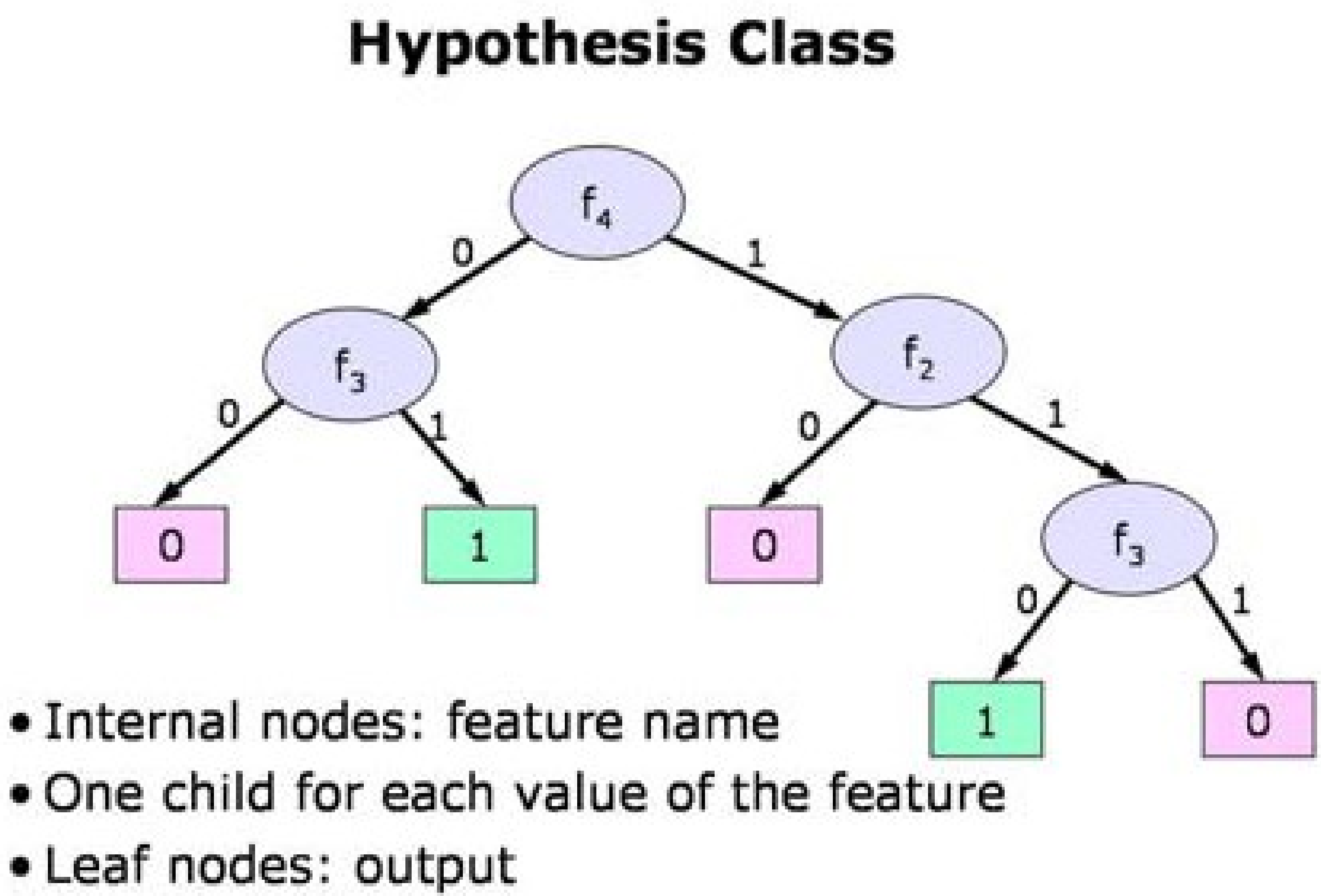
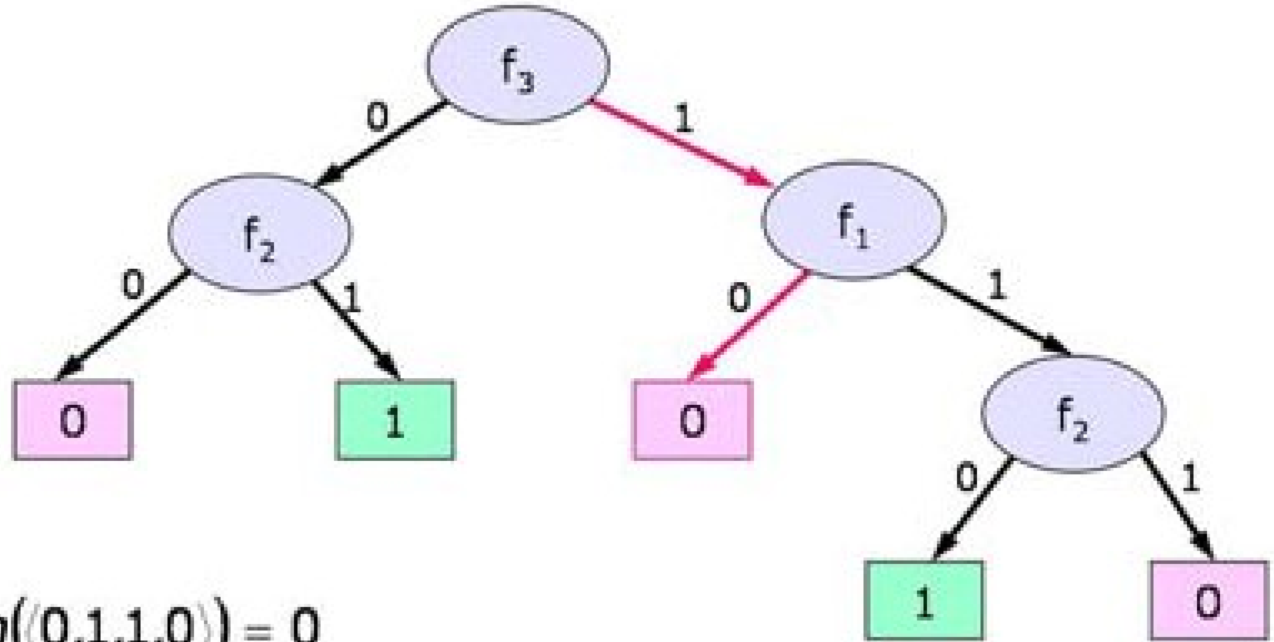
Decision Trees

Decision Trees

- DNF learning algorithm is a bit cumbersome and inefficient. Also, the exact effect of the heuristic is unclear.
- Still assume binary inputs and output, but much more broadly applicable.

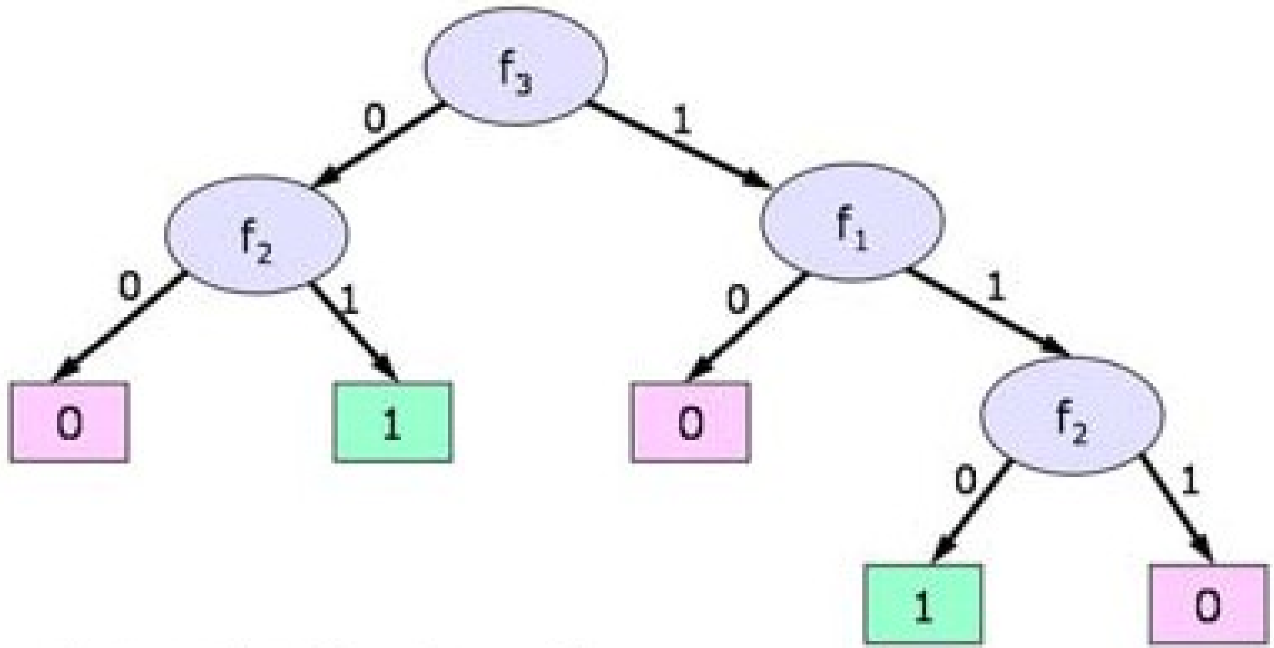

Hypothesis Class
f4
0
1
f3
f2
0
1
0
1
0
1
0
f3
0
1
1
0
• Internal nodes: feature name
• One child for each value of the feature
• Leaf nodes: output

Hypothesis Class



$h(\langle 0,1,1,0 \rangle) = 0$

Hypothesis Class



$$h = (\neg f_3 \wedge f_2) \vee (f_3 \wedge f_1 \wedge \neg f_2)$$

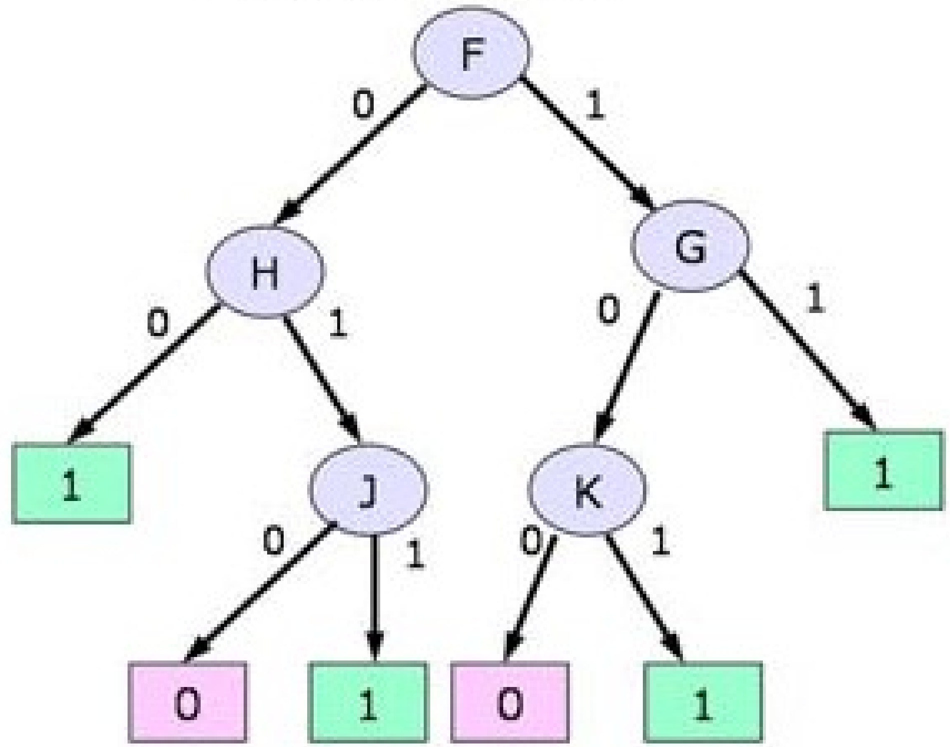
Tree Bias

- Both decision trees and DNF with negation can represent any Boolean function. So why bother with trees?
- Because we have a nice algorithm for growing trees that is consistent with a bias for simple trees (few nodes)

Tree Bias

- Both decision trees and DNF with negation can represent any Boolean function. So why bother with trees?
- Because we have a nice algorithm for growing trees that is consistent with a bias for simple trees (few nodes)
- Too hard to find the smallest good tree, so we'll be greedy again
- Have to watch out for overfitting

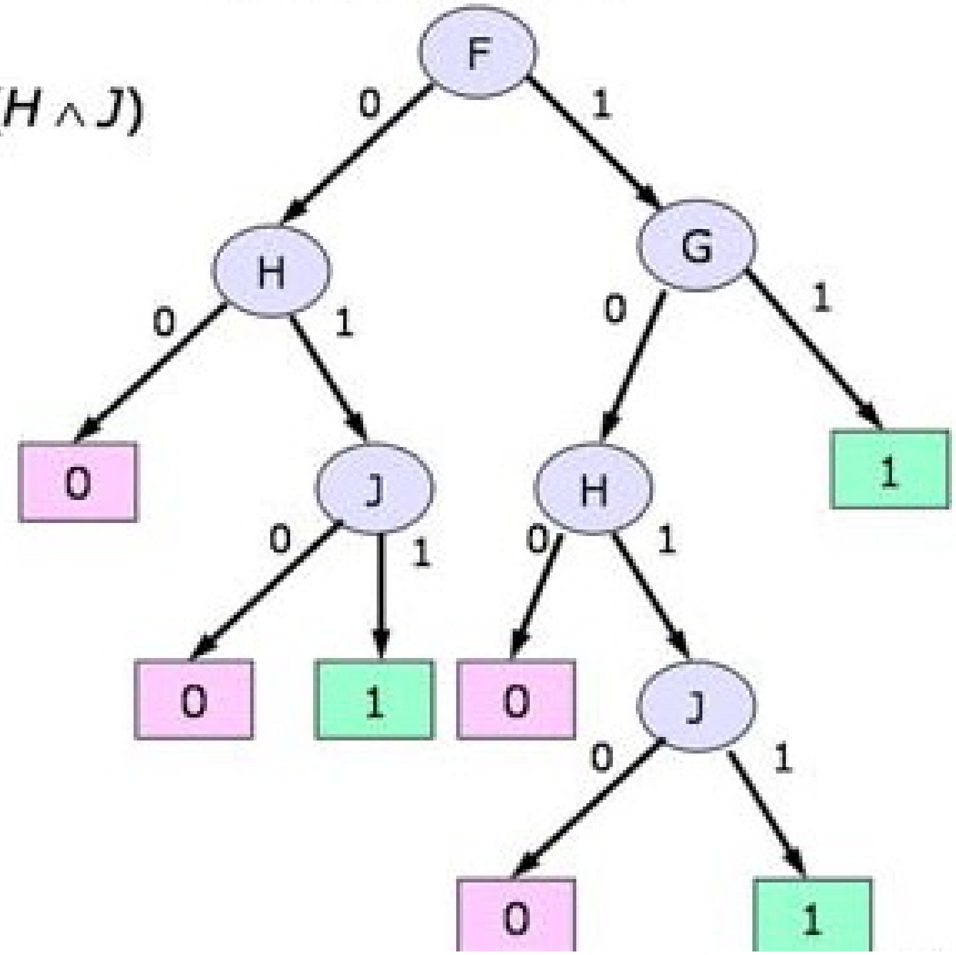
Trees vs DNF



$$(\neg F \wedge \neg H) \vee (\neg F \wedge H \wedge J) \vee (F \wedge \neg G \wedge K) \vee (F \wedge G)$$

Trees vs DNF

$(F \wedge G) \vee (H \wedge J)$



Algorithm

- Developed in parallel in AI by Quinlan and in statistics by Breiman, Friedman, Olsen and Stone

Algorithm

- Developed in parallel in AI by Quinlan and in statistics by Breiman, Friedman, Olsen and Stone

```
BuildTree (Data)
```

Algorithm

- Developed in parallel in AI by Quinlan and in statistics by Breiman, Friedman, Olsen and Stone

```
BuildTree (Data)
    if all elements of Data have the same y value, then
        MakeLeafNode(y)
```

Algorithm

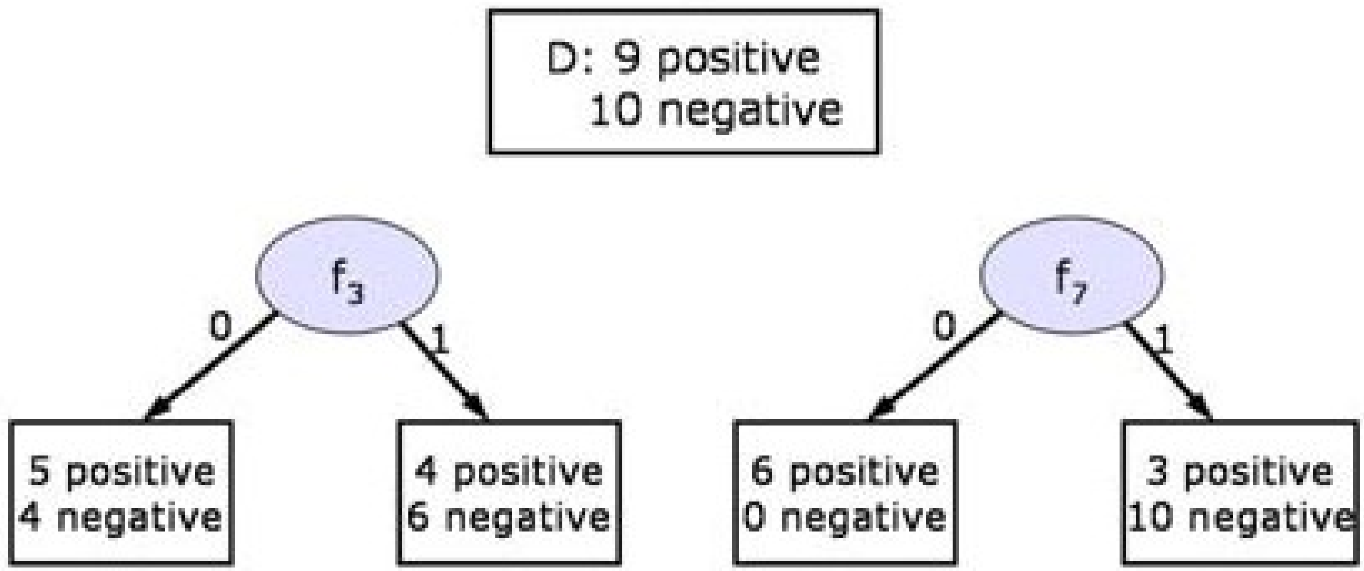
- Developed in parallel in AI by Quinlan and in statistics by Breiman, Friedman, Olsen and Stone

```
BuildTree (Data)
    if all elements of Data have the same y value, then
        MakeLeafNode(y)
    else
        feature := PickBestFeature(Data)
        MakeInternalNode(feature,
            BuildTree(SelectFalse(Data, feature)),
            BuildTree(SelectTrue(Data, feature)))
```

Let's Split

D: 9 positive 10 negative

Let's Split



D: 9 positive
10 negative
f3
0
1
5 positive
4 negative
4 positive
6 negative
f7
0
1
6 positive
0 negative
3 positive
10 negative

Entropy

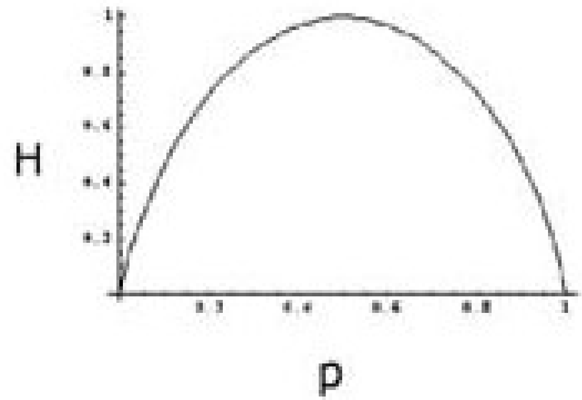
p : proportion of positive examples in a data set

$$H = -p \log_2 p - (1-p)\log_2(1-p)$$

Entropy

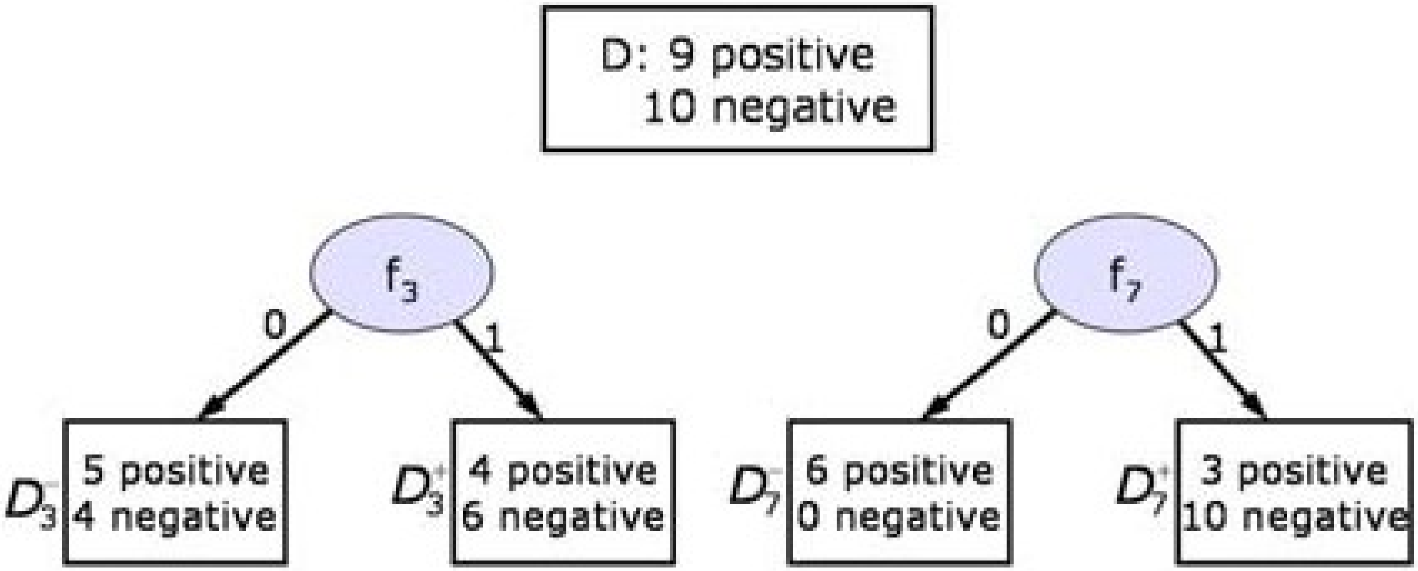
p : proportion of positive examples in a data set

$$H = -p\log_2 p - (1-p)\log_2(1-p)$$

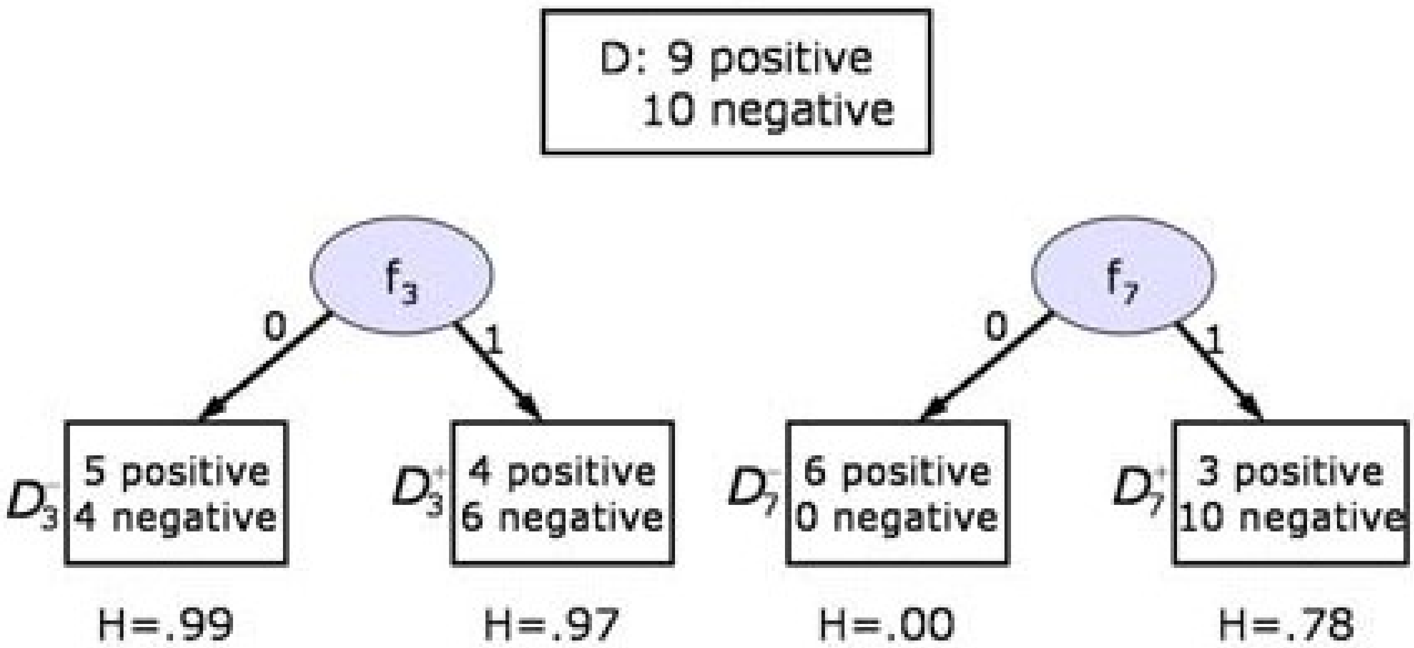


$$0\log_2 0 = 0$$

Let's Split

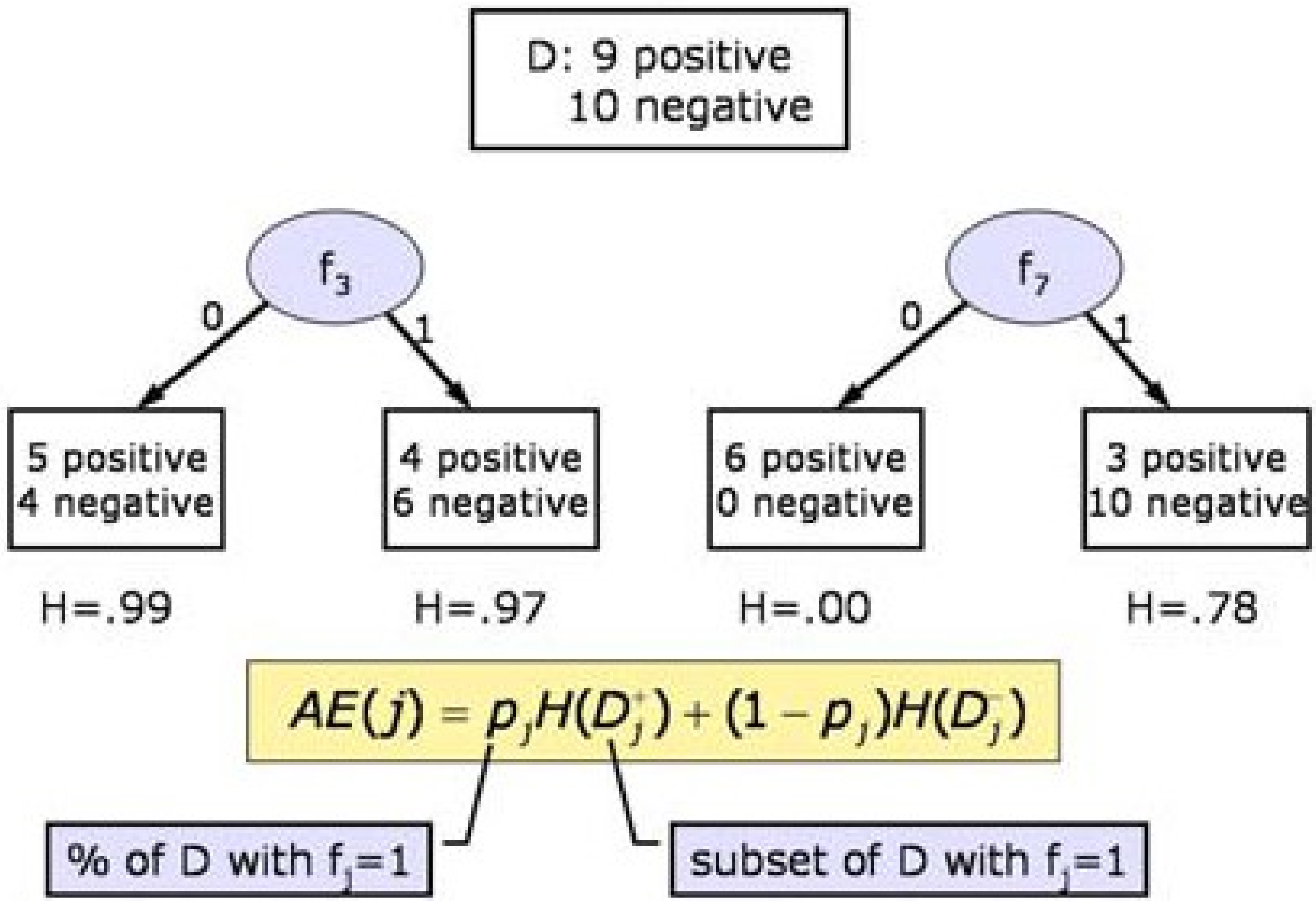

D: 9 positive
10 negative
f3
0
1
f7
0
1
D3-
5 positive
4 negative
D3+
4 positive
6 negative
D7-
6 positive
0 negative
D7+
3 positive
10 negative

Let's Split

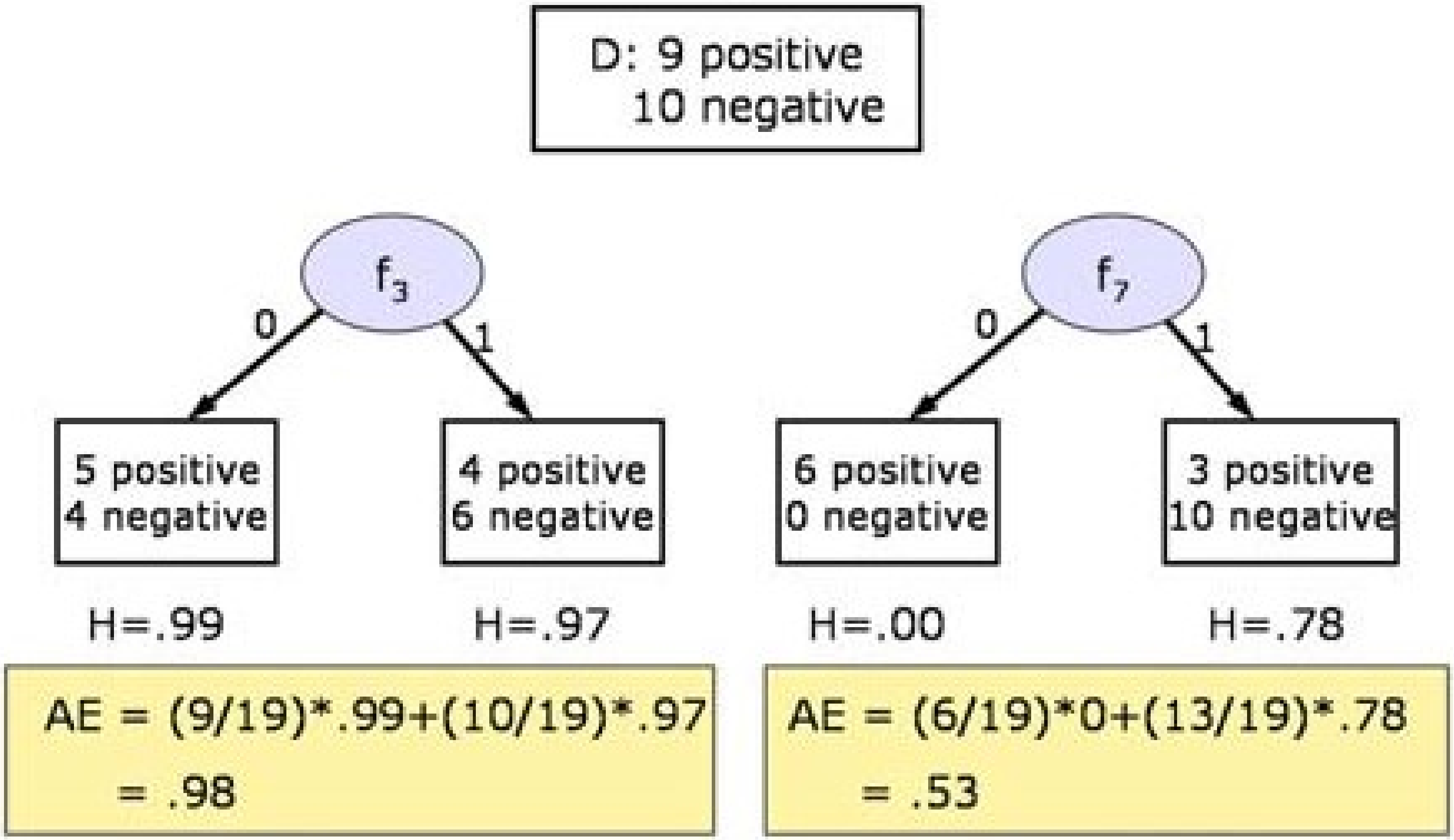


D: 9 positive
10 negative
f3
0
1
D3−
5 positive
4 negative
H=.99
D3+
4 positive
6 negative
H=.97
f7
0
1
D7−
6 positive
0 negative
H=.00
D7+
3 positive
10 negative
H=.78

Let's Split


D: 9 positive
10 negative
f_3
0
1
5 positive
4 negative
4 positive
6 negative
H=.99
H=.97
f_7
0
1
6 positive
0 negative
3 positive
10 negative
H=.00
H=.78
$AE(j) = p_j H(D_j^+) + (1 - p_j) H(D_j^-)$
% of D with f_j=1
subset of D with f_j=1

Let's Split


D: 9 positive
10 negative
f3
0
1
5 positive
4 negative
4 positive
6 negative
H=.99
H=.97
AE = (9/19)*.99+(10/19)*.97
= .98
f7
0
1
6 positive
0 negative
3 positive
10 negative
H=.00
H=.78
AE = (6/19)*0+(13/19)*.78
= .53

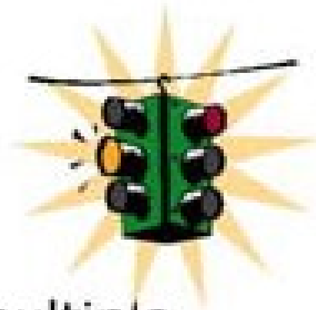
Algorithm

- Developed in parallel in AI by Quinlan and in statistics by Breiman, Friedman, Olshen and Stone

```
BuildTree (Data)
    if all elements of Data have the same y value, then
        MakeLeafNode(y)
    else
        feature := PickBestFeature(Data)
        MakeInternalNode(feature,
            BuildTree(SelectFalse(Data, feature)),
            BuildTree(SelectTrue(Data, feature)))
```

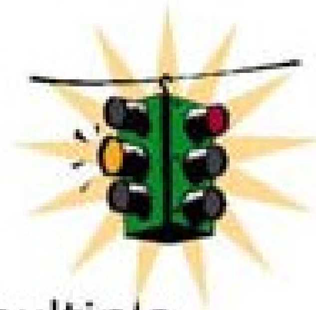
- Best feature minimizes average entropy of data in the children

Stopping



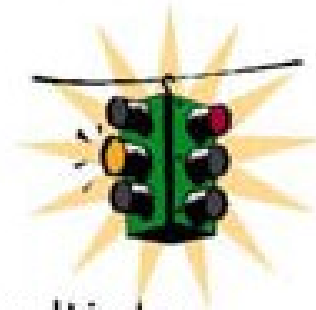
- Stop recursion if data contains only multiple instances of the same x with different y values

Stopping



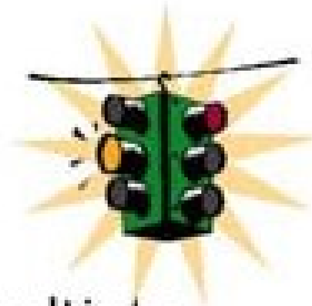
- Stop recursion if data contains only multiple instances of the same x with different y values
  - Make leaf node with output equal to the y value that occurs in the majority of the cases in the data

Stopping



- Stop recursion if data contains only multiple instances of the same x with different y values
  - Make leaf node with output equal to the y value that occurs in the majority of the cases in the data
- Consider stopping to avoid overfitting when:

Stopping

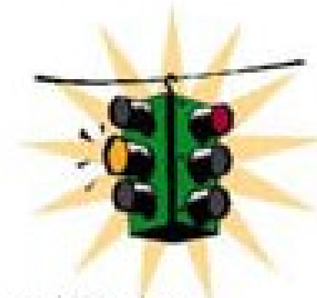


- Stop recursion if data contains only multiple instances of the same x with different y values
  - Make leaf node with output equal to the y value that occurs in the majority of the cases in the data
- Consider stopping to avoid overfitting when:
  - entropy of a data set is below some threshold

Stopping

- Stop recursion if data contains only multiple instances of the same x with different y values
  - Make leaf node with output equal to the y value that occurs in the majority of the cases in the data
- Consider stopping to avoid overfitting when:
  - entropy of a data set is below some threshold
  - number elements in a data set is below threshold

Stopping



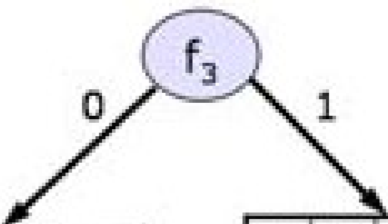
- Stop recursion if data contains only multiple instances of the same x with different y values
  - Make leaf node with output equal to the y value that occurs in the majority of the cases in the data
- Consider stopping to avoid overfitting when:
  - entropy of a data set is below some threshold
  - number elements in a data set is below threshold
  - best next split doesn't decrease average entropy (but this can get us into trouble)

Simulation

f_1	f_2	f_3	f_4	y
0	1	1	0	0
1	0	1	1	1
1	1	1	0	1
0	0	1	1	1
1	0	0	1	0
0	1	1	1	0

- $H(D) = .92$
- $AE_1=.92$, $AE_2=.92$, $AE_3=.81$, $AE_4=1$

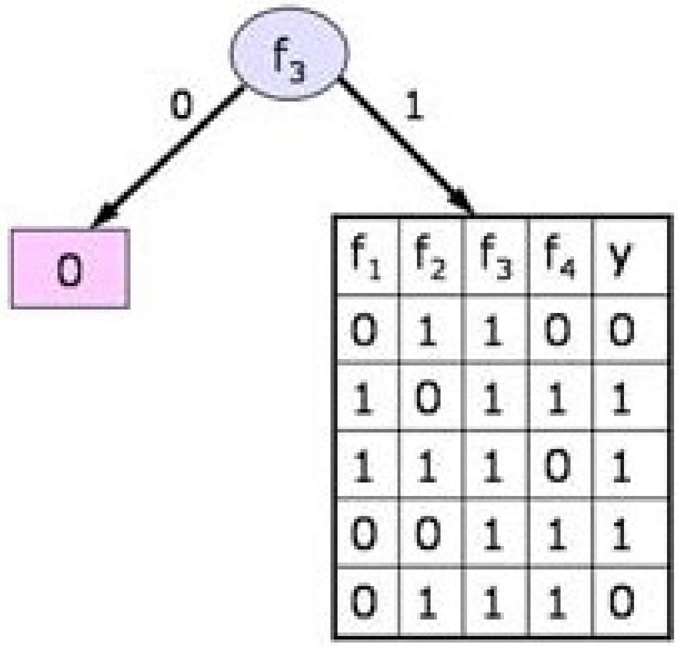
Simulation



f_1	f_2	f_3	f_4	y
1	0	0	1	0

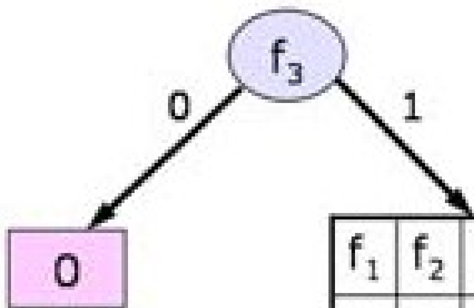
f_1	f_2	f_3	f_4	y
0	1	1	0	0
1	0	1	1	1
1	1	1	0	1
0	0	1	1	1
0	1	1	1	0

Simulation



f_1	f_2	f_3	f_4	y
0	1	1	0	0
1	0	1	1	1
1	1	1	0	1
0	0	1	1	1
0	1	1	1	0

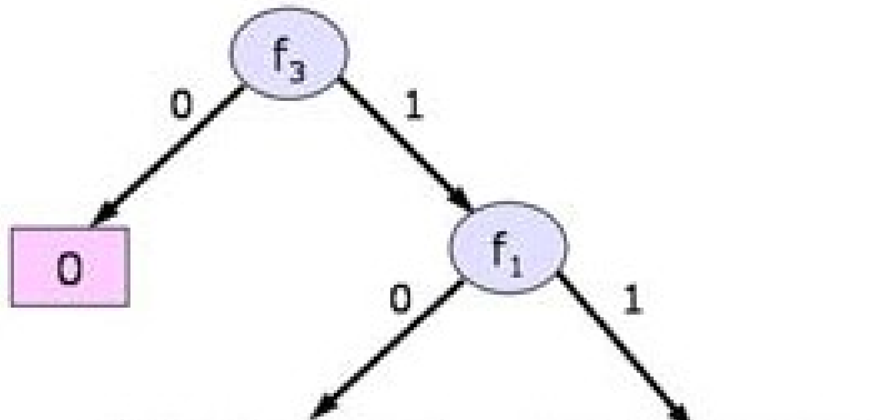
Simulation



f_1	f_2	f_3	f_4	y
0	1	1	0	0
1	0	1	1	1
1	1	1	0	1
0	0	1	1	1
0	1	1	1	0

- AE_1=.55, AE_2=.55, AE_4=.95

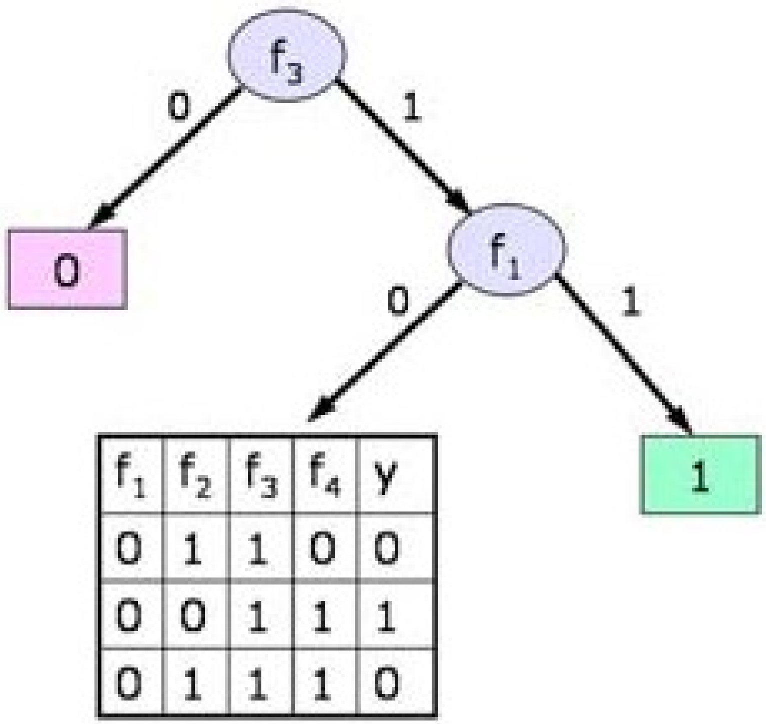
Simulation



f_1	f_2	f_3	f_4	y
0	1	1	0	0
0	0	1	1	1
0	1	1	1	0

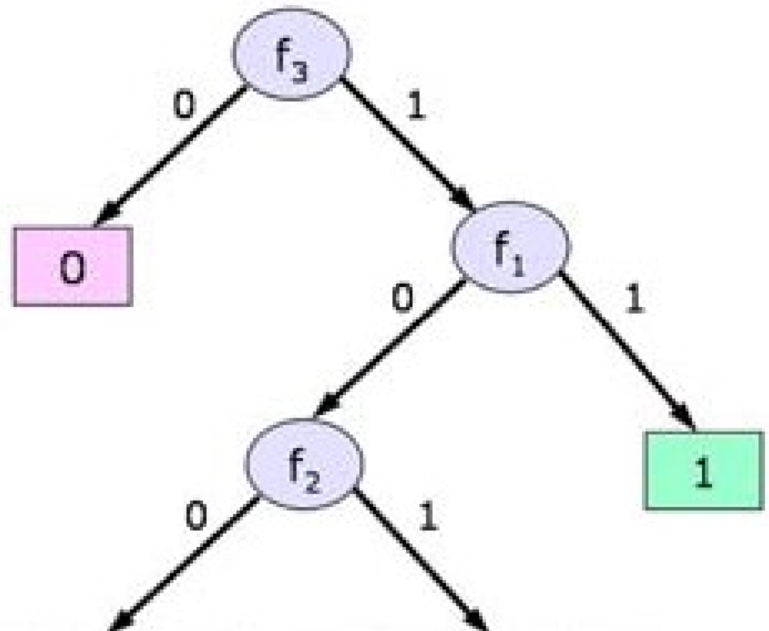
f_1	f_2	f_3	f_4	y
1	0	1	1	1
1	1	1	0	1

Simulation



f_1	f_2	f_3	f_4	y
0	1	1	0	0
0	0	1	1	1
0	1	1	1	0

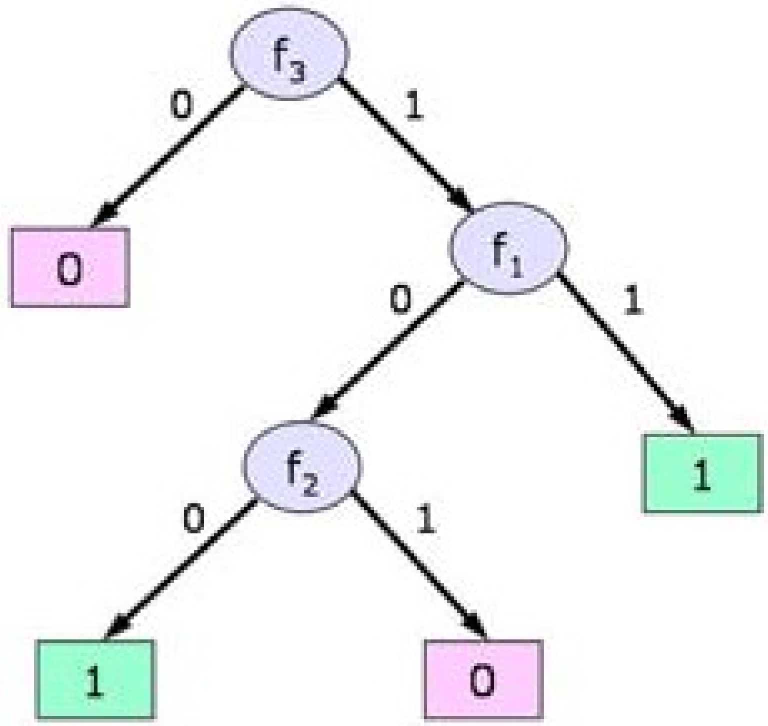
Simulation



f_1	f_2	f_3	f_4	y
0	0	1	1	1

f_1	f_2	f_3	f_4	y
0	1	1	0	0
0	1	1	1	0

Simulation



Exclusive OR

$$\boxed{(A \wedge \neg B) \vee (\neg A \wedge B)}$$

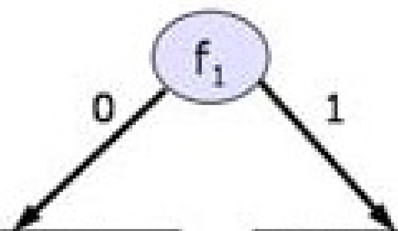
Exclusive OR

$$(A \wedge \neg B) \vee (\neg A \wedge B)$$

f_1	f_2	f_3	f_4	y
0	1	1	0	0
1	0	1	0	0
1	1	1	0	1
0	0	0	1	1
1	0	0	1	0
0	1	0	1	0

- H(D) = .92
- AE_1=.92, AE_2=.92, AE_3=.92, AE_4=.92

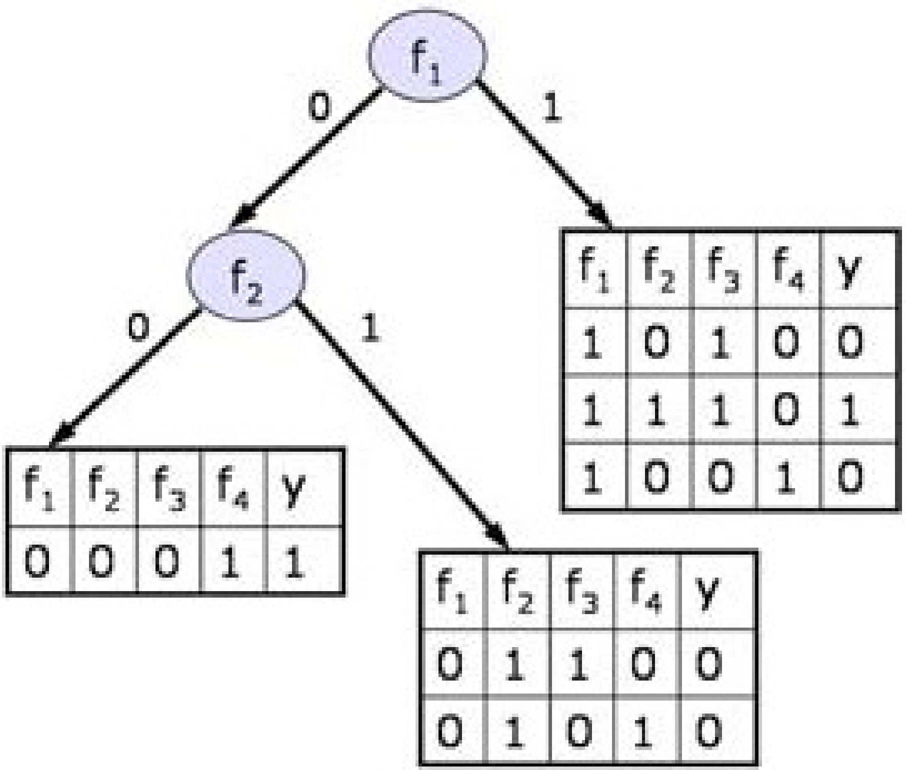
Exclusive OR



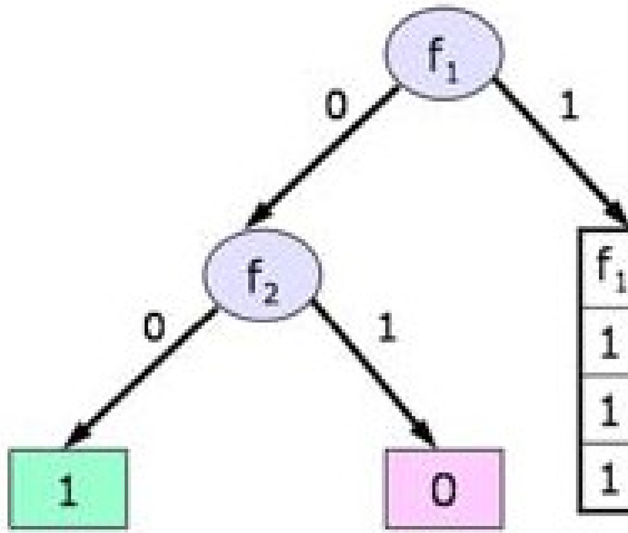
f_1	f_2	f_3	f_4	y
0	1	1	0	0
0	0	0	1	1
0	1	0	1	0

f_1	f_2	f_3	f_4	y
1	0	1	0	0
1	1	1	0	1
1	0	0	1	0

Exclusive OR

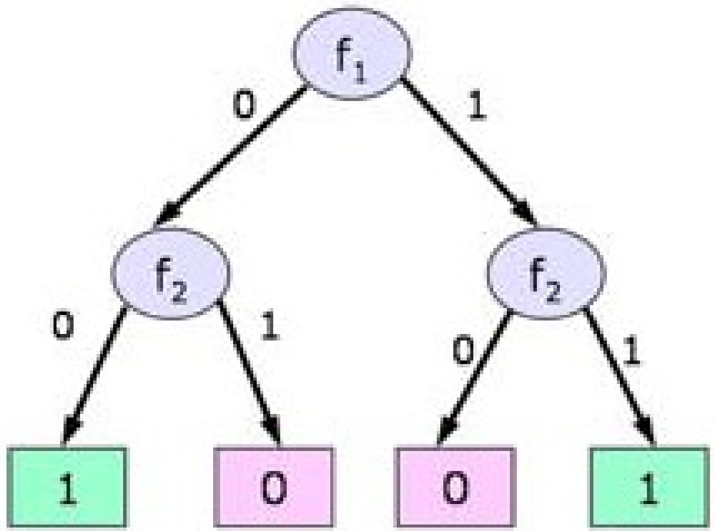


Exclusive OR



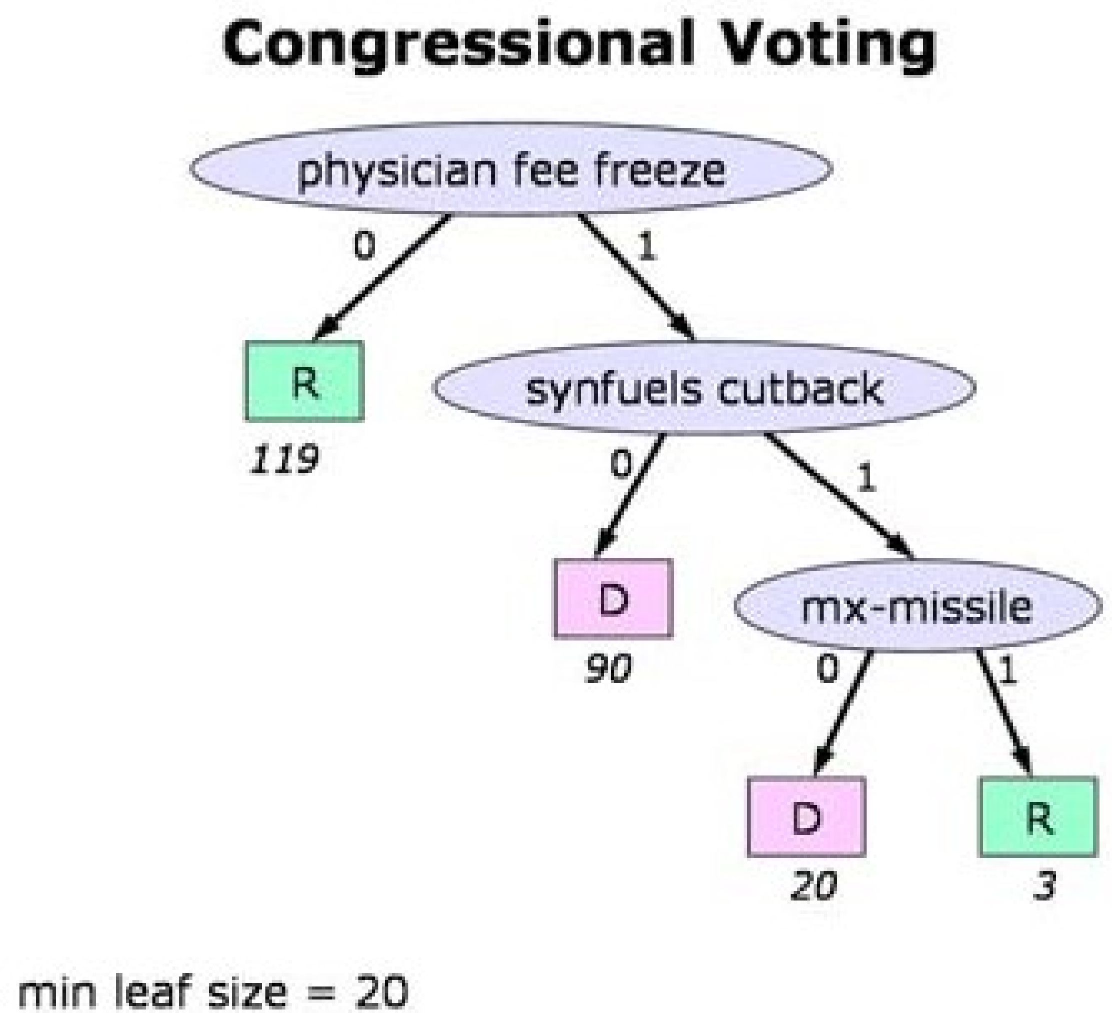
f_1	f_2	f_3	f_4	y
1	0	1	0	0
1	1	1	0	1
1	0	0	1	0

Exclusive OR



Pruning

- Best way to avoid overfitting and not get tricked by short-term lack of progress
  - Grow tree as far as possible
    - leaves are uniform or contain a single X
  - Prune the tree until it performs well on held-out data
  - Amount of pruning is like epsilon in the DNF algorithm


Congressional Voting
physician fee freeze
0
1
R
119
synfuels cutback
0
1
D
90
mx-missile
0
1
D
20
R
3
min leaf size = 20

Data Mining

- Making useful predictions in (usually corporate) applications
- Decision trees very popular because
  - easy to implement
  - efficient (even on huge data sets)
  - easy for humans to understand resulting hypotheses

Naive Bayes Algorithm

Naïve Bayes

- Founded on Bayes' rule for probabilistic inference
- Update probability of hypotheses based on evidence
- Choose hypothesis with the maximum probability after the evidence has been incorporated

Rev. Thomas Bayes

Naïve Bayes

- Founded on Bayes' rule for probabilistic inference
- Update probability of hypotheses based on evidence
- Choose hypothesis with the maximum probability after the evidence has been incorporated

Rev. Thomas Bayes

- Algorithm is particularly useful for domains with **lots** of features

Example

f_1	f_2	f_3	f_4	y
0	1	1	0	1
0	0	1	1	1
1	0	1	0	1
0	0	1	1	1
0	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	0	0	0	0
1	1	0	1	0
1	0	1	1	0

- $R_1(1,1)=1/5$: fraction of all **positive** examples that have feature 1 **on**
- $R_1(0,1)=4/5$: fraction of all **positive** examples that have feature 1 **off**

Example

f_1	f_2	f_3	f_4	y
0	1	1	0	1
0	0	1	1	1
1	0	1	0	1
0	0	1	1	1
0	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	0	0	0	0
1	1	0	1	0
1	0	1	1	0

- $R_1(1,1)=1/5$: fraction of all **positive** examples that have feature 1 **on**
- $R_1(0,1)=4/5$: fraction of all **positive** examples that have feature 1 **off**
- $R_1(1,0)=5/5$: fraction of all **negative** examples that have feature 1 **on**
- $R_1(0,0)=0/5$: fraction of all **negative** examples that have feature 1 **off**

Example

f_1	f_2	f_3	f_4	y
0	1	1	0	1
0	0	1	1	1
1	0	1	0	1
0	0	1	1	1
0	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	0	0	0	0
1	1	0	1	0
1	0	1	1	0

$R_1(1,1)=1/5$ $R_1(0,1)=4/5$
$R_1(1,0)=5/5$ $R_1(0,0)=0/5$

$R_2(1,1)=1/5$ $R_2(0,1)=4/5$
$R_2(1,0)=2/5$ $R_2(0,0)=3/5$

$R_3(1,1)=4/5$ $R_3(0,1)=1/5$
$R_3(1,0)=1/5$ $R_3(0,0)=4/5$

$R_4(1,1)=2/5$ $R_4(0,1)=3/5$
$R_4(1,0)=4/5$ $R_4(0,0)=1/5$

Prediction

$R_1(1,1)=1/5$	$R_1(0,1)=4/5$
$R_1(1,0)=5/5$	$R_1(0,0)=0/5$
$R_2(1,1)=1/5$	$R_2(0,1)=4/5$
$R_2(1,0)=2/5$	$R_2(0,0)=3/5$
$R_3(1,1)=4/5$	$R_3(0,1)=1/5$
$R_3(1,0)=1/5$	$R_3(0,0)=4/5$
$R_4(1,1)=2/5$	$R_4(0,1)=3/5$
$R_4(1,0)=4/5$	$R_4(0,0)=1/5$

Prediction

$R_1(1,1)=1/5$	$R_1(0,1)=4/5$
$R_1(1,0)=5/5$	$R_1(0,0)=0/5$
$R_2(1,1)=1/5$	$R_2(0,1)=4/5$
$R_2(1,0)=2/5$	$R_2(0,0)=3/5$
$R_3(1,1)=4/5$	$R_3(0,1)=1/5$
$R_3(1,0)=1/5$	$R_3(0,0)=4/5$
$R_4(1,1)=2/5$	$R_4(0,1)=3/5$
$R_4(1,0)=4/5$	$R_4(0,0)=1/5$

- New x = <0,0,1,1>
- $S(1) = R_1(0,1)*R_2(0,1)*R_3(1,1)*R_4(1,1) = .205$

Prediction

$R_1(1,1)=1/5$	$R_1(0,1)=4/5$
$R_1(1,0)=5/5$	$R_1(0,0)=0/5$
$R_2(1,1)=1/5$	$R_2(0,1)=4/5$
$R_2(1,0)=2/5$	$R_2(0,0)=3/5$
$R_3(1,1)=4/5$	$R_3(0,1)=1/5$
$R_3(1,0)=1/5$	$R_3(0,0)=4/5$
$R_4(1,1)=2/5$	$R_4(0,1)=3/5$
$R_4(1,0)=4/5$	$R_4(0,0)=1/5$

- New $x = \langle 0,0,1,1 \rangle$
- $S(1) = R_1(0,1)*R_2(0,1)*R_3(1,1)*R_4(1,1) = .205$
- $S(0) = R_1(0,0)*R_2(0,0)*R_3(1,0)*R_4(1,0) = 0$

Prediction

- $R_1(1,1)=1/5$ $R_1(0,1)=4/5$
- $R_1(1,0)=5/5$ $R_1(0,0)=0/5$
- $R_2(1,1)=1/5$ $R_2(0,1)=4/5$
- $R_2(1,0)=2/5$ $R_2(0,0)=3/5$
- $R_3(1,1)=4/5$ $R_3(0,1)=1/5$
- $R_3(1,0)=1/5$ $R_3(0,0)=4/5$
- $R_4(1,1)=2/5$ $R_4(0,1)=3/5$
- $R_4(1,0)=4/5$ $R_4(0,0)=1/5$

- New x = $<0,0,1,1>$
- $S(1) = R_1(0,1)*R_2(0,1)*R_3(1,1)*R_4(1,1) = .205$
- $S(0) = R_1(0,0)*R_2(0,0)*R_3(1,0)*R_4(1,0) = 0$
- $S(1) > S(0)$, so predict class 1

Learning Algorithm

- Estimate from the data, for all j:

$$R_j(1,1) = \frac{\#(x_j^i = 1 \wedge y^i = 1)}{\#(y^i = 1)}$$

Learning Algorithm

- Estimate from the data, for all j:

$$R_j(1,1) = \frac{\#(x_j^i = 1 \wedge y^i = 1)}{\#(y^i = 1)}$$

$$R_j(0,1) = 1 - R_j(1,1)$$

Learning Algorithm

- Estimate from the data, for all j:

$$R_j(1,1) = \frac{\#(x_j^i = 1 \wedge y^i = 1)}{\#(y^i = 1)}$$

$$R_j(0,1) = 1 - R_j(1,1)$$

$$R_j(1,0) = \frac{\#(x_j^i = 1 \wedge y^i = 0)}{\#(y^i = 0)}$$

$$R_j(0,0) = 1 - R_j(1,0)$$

Prediction Algorithm

- Given a new x,

$$S(1) = \prod_j \begin{cases} R_j(1,1) & \text{if } x_j = 1 \\ R_j(0,1) & \text{otherwise} \end{cases}$$

Prediction Algorithm

- Given a new x,

$$S(1) = \prod_j \begin{cases} R_j(1,1) & \text{if } x_j = 1 \\ R_j(0,1) & \text{otherwise} \end{cases}$$

$$S(0) = \prod_j \begin{cases} R_j(1,0) & \text{if } x_j = 1 \\ R_j(0,0) & \text{otherwise} \end{cases}$$

Prediction Algorithm

- Given a new x,

$$S(1) = \prod_j \begin{cases} R_j(1,1) & \text{if } x_j = 1 \\ R_j(0,1) & \text{otherwise} \end{cases}$$

$$S(0) = \prod_j \begin{cases} R_j(1,0) & \text{if } x_j = 1 \\ R_j(0,0) & \text{otherwise} \end{cases}$$

- Output 1 if $S(1) > S(0)$

Prediction Algorithm

- Given a new x,

$$\log S(1) = \sum_j \begin{cases} \log R_j(1,1) & \text{if } x_j = 1 \\ \log R_j(0,1) & \text{otherwise} \end{cases}$$

$$\log S(0) = \sum_j \begin{cases} \log R_j(1,0) & \text{if } x_j = 1 \\ \log R_j(0,0) & \text{otherwise} \end{cases}$$

- Output 1 if $\log S(1) > \log S(0)$

Better to add logs than to multiply small probabilities

Laplace Correction

- Avoid getting 0 or 1 as an answer:

$$R_j(1,1) = \frac{\#(x_j^i = 1 \wedge y^i = 1) + \mathbf{1}}{\#(y^i = 1) + \mathbf{2}}$$

$$R_j(0,1) = 1 - R_j(1,1)$$

$$R_j(1,0) = \frac{\#(x_j^i = 1 \wedge y^i = 0) + \mathbf{1}}{\#(y^i = 0) + \mathbf{2}}$$

$$R_j(0,0) = 1 - R_j(1,0)$$

Example with Correction

f_1	f_2	f_3	f_4	y
0	1	1	0	1
0	0	1	1	1
1	0	1	0	1
0	0	1	1	1
0	0	0	0	1
1	0	0	1	0
1	1	0	1	0
1	0	0	0	0
1	1	0	1	0
1	0	1	1	0

- $R_1(1,1)=2/7$ $R_1(0,1)=5/7$
- $R_1(1,0)=6/7$ **$R_1(0,0)=1/7$**
- $R_2(1,1)=2/7$ $R_2(0,1)=5/7$
- $R_2(1,0)=3/7$ $R_2(0,0)=4/7$
- $R_3(1,1)=5/7$ $R_3(0,1)=2/7$
- $R_3(1,0)=2/7$ $R_3(0,0)=5/7$
- $R_4(1,1)=3/7$ $R_4(0,1)=4/7$
- $R_4(1,0)=5/7$ $R_4(0,0)=2/7$

Prediction with Correction

- $R_1(1,1)=2/7$ $R_1(0,1)=5/7$
- $R_1(1,0)=6/7$ $R_1(0,0)=1/7$
- $R_2(1,1)=2/7$ $R_2(0,1)=5/7$
- $R_2(1,0)=3/7$ $R_2(0,0)=4/7$
- $R_3(1,1)=5/7$ $R_3(0,1)=2/7$
- $R_3(1,0)=2/7$ $R_3(0,0)=5/7$
- $R_4(1,1)=3/7$ $R_4(0,1)=4/7$
- $R_4(1,0)=5/7$ $R_4(0,0)=2/7$

- New x = $<0,0,1,1>$
- $S(1) = R_1(0,1)*R_2(0,1)*R_3(1,1)*R_4(1,1) = .156$
- $S(0) = R_1(0,0)*R_2(0,0)*R_3(1,0)*R_4(1,0) = .017$
- $S(1) > S(0)$, so predict class 1

Hypothesis Space

- Output 1 if

$$\prod_j \alpha_j x_j + (1-\alpha_j)(1-x_j) > \prod_j \beta_j x_j + (1-\beta_j)(1-x_j)$$

- Depends on parameters $\alpha_1 \ldots \alpha_n, \beta_1 \ldots \beta_n$ (which we set to be the R_j values)

Hypothesis Space

- Output 1 if

$$\prod_j \alpha_j x_j + (1-\alpha_j)(1-x_j) > \prod_j \beta_j x_j + (1-\beta_j)(1-x_j)$$

- Depends on parameters $\alpha_1 \ldots \alpha_n, \beta_1 \ldots \beta_n$ (which we set to be the R_j values)

- Our method of computing parameters doesn't minimize training set error, but it's fast!

Hypothesis Space

- Output 1 if

$$\prod_j \alpha_j x_j + (1-\alpha_j)(1-x_j) > \prod_j \beta_j x_j + (1-\beta_j)(1-x_j)$$

- Depends on parameters $\alpha_1 \ldots \alpha_n, \beta_1 \ldots \beta_n$ (which we set to be the R_j values)

- Our method of computing parameters doesn't minimize training set error, but it's fast!

- Weight of feature j's "vote" in favor of output 1:

$$\log \frac{\alpha_j}{1-\alpha_j} - \log \frac{\beta_j}{1-\beta_j}$$

Exclusive Or

f_1	f_2	f_3	f_4	y
0	1	1	0	0
1	0	1	0	0
1	0	0	1	0
0	1	0	1	0
1	1	1	0	1
0	0	0	1	1

Exclusive Or

f_1	f_2	f_3	f_4	y
0	1	1	0	0
1	0	1	0	0
1	0	0	1	0
0	1	0	1	0
1	1	1	0	1
0	0	0	1	1

- $R_1(1,1)=2/4$ $R_1(0,1)=2/4$
- $R_1(1,0)=3/6$ $R_1(0,0)=3/6$
- $R_2(1,1)=2/4$ $R_2(0,1)=2/4$
- $R_2(1,0)=3/6$ $R_2(0,0)=3/6$
- $R_3(1,1)=2/4$ $R_3(0,1)=2/4$
- $R_3(1,0)=3/6$ $R_3(0,0)=3/6$
- $R_4(1,1)=2/4$ $R_4(0,1)=2/4$
- $R_4(1,0)=3/6$ $R_4(0,0)=3/6$

Exclusive Or

f_1	f_2	f_3	f_4	y
0	1	1	0	0
1	0	1	0	0
1	0	0	1	0
0	1	0	1	0
1	1	1	0	1
0	0	0	1	1

- $R_1(1,1)=2/4$ $R_1(0,1)=2/4$
- $R_1(1,0)=3/6$ $R_1(0,0)=3/6$
- $R_2(1,1)=2/4$ $R_2(0,1)=2/4$
- $R_2(1,0)=3/6$ $R_2(0,0)=3/6$
- $R_3(1,1)=2/4$ $R_3(0,1)=2/4$
- $R_3(1,0)=3/6$ $R_3(0,0)=3/6$
- $R_4(1,1)=2/4$ $R_4(0,1)=2/4$
- $R_4(1,0)=3/6$ $R_4(0,0)=3/6$

- For any new x
- S(1) = .5 * .5 * .5 * .5 = .0625
- S(0) = .5 * .5 * .5 * .5 = .0625
- We're indifferent between classes

Congressional Voting

Congressional Voting

- Accuracy on the congressional voting domain is about 0.91
- Somewhat worse than decision trees (0.95)
- Decision trees can express more complex hypotheses over combinations of attributes

Congressional Voting

- Accuracy on the congressional voting domain is about 0.91
- Somewhat worse than decision trees (0.95)
- Decision trees can express more complex hypotheses over combinations of attributes
- Domain is small enough so that speed is not an issue
- So, prefer trees or DNF in this domain

Congressional Voting

-6.82	physician-fee-freeze
-4.20	el-salvador-aid
-4.20	crime
-3.56	education-spending
3.36	adoption-of-the-budget-resolution
3.25	aid-to-nicaraguan-contras
3.07	mx-missile
-2.51	superfund-right-to-sue
2.40	duty-free-exports
2.14	anti-satellite-test-ban
-2.07	religious-groups-in-schools
2.01	export-administration-act-south-africa
1.66	synfuels-corporation-cutback
1.63	handicapped-infants
-0.17	immigration
-0.08	water-project-cost-sharing

republican
democrat

Probabilistic Inference

Probabilistic Inference

- Think of features and output as random variables
- Learn $\Pr(Y = 1 | f_1, \ldots, f_n)$
- Given new example, compute probability it has value 1
- Generate answer 1 if that value is > 0.5, else 0

- Concentrate on estimating this distribution from data

$$\Pr(Y = 1 | f_1, \ldots, f_n)$$

Bayes' Rule

- Generically:

$$\Pr(A \mid B) = \Pr(B \mid A)\frac{\Pr(A)}{\Pr(B)}$$

Bayes' Rule

- Generically:

$$\Pr(A \mid B) = \Pr(B \mid A)\frac{\Pr(A)}{\Pr(B)}$$

- Specifically:

$$\Pr(Y = 1 \mid f_1 \ldots f_n) = \Pr(f_1 \ldots f_n \mid Y = 1)\frac{\Pr(Y = 1)}{\Pr(f_1 \ldots f_n)}$$

Bayes' Rule

- Generically:

$$\Pr(A \mid B) = \Pr(B \mid A)\frac{\Pr(A)}{\Pr(B)}$$

- Specifically:

$$\Pr(Y = 1 \mid f_1 \ldots f_n) = \Pr(f_1 \ldots f_n \mid Y = 1)\frac{\Pr(Y = 1)}{\underbrace{\Pr(f_1 \ldots f_n)}_{\text{independent of Y}}}$$

Bayes' Rule

- Generically:

$$\Pr(A \mid B) = \Pr(B \mid A)\frac{\Pr(A)}{\Pr(B)}$$

- Specifically:

$$\Pr(Y = 1 \mid f_1 \ldots f_n) = \Pr(f_1 \ldots f_n \mid Y = 1)\frac{\overbrace{\Pr(Y = 1)}^{\text{prior}}}{\underbrace{\Pr(f_1 \ldots f_n)}_{\text{independent of Y}}}$$

Bayes' Rule

- Generically:

$$\Pr(A \mid B) = \Pr(B \mid A)\frac{\Pr(A)}{\Pr(B)}$$

- Specifically:

$$\Pr(Y = 1 \mid f_1 \ldots f_n) = \Pr(f_1 \ldots f_n \mid Y = 1)\frac{\overbrace{\Pr(Y = 1)}^{\text{prior}}}{\underbrace{\Pr(f_1 \ldots f_n)}_{\text{independent of Y}}}$$

- Concentrate on:

$$\Pr(f_1 \ldots f_n \mid Y = 1)$$

Why is Bayes Naïve?

- Make a big independence assumption

$$\Pr(f_1 \ldots f_n \mid Y = 1) = \prod_j \Pr(f_j \mid Y = 1)$$

Learning Algorithm

- Estimate from the data, for all j:

$$R(f_j = 1 \mid Y = 1) = \frac{\#(x_j^i = 1 \wedge y^i = 1)}{\#(y^i = 1)}$$

$$R(f_j = 0 \mid Y = 1) = 1 - R(f_j = 1 \mid Y = 1)$$

$$R(f_j = 1 \mid Y = 0) = \frac{\#(x_j^i = 1 \wedge y^i = 0)}{\#(y^i = 0)}$$

$$R(f_j = 0 \mid Y = 0) = 1 - R(f_j = 1 \mid Y = 0)$$

Prediction Algorithm

- Given a new x,

$$S(x_1 \ldots x_n \mid Y = 1) = \prod_j \begin{cases} R(f_j = 1 \mid Y = 1) & \text{if } x_j = 1 \\ R(f_j = 0 \mid Y = 1) & \text{otherwise} \end{cases}$$

$$S(x_1 \ldots x_n \mid Y = 0) = \prod_j \begin{cases} R(f_j = 1 \mid Y = 0) & \text{if } x_j = 1 \\ R(f_j = 0 \mid Y = 0) & \text{otherwise} \end{cases}$$

- Output 1 if

$$S(x_1 \ldots x_n \mid Y = 1) > S(x_1 \ldots x_n \mid Y = 0)$$

References

- Flach, P. (2012). Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press
- Russell S., & Norvig, P. (2021). Artificial Intelligence A Modern Approach. 4th Edition: Pearson Education https://www.academia.edu/45126798/Artificial_Intelligence_A_Modern_Approach_4th_Edition_?auto=download
- Rich E., Knight K. & Nair, S. B. (2011). Artificial Intelligence 3rd Edition: Tata McGraw Hill
- Han, J.., Kamber, M., Pei, J (2012). Data mining: concepts and techniques.Elsevier Inc.
- Nilsson, N.J. (1998). Artificial Intelligence: a new synthesis. Morgan Kaufmann.
- Lucila Ohno-Machado, and Peter Szolovits. HST.947 Medical Artificial Intelligence. Spring 2005. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.

Thank You