Recap

- Second to last Tuesday
  - Machine Learning Experiment
  - Convert data set into correct format
  - Run machine learning workflow
    - `weka.classifiers.Evaluation`
    - `weka.classifiers.bayes.NaiveBayes`
- Last Tuesday
  - Our first ML algorithm: Naive Bayes
  - Based on $p(c|f)$ – probability of class c given feature values f
  - Naive: Assumes that features are independent of each other
    - This is (usually) not the case

Decision Trees

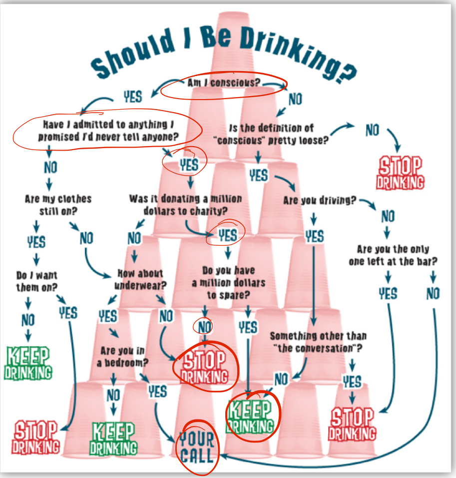
Sprachverarbeitung (VL + Ü)

Nils Reiter

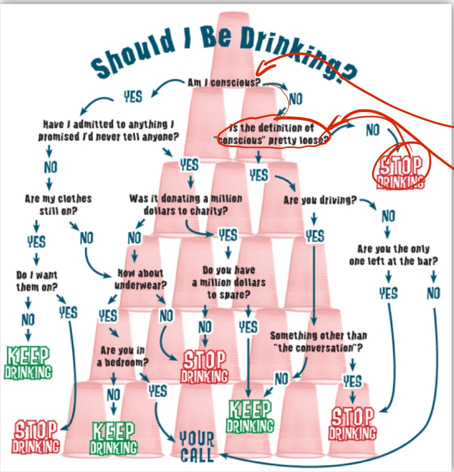
May 25, 2023

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Prediction Model – Toy Example



Prediction Model – Toy Example



- What are the instances?

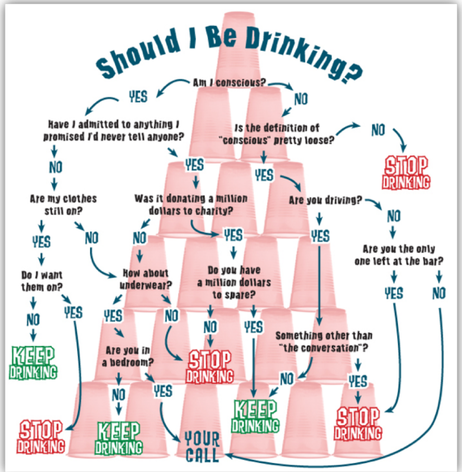
prediction(x):

1.

3. 'consciousness' => x : No

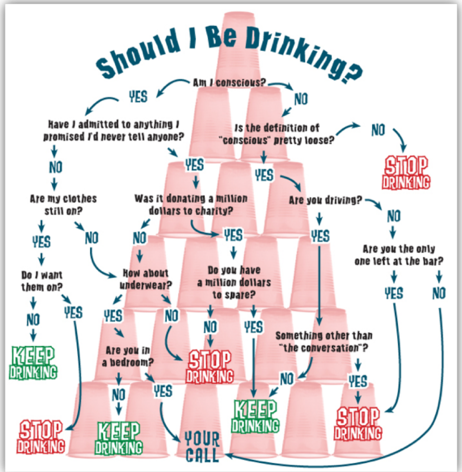
'defn of conscious' => x : No

Prediction Model – Toy Example



- What are the instances?
  - Situations we are in
    (this is not really automatisable)

Prediction Model – Toy Example



- What are the instances?
  - Situations we are in
    (this is not really automatisable)
- What are the features?

Prediction Model – Toy Example

- What are the instances?
  - Situations we are in (this is not really automatisable)
- What are the features?
  - Consciousness
  - Clothing situation
  - Promises made
  - Whether we are driving
  - ...

Conscio	Clothing	Driving	Class
+	+	+	STOP
−	+	−	Keep

Trees

- Well-established data structure in CS

S

NP VP

Der Hund schläft.

html

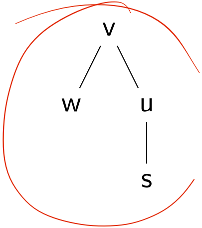
- head
- body
  - div
    - p

Trees

- Well-established data structure in CS
- A tree is a pair that contains
  - some value and
  - a (possibly empty) set of children
    - Children are also trees

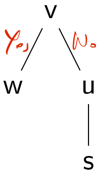
Trees

- Well-established data structure in CS
- A tree is a pair that contains
  - some value and
  - a (possibly empty) set of children
    - Children are also trees
- Recursive definition: "A tree is something and a bunch of sub trees"
  - Recursion is an important ingredient in many algorithms and data structures

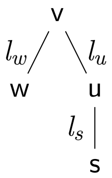


Trees

- Well-established data structure in CS
- A tree is a pair that contains
  - some value and
  - a (possibly empty) set of children
    - Children are also trees
- Recursive definition: “A tree is something and a bunch of sub trees”
  - Recursion is an important ingredient in many algorithms and data structures
- If the tree has labels on the edges, the pair becomes a triple

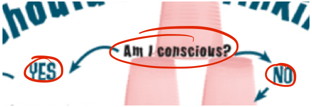


Trees

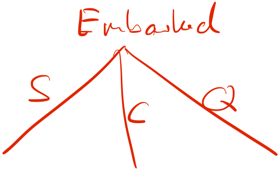


- Well-established data structure in CS
- A tree is a pair that contains
  - some value and
  - a (possibly empty) set of children
    - Children are also trees
- Recursive definition: "A tree is something and a bunch of sub trees"
  - Recursion is an important ingredient in many algorithms and data structures
- If the tree has labels on the edges, the pair becomes a triple

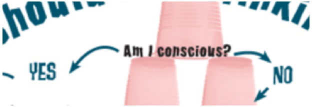
Prediction Model



- Each non-leaf node in the tree represents one feature
- Each leaf node represents a class label
- Each branch at this node represents one possible feature value
  - Number of branches $= |v(f_i)|$ (number of possible values)



Prediction Model



- Each non-leaf node in the tree represents one feature
- Each leaf node represents a class label
- Each branch at this node represents one possible feature value
  - Number of branches $= |v(f_i)|$ (number of possible values)
- Make a prediction for x:
  1. Start at root node
  2. If it's a leaf node
     - assign the class label
  3. Else
     - Check node which feature is to be tested (f_i)
     - Extract $f_i(x)$
     - Follow corresponding branch
     - Go to 2

$\underset{c \in C}{\operatorname{argmax}} \; p(c|F)$

Learning Algorithm

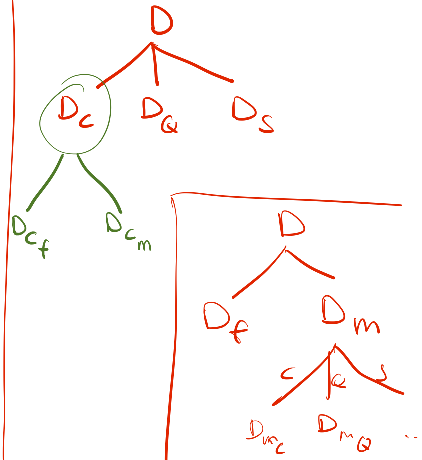
- Core idea: The tree represents splits of the training data

Learning Algorithm

C, Q, S

- Core idea: The tree represents splits of the training data
  1. Start with the full data set D_{train} as D
  2. If D only contains members of a single class:
     - Done.
  3. Else:
     - Select a feature f_i
     - Extract feature values of all instances in D
     - Split the data set according to f_i: $D = D_a \cup D_b \cup D_c \ldots$
       $D_\alpha = \{x \in D | f_i(x) = \alpha\}, \quad a, b, c \in v(f_i)$
     - Go back to 2

1. $D = D_{train}$
2. N.
3. f = Embarked



Learning Algorithm

- Core idea: The tree represents splits of the training data
  1. Start with the full data set D_{train} as D
  2. If D only contains members of a single class:
     - Done.
  3. Else:
     - Select a feature f_i
     - Extract feature values of all instances in D
     - Split the data set according to f_i: $D = D_a \cup D_b \cup D_c \ldots$
       $D_\alpha = \{x \in D | f_i(x) = \alpha\}, \quad a, b, c \in v(f_i)$
     - Go back to 2
- Remaining question: How to select features?

f_1	f_2	f_3	f_4	C
C	G	U	N	A
D	G	U	N	A
E	G	U	O	A
F	G	L	P	A
C	H	U	N	B
D	H	L	P	B
E	H	M	Q	B
F	H	M	Q	B

Feature Selection

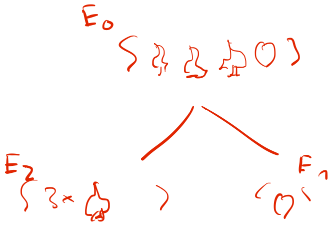
- What is a good feature?
  - One that maximizes homogeneity in the split data set

Feature Selection

- What is a good feature?
  - One that maximizes homogeneity in the split data set
- "Homogeneity"
  - Increase
    $\{\spadesuit\spadesuit\spadesuit\heartsuit\} = \{\heartsuit\} \cup \{\spadesuit\spadesuit\spadesuit\}$
  - No increase
    $\{\spadesuit\spadesuit\spadesuit\heartsuit\} = \{\spadesuit\} \cup \{\spadesuit\spadesuit\heartsuit\}$

$$IG = E_0 - (E_2 + E_1)/2$$

Feature Selection



- What is a good feature?
  - One that maximizes homogeneity in the split data set
- "Homogeneity"
  - Increase
    E_0 E_1 E_2
    $\{\spadesuit\spadesuit\spadesuit\heartsuit\} = \{\heartsuit\} \cup \{\spadesuit\spadesuit\spadesuit\}$ ← better split!
  - No increase
    $\{\spadesuit\spadesuit\spadesuit\heartsuit\} = \{\spadesuit\} \cup \{\spadesuit\spadesuit\heartsuit\}$
- Homogeneity: Entropy/information

Shannon (1948)

Feature Selection

- What is a good feature?
  - One that maximizes homogeneity in the split data set
- "Homogeneity"
  - Increase
    $\{\spadesuit\spadesuit\spadesuit\heartsuit\} = \{\heartsuit\} \cup \{\spadesuit\spadesuit\spadesuit\} \leftarrow$ better split!
  - No increase
    $\{\spadesuit\spadesuit\spadesuit\heartsuit\} = \{\spadesuit\} \cup \{\spadesuit\spadesuit\heartsuit\}$
- Homogeneity: Entropy/information Shannon (1948)
- Rule: Always select the feature with the highest *information gain* (IG)
  - (= the highest reduction in entropy = the highest increase in homogeneity)

Entropy

Intuition

- Measures the amount of uncertainty
- How uncertain is the next symbol in these sequences?
  - aaaaaaaaaaaaaa _ a

Entropy
Intuition

- Measures the amount of uncertainty
- How uncertain is the next symbol in these sequences?
  - aaaaaaaaaaaaaa – only one symbol, very certain

Entropy

Intuition

- Measures the amount of uncertainty
- How uncertain is the next symbol in these sequences?
  - aaaaaaaaaaaaa – only one symbol, very certain
  - abbaabbabbaaba

Entropy

Intuition

- Measures the amount of uncertainty
- How uncertain is the next symbol in these sequences?
  - aaaaaaaaaaaaaa – only one symbol, very certain
  - abbaabbabbaaba – two symbols, evenly distributed, 50:50

Entropy

Intuition

- Measures the amount of uncertainty
- How uncertain is the next symbol in these sequences?
  - aaaaaaaaaaaaaa – only one symbol, very certain
  - abbaabbabbaaba – two symbols, evenly distributed, 50:50
  - aaaaabbaaaaaba

Entropy

Intuition

- Measures the amount of uncertainty
- How uncertain is the next symbol in these sequences?
  - aaaaaaaaaaaaa – only one symbol, very certain
  - abbaabbabbaaba – two symbols, evenly distributed, 50:50
  - aaaaabbaaaaaba – two symbols, unevenly distributed, 75:25

Entropy

Intuition

- Measures the amount of uncertainty
- How uncertain is the next symbol in these sequences?
  - aaaaaaaaaaaaa – only one symbol, very certain
  - abbaabbabbaaba – two symbols, evenly distributed, 50:50
  - aaaaabbaaaaaba – two symbols, unevenly distributed, 75:25
  - cbabcababcbaca

Entropy

Intuition

- Measures the amount of uncertainty
- How uncertain is the next symbol in these sequences?
    - aaaaaaaaaaaaaa – only one symbol, very certain
    - abbaabbabbaaba – two symbols, evenly distributed, 50:50
    - aaaaabbaaaaaba – two symbols, unevenly distributed, 75:25
    - cbabcababcbaca – three symbols, evenly distributed, 33:66

Entropy

Intuition

- Measures the amount of uncertainty
- How uncertain is the next symbol in these sequences?
  - aaaaaaaaaaaaa – only one symbol, very certain
  - abbaabbabbaaba – two symbols, evenly distributed, 50:50
  - aaaaabbaaaaaba – two symbols, unevenly distributed, 75:25
  - cbabcababcbaca – three symbols, evenly distributed, 33:66
  - nmkfjigeahldcb

Entropy

Intuition

- Measures the amount of uncertainty
- How uncertain is the next symbol in these sequences?
  - aaaaaaaaaaaaa – only one symbol, very certain
  - abbaabbabbaaba – two symbols, evenly distributed, 50:50
  - aaaaabbaaaaaba – two symbols, unevenly distributed, 75:25
  - cbabcababcbaca – three symbols, evenly distributed, 33:66
  - nmkfjigeahldcb – 14 symbols, very uncertain
- Certainty depends on number of different symbols and on their distribution

Entropy (Shannon, 1948)

$$H(\{+,+,+,-\})$$

$$= -\Big(\frac{3}{4}\log_b \frac{3}{4} \qquad // +$$

$$+$$

$$\frac{1}{4}\log_b \frac{1}{4}\Big) \qquad // -$$

Anzahl Kategorial Featurewerte

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_b p(x_i)$$

Entropy (Shannon, 1948)

entropy of random variable X

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

Entropy (Shannon, 1948)

entropy of random variable X

number of classes present in X

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

Entropy (Shannon, 1948)

entropy of random variable X

number of classes present in X

relative frequency of the class

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

Entropy (Shannon, 1948)

entropy of random variable X

number of classes present in X

relative frequency of the class

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

$\log_b(x) = y$
exactly if
$b^y = x$:
$2^5 = 32 \Leftrightarrow \log_2 32 = 5$

Entropy (Shannon, 1948)

entropy of random variable X

number of classes present in X

relative frequency of the class

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

$\log_b(x) = y$
exactly if
$b^y = x$:
$2^5 = 32 \Leftrightarrow \log_2 32 = 5$

Interpretation

Entropy is the average number of bits* we need to specify an outcome of the random variable (* for $b = 2$)

Entropy (Shannon, 1948)

Examples

$$
\begin{aligned}
H(\{\spadesuit\spadesuit\spadesuit\spadesuit\}) &= -\frac{4}{4}\log_2\frac{4}{4} = 0 \\
H(\{\spadesuit\spadesuit\spadesuit\heartsuit\}) &= -\left(\underbrace{\frac{3}{4}\log_2\frac{3}{4}}_{\spadesuit} + \underbrace{\frac{1}{4}\log_2\frac{1}{4}}_{\heartsuit}\right) = 0.811 \\
H(\{\spadesuit\spadesuit\heartsuit\heartsuit\}) &= \ldots = 1 = H(\{\spadesuit\spadesuit\spadesuit\heartsuit\heartsuit\heartsuit\}) = \ldots
\end{aligned}
$$

Entropy (Shannon, 1948)

Examples

$$
\begin{aligned}
H(\{\spadesuit\spadesuit\spadesuit\spadesuit\}) &= -\frac{4}{4}\log_2\frac{4}{4} = 0 \\
H(\{\spadesuit\spadesuit\spadesuit\heartsuit\}) &= -\left(\underbrace{\frac{3}{4}\log_2\frac{3}{4}}_{\spadesuit} + \underbrace{\frac{1}{4}\log_2\frac{1}{4}}_{\heartsuit}\right) = 0.811 \\
H(\{\spadesuit\spadesuit\heartsuit\heartsuit\}) &= \ldots = 1 = H(\{\spadesuit\spadesuit\spadesuit\heartsuit\heartsuit\heartsuit\}) = \ldots \\
H(\{\spadesuit\spadesuit\heartsuit\heartsuit\clubsuit\clubsuit\}) &= 1.585 \\
H(\{\spadesuit\heartsuit\clubsuit\diamondsuit\}) &= 2 \\
H(\{nmkfjigeahldcb\}) &= 3.807
\end{aligned}
$$

April 14

Point-wise Mutual Information

MS99, pp. 178 ff.

- Point-wise: Statement about values of random variable (i.e., occurrence of specific word)
  - Non-pointwise mutual information makes a statement about random variables themselves
- Mutual: Symmetric
  - One word provides information to the next and vice versa

the dog

$$
\begin{aligned}
I(w_1, w_2) &= \log_2 \frac{p(w_1, w_2)}{p(w_1)p(w_2)} \\
p(w_i) &= \text{Probability of word } w_i \\
p(w_i, w_j) &= \text{Probability of both words appearing together, up to a certain distance} \\
\log_2 x = y &\equiv 2^y = x
\end{aligned}
$$

Entropy

Mutual Information

- Entropy: Amount of uncertainty in a random variable
  - Joint entropy: Amount of uncertainty in two random variables
  - Conditional entropy: Amount of uncertainty, when another random variable is known

Entropy

Mutual Information

- Entropy: Amount of uncertainty in a random variable
  - Joint entropy: Amount of uncertainty in two random variables
  - Conditional entropy: Amount of uncertainty, when another random variable is known
- Mutual Information
  - Reduction of entropy in one random variable by knowing about the other
  - $MI(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = \sum_{x,y} p(x, y) \log_2 \frac{p(x,y)}{p(x)p(y)}$

Entropy

Mutual Information

- Entropy: Amount of uncertainty in a random variable
  - Joint entropy: Amount of uncertainty in two random variables
  - Conditional entropy: Amount of uncertainty, when another random variable is known
- Mutual Information
  - Reduction of entropy in one random variable by knowing about the other
  - $MI(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = \sum_{x,y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}$
- Point-wise Mutual Information
  - Statement about values of random variable (i.e., occurrence of specific word)
  - $I(w_1, w_2) = \log_2 \frac{p(w_1,w_2)}{p(w_1)p(w_2)}$

Manning/Schütze, 1999, 67

Feature Selection

$$\{\spadesuit\spadesuit\spadesuit\heartsuit\}$$

$$\{\heartsuit\} \quad \{\spadesuit\spadesuit\spadesuit\}$$

$$\begin{aligned}
H(\{\spadesuit\spadesuit\spadesuit\heartsuit\}) &= H([3,1]) = 0.562 \\
H(\{\heartsuit\}) &= H([1]) = 0 \\
H(\{\spadesuit\spadesuit\spadesuit\}) &= H([3]) = 0
\end{aligned}$$

Feature Selection

$$\{\spadesuit\spadesuit\spadesuit\heartsuit\}$$
$$/ \quad \backslash$$
$$\{\heartsuit\} \; \{\spadesuit\spadesuit\spadesuit\}$$

$$\begin{aligned} H(\{\spadesuit\spadesuit\spadesuit\heartsuit\}) &= H([3,1]) = 0.562 \\ H(\{\heartsuit\}) &= H([1]) = 0 \\ H(\{\spadesuit\spadesuit\spadesuit\}) &= H([3]) = 0 \end{aligned}$$

$$\{\spadesuit\spadesuit\spadesuit\heartsuit\}$$
$$/ \quad \backslash$$
$$\{\spadesuit\} \; \{\spadesuit\spadesuit\heartsuit\}$$

$$\begin{aligned} H(\{\spadesuit\spadesuit\spadesuit\heartsuit\}) &= H([3,1]) = 0.562 \\ H(\{\spadesuit\}) &= H([1]) = 0 \\ H(\{\spadesuit\spadesuit\heartsuit\}) &= H([2,1]) = 0.637 \end{aligned}$$

Feature Selection

$$\{\spadesuit\spadesuit\spadesuit\heartsuit\}$$
$$/ \quad \backslash$$
$$\{\heartsuit\} \; \{\spadesuit\spadesuit\spadesuit\}$$

$$\{\spadesuit\spadesuit\spadesuit\heartsuit\}$$
$$/ \quad \backslash$$
$$\{\spadesuit\} \; \{\spadesuit\spadesuit\heartsuit\}$$

$$\begin{aligned}
H(\{\spadesuit\spadesuit\spadesuit\heartsuit\}) &= H([3,1]) = 0.562 \\
H(\{\heartsuit\}) &= H([1]) = 0 \\
H(\{\spadesuit\spadesuit\spadesuit\}) &= H([3]) = 0
\end{aligned}$$

$$\begin{aligned}
H(\{\spadesuit\spadesuit\spadesuit\heartsuit\}) &= H([3,1]) = 0.562 \\
H(\{\spadesuit\}) &= H([1]) = 0 \\
H(\{\spadesuit\spadesuit\heartsuit\}) &= H([2,1]) = 0.637
\end{aligned}$$

$$\begin{aligned}
IG(f_1) &= H(\{\spadesuit\spadesuit\spadesuit\heartsuit\}) - \varnothing\big(H(\{\heartsuit\}), H(\{\spadesuit\spadesuit\spadesuit\})\big) \\
&= 0.562 - 0 = 0.562 \\
IG(f_2) &= H(\{\spadesuit\spadesuit\spadesuit\heartsuit\}) - \varnothing\big(H(\{\spadesuit\}), H(\{\spadesuit\spadesuit\heartsuit\})\big) \\
&= 0.562 - (\frac{3}{4}0.637 + \frac{1}{4}0) \\
&= 0.562 - 0.562 - 0.477 = 0.085
\end{aligned}$$

Feature Selection using Entropy

- We calculate entropy for the target class
- But in different sub sets of the data set

Feature Selection using Entropy

- We calculate entropy for the target class
- But in different sub sets of the data set

Listing 2: Feature selection in pseudo code for a data set D

```
function select_feature(D):
  base_entropy = entropy(D)
  ig_map = {}
  foreach feature f:
    weighted_feature_entropy = 0
    foreach feature value v:
      D_v = subset of D with all instances that have the value v
      sub_entropy = entropy(D_v)
      sub_size = length(D_v)
      weighted_feature_entropy = weighted_feature_entropy + ( sub_entropy * sub_size )
    information_gain = base_entropy - ( (weighted_feature_entropy) / length(D) )
    ig_map.put(f, information_gain)
  return maximum from ig_map
```

ID3

J. Ross Quinlan (1986). »Induction of Decision Trees«. In: *Machine Learning* 1.1, pp. 81–106. DOI: 10.1007/BF00116251

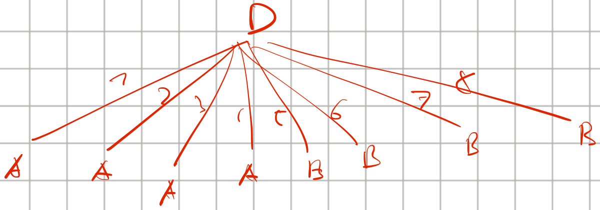
Limitations

- Only categorical attributes
- Cannot handle missing values
- Tends to overfit: »In my experience, almost all decision trees can benefit from simplification« (Quinlan, 1993, 36)
  - Even today, overfitting is a huge challenge for ML algorithms!

$\Rightarrow$ Extension: C4.5 (Quinlan, 1993)

Subsection 1

Example: Spam Classification



f_1	f_2	f_3	C
1	C	G	A
2	D	G	A
3	F	H	A
4	F	H	A
5	F	G	B
6	F	G	B
7	D	H	B
8	C	H	B

Data set (the same as last week)

- Data set: 100 e-mails, manually classified as spam or not spam (50/50)
  - Classes $C = \{\text{true}/1, \text{false}/0\}$
- Features: Presence of each of these tokens (manually selected): ›casino‹, ›enlargement‹, ›meeting‹, ›profit‹, ›super‹, ›text‹, ›xxx‹

Mail	›casino‹	›enlargement‹	›meeting‹	›profit‹	›super‹	›text‹	›xxx‹	C
1	1	1	0	0	1	1	1	0
2	0	1	0	1	0	0	0	1
3	1	0	1	0	1	0	0	0
4	1	1	1	0	0	0	0	0
5	0	1	1	0	0	1	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Learning Algorithm

First step: Use the full data set

$$H(\text{full data set}) = 1$$

Learning Algorithm

First step: Use the full data set

$$\begin{aligned} H(\text{full data set}) &= 1 \\ H(\text{›casino‹} = 1) &= 0.9991 \\ H(\text{›casino‹} = 0) &= 0.9985 \end{aligned}$$

Learning Algorithm

First step: Use the full data set

$$
\begin{aligned}
H(\text{full data set}) &= 1 \\
H(\text{›casino‹} = 1) &= 0.9991 \\
H(\text{›casino‹} = 0) &= 0.9985 \\
H(\text{›casino‹}) &= \frac{(56 \times 0.9991) + (44 \times 0.9985)}{100} = 0.9989 \\
IG(\text{›casino‹}) &= 1 - 0.9989 = 0.0012 \\
IG(\text{›profit‹}) &= 0.0073 \\
&\vdots
\end{aligned}
$$

Learning Algorithm

›profit‹

0 1

First step: Use the full data set

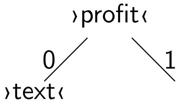
$$
\begin{aligned}
H(\text{full data set}) &= 1 \\
H(\text{›casino‹} = 1) &= 0.9991 \\
H(\text{›casino‹} = 0) &= 0.9985 \\
H(\text{›casino‹}) &= \frac{(56 \times 0.9991) + (44 \times 0.9985)}{100} = 0.9989 \\
IG(\text{›casino‹}) &= 1 - 0.9989 = 0.0012 \\
IG(\text{›profit‹}) &= 0.0073 \\
\vdots \quad &\quad \vdots
\end{aligned}
$$

Learning Algorithm

Next step: Use the data set *after* application of the first selected feature
›profit‹ = 0

$$
\begin{aligned}
H(\text{data set}) &= 0.99403 \\
H(\text{›casino‹} = 1) &= 0.9910 \\
H(\text{›casino‹} = 0) &= 0.9963 \\
IG(\text{›casino‹}) &= 0.00029 \\
IG(\text{›text‹}) &= 0.01151
\end{aligned}
$$

Learning Algorithm



Next step: Use the data set *after* application of the first selected feature

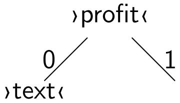
›profit‹ = 0

$$\begin{aligned} H(\text{data set}) &= 0.99403 \\ H(\text{›casino‹} = 1) &= 0.9910 \\ H(\text{›casino‹} = 0) &= 0.9963 \\ IG(\text{›casino‹}) &= 0.00029 \\ IG(\text{›text‹}) &= 0.01151 \end{aligned}$$

›profit‹ = 1

$$\begin{aligned} H(\text{data set}) &= 0.99107 \\ H(\text{›casino‹} = 1) &= 0.9366 \\ H(\text{›casino‹} = 0) &= 1 \\ IG(\text{›casino‹}) &= 0.0150 \\ IG(\text{›meeting‹}) &= 0.00029 \end{aligned}$$

Learning Algorithm



Next step: Use the data set *after* application of the first selected feature

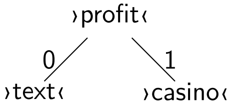
›profit‹ $= 0$

$$
\begin{aligned}
H(\text{data set}) &= 0.99403\\
H(\text{›casino‹} = 1) &= 0.9910\\
H(\text{›casino‹} = 0) &= 0.9963\\
IG(\text{›casino‹}) &= 0.00029\\
IG(\text{›text‹}) &= 0.01151
\end{aligned}
$$

›profit‹ $= 1$

$$
\begin{aligned}
H(\text{data set}) &= 0.99107\\
H(\text{›casino‹} = 1) &= 0.9366\\
H(\text{›casino‹} = 0) &= 1\\
IG(\text{›casino‹}) &= 0.0150\\
IG(\text{›meeting‹}) &= 0.00029
\end{aligned}
$$

Learning Algorithm



Next step: Use the data set *after* application of the first selected feature

›profit‹ = 0

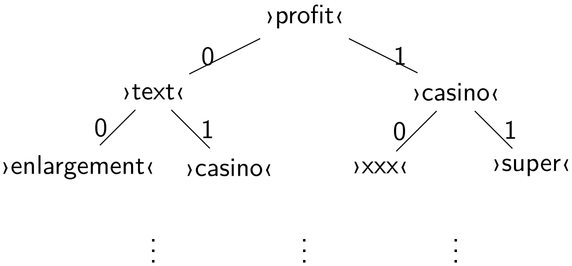
$$
\begin{aligned}
H(\text{data set}) &= 0.99403 \\
H(\text{›casino‹} = 1) &= 0.9910 \\
H(\text{›casino‹} = 0) &= 0.9963 \\
IG(\text{›casino‹}) &= 0.00029 \\
IG(\text{›text‹}) &= 0.01151
\end{aligned}
$$

›profit‹ = 1

$$
\begin{aligned}
H(\text{data set}) &= 0.99107 \\
H(\text{›casino‹} = 1) &= 0.9366 \\
H(\text{›casino‹} = 0) &= 1 \\
IG(\text{›casino‹}) &= 0.0150 \\
IG(\text{›meeting‹}) &= 0.00029
\end{aligned}
$$

Learning Algorithm

Next step: Use the data set *after* application of the first two layers of selected features



Section 1

Summary

Summary

- Naive Bayes in Weka
- Decision Tree
  - Transparent prediction model: Easy to apply by humans
  - Learning algorithm
    - Recursively split the training data set according to features
    - Use information gain to maximize the homogeneity in the sub sets
  - Compared with Naive Bayes
    - Feature dependence modeled through tree structure
  - DT in Weka: Try for yourselves! ☺