made automatic processing of email desirable. Two learning methods, naïve bayesian learning with bag-valued features and the RIPPER rule-learning algorithm have shown promise in other text categorization tasks. I present three experiments in automatic mail foldering and spam filtering, showing that naïve bayes outperforms RIPPER in classification accuracy.

1 Introduction

The volume of email that we get is constantly growing. Most modern mail reading software packages provide some form

would become

$$from = \{jefferson, provost, jp, cs, utexas, edu\}.$$

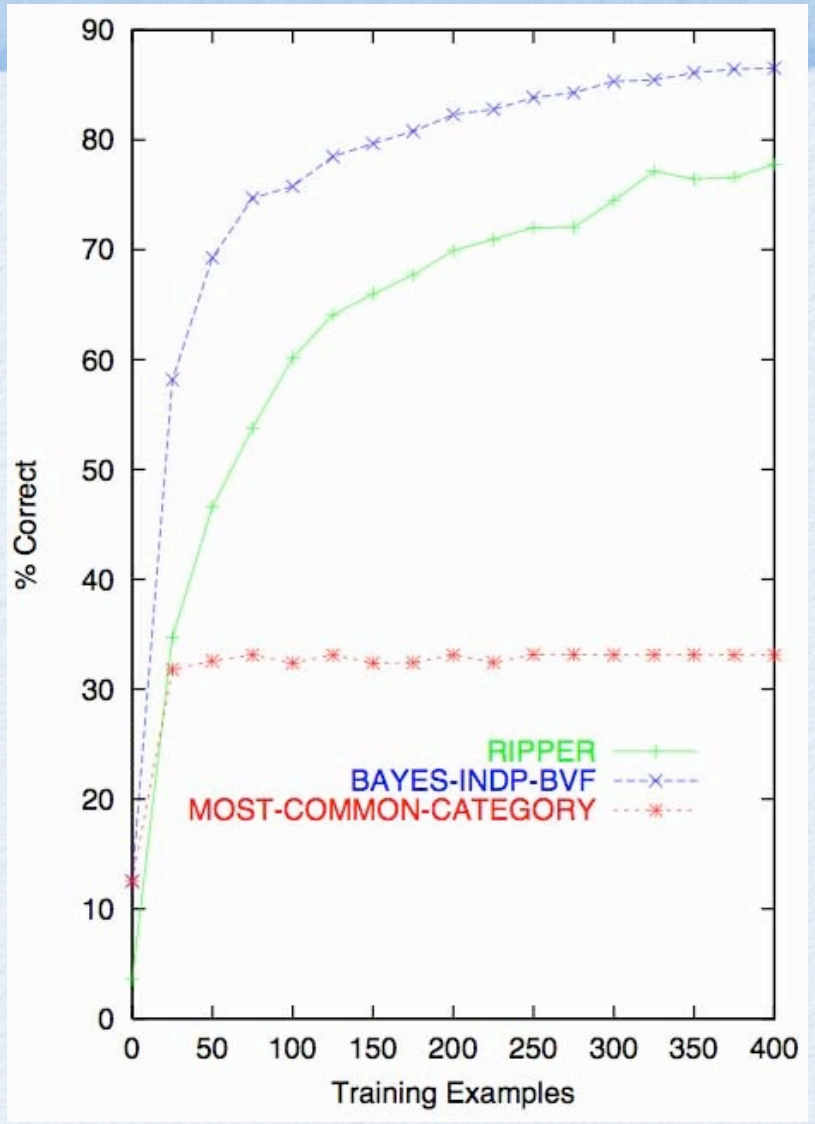
Many high-frequency, low-information content words, such as "a," "an," "the," most prepositions and conjunctions, and all single-character words are removed from the token stream before bagging, however, no complex stemming is performed.

2.2 Algorithms

The experiments in this paper compare two learning algorithms: a naïve bayesian algorithm used by Mooney et al. (1998) for text categorization, and RIPPER, a rule-learning approach used by Cohen (1996) for categorization of email

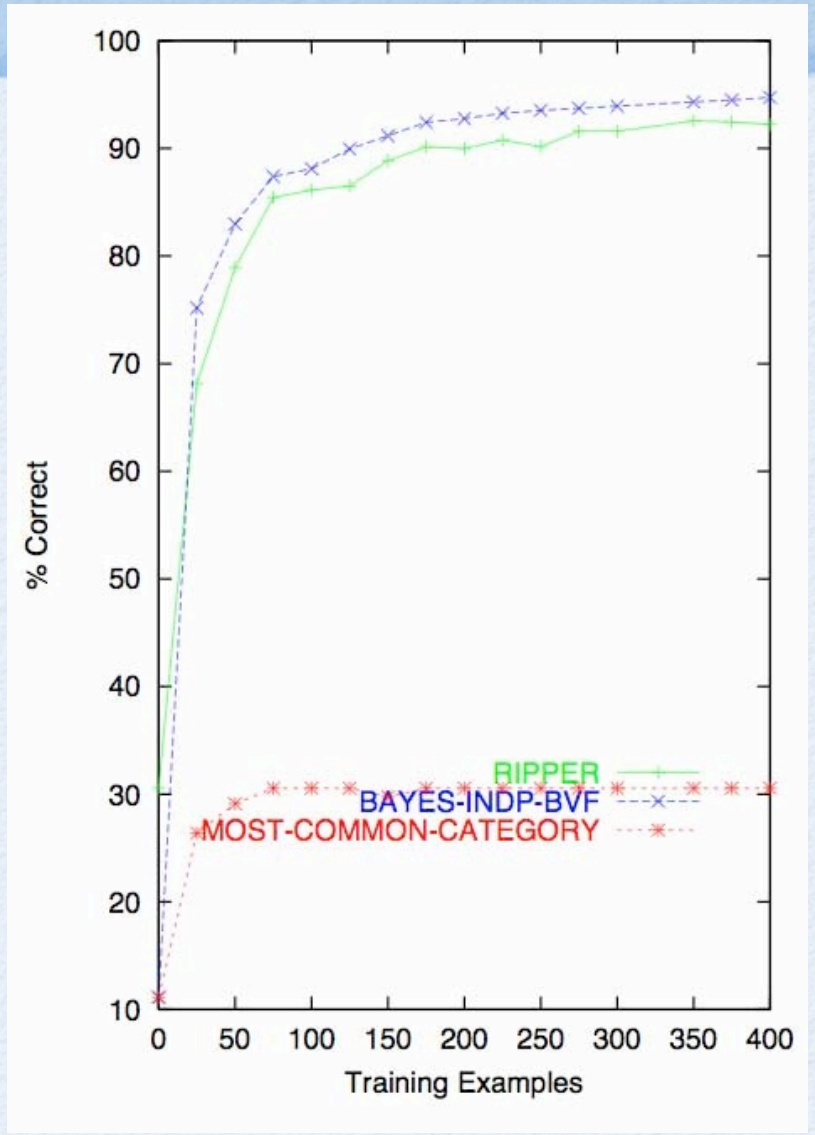
HAND SORTED DATA

[Provost 1999]

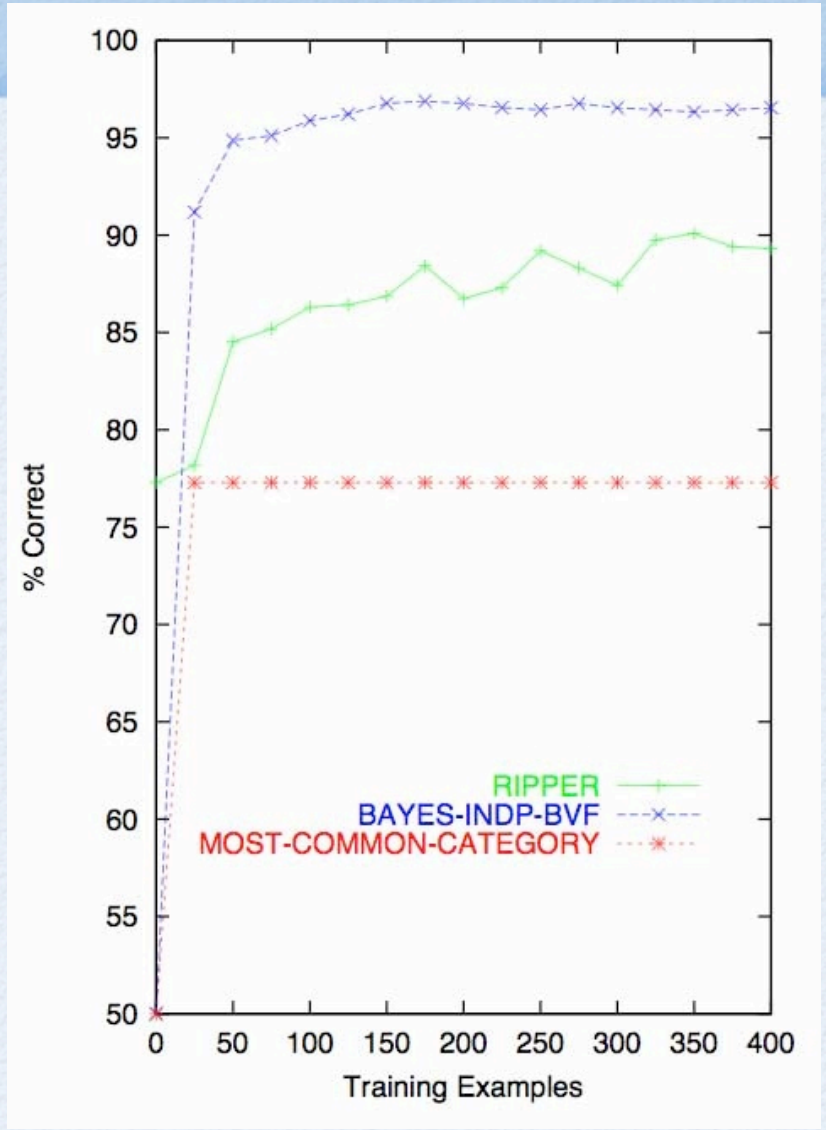


MACHINE SORTED E-MAIL

[Provost 1999]



SPAM



[Provost 1999]