Recap

- Evaluation of machine learning models
- Accuracy, error rate
  - Single score for entire classification
- Precision, Recall, F-Score
  - Scores for each class
  - Precision: How many of the items classified as c are truly category c?
  - Recall: How many of the items that are truly c did the system find?
- Baseline

Das Data Center for the Humanities (DCH) an der Universität zu Köln sucht
zum nächstmöglichen Zeitpunkt eine

Studentische Hilfskraft (w/m/d)

(für bis zu 24 Monate, 19 Stunden/Woche)

Das Data Center for the Humanities (DCH) berät und unterstützt als geisteswissenschaftliches Datenzentrum an der Universität zu Köln Forschende der Philosophischen Fakultät im Bereich Forschungsdatenmanagement. Zur Unterstützung im Bereich Öffentlichkeitsarbeit, Veranstaltungsorganisation und Kommunikation sucht das DCH zum nächstmöglichen Zeitpunkt eine wissenschaftliche Hilfskraft (WHB).

Aufgaben

- Unterstützung und Zuarbeit innerhalb fachwissenschaftlicher und technischer Recherche
- Unterstützung bei der Organisation von internen und öffentlichen Workshops und Vernetzungsveranstaltungen
- Redaktionelle Arbeiten und Layout des Jahresberichts des DCH
- Design von wissenschaftlichen Postern und Awareness-Materialien im Forschungsdatenmanagement
- Unterstützung des DCH-Teams bei der technischen und inhaltlichen Betreuung von Websites

Notwendige Kenntnisse und Kompetenzen

- Geisteswissenschaftliches Studium, idealerweise mit starkem Forschungsschwerpunkt
- gute/sehr gute Deutsch- und Englischkenntnisse

Wünschenswerte Kenntnisse und Kompetenzen

- Interesse an Content Management Systemen (Typo3, Wordpress)
- Adobe InDesign- und Illustrator-Kenntnisse
- Erfahrung im Umgang mit digitalen Daten
- Erfahrung bei der Abfassung von wissenschaftlichen Texten
- Verständnis für Strukturen an Hochschulen
- Selbstsicherer Umgang mit Forscher:innen verschiedener Disziplinen

Bewerbungsfrist: 31.05.2024

Bewerbungen mit Lebenslauf sind elektronisch einzureichen bei patrick.helling@uni-koeln.de

Für weitere Informationen und Nachfragen wenden Sie sich bitte an Patrick Helling unter patrick.helling@uni-koeln.de

UNIVERSITÄT ZU KÖLN

Decision Tree

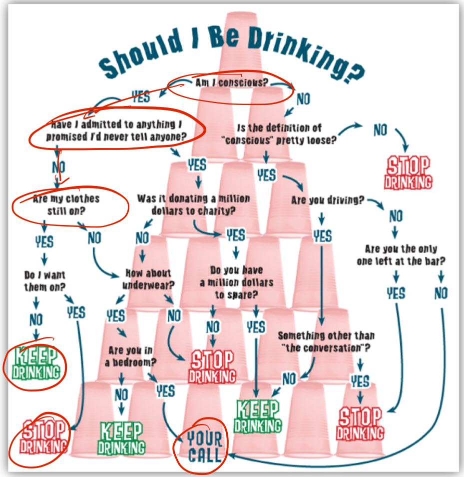
Sprachverarbeitung (VL + Ü)

Nils Reiter

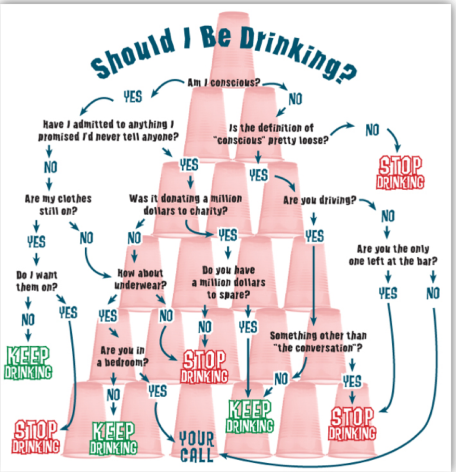
May 16, 2024

INSTITUT FÜR DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Prediction Model – Toy Example



Prediction Model – Toy Example

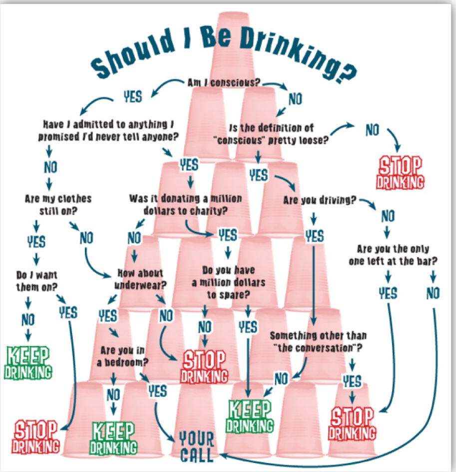


- What are the instances?

Prediction Model – Toy Example

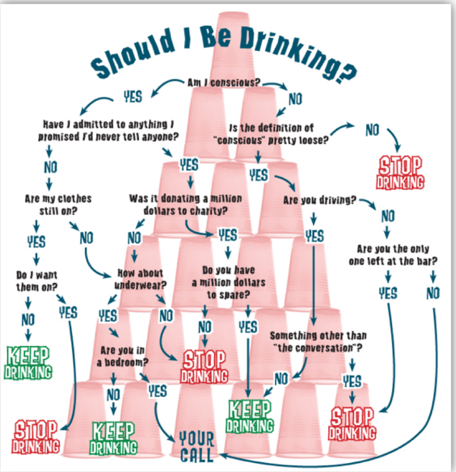
- What are the instances?
  - Situations we are in
    (this is not really automatisable)

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

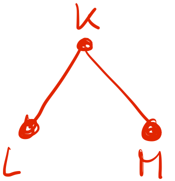


- What are the instances?
  - Situations we are in (this is not really automatisable)
- What are the features?
  - Consciousness
  - Clothing situation
  - Promises made
  - Whether we are driving
  - ...

Trees

- Well-established data structure in CS

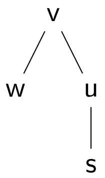
Trees



$$= \left(K, \left\{ (L, \emptyset), (M, \emptyset) \right\}\right)$$

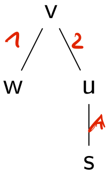
- Well-established data structure in CS
- A tree is a pair that contains
  - some value and
  - a (possibly empty) set of children
    - Children are also trees

Trees



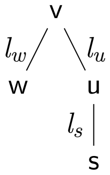
- Well-established data structure in CS
- A tree is a pair that contains
  - some value and
  - a (possibly empty) set of children
    - Children are also trees
- Recursive definition: "A tree is something and a bunch of sub trees"
  - Recursion is an important ingredient in many algorithms and data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

Trees



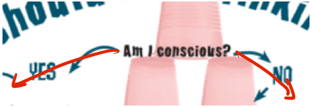
- Well-established data structure in CS
- A tree is a pair that contains
  - some value and
  - a (possibly empty) set of children
    - Children are also trees
- Recursive definition: "A tree is something and a bunch of sub trees"
  - Recursion is an important ingredient in many algorithms and data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

Trees

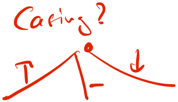


- Well-established data structure in CS
- A tree is a pair that contains
  - some value and
  - a (possibly empty) set of children
    - Children are also trees
- Recursive definition: “A tree is something and a bunch of sub trees”
  - Recursion is an important ingredient in many algorithms and data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

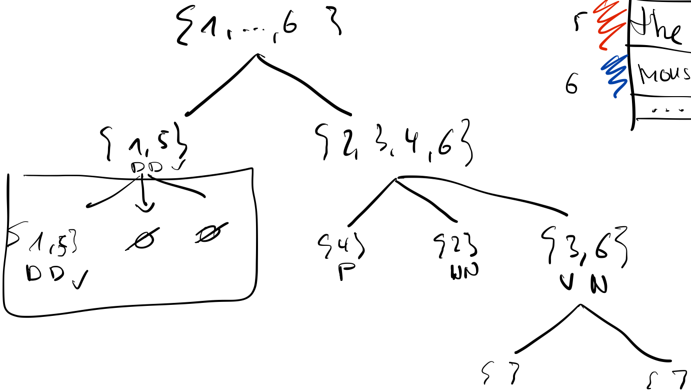
- Each non-leaf node in the tree represents one feature
- Each leaf node represents a class label
- Each branch at this node represents one possible feature value
  - Number of branches $= |v(f_i)|$ (number of possible values)
- Make a prediction for x:
  1. Start at root node
  2. If it's a leaf node
     - 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

KEEP DRINKING

Learning Algorithm

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

1	The	Det	↑	–	3
2	cat	NN	↓	the	3
3	sleeps	V	↓	cat	5
4	.	P	–	sleeps	1
5	The	D	↑	.	3
6	mouse	N	↓	the	5
	...	...			



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

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?

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
    $\{\spadesuit\spadesuit\spadesuit\heartsuit\} = \{\spadesuit\} \cup \{\spadesuit\spadesuit\heartsuit\}$

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)

Feature Selection

- What is a good feature?
  - One that maximizes homogeneity in the split data set
- "Homogeneity"
  - Increase
    $\{\spadesuit\spadesuit\spadesuit\heartsuit\} = \{\heartsuit\} \cup \{\spadesuit\spadesuit\spadesuit\}$ ← 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

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?
  - aaaaaaaaaaaaaa – 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:~~6~~6 33:33

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?
  - 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
  - nmkfjigeahldcb – 14 symbols, very uncertain
- Certainty depends on number of different symbols and on their distribution

Entropy (Shannon, 1948)

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

Entropy (Shannon, 1948)

Casing = { ↑ , ↓ , − }

{ ↑ ↑ ↓ ↓ − }

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

$$H(\{\spadesuit\spadesuit\spadesuit\spadesuit\}) = -\frac{4}{4}\log_2\left(\frac{4}{4}\right) = 0 \checkmark$$

$$H(\{\spadesuit\spadesuit\spadesuit\heartsuit\}) = -\left(\underbrace{\frac{3}{4}\log_2\left(\frac{3}{4}\right)}_{\spadesuit} + \underbrace{\frac{1}{4}\log_2\left(\frac{1}{4}\right)}_{\heartsuit}\right) = 0.811$$

$$H(\{\spadesuit\spadesuit\heartsuit\heartsuit\}) = \ldots = 1 = H(\{\spadesuit\spadesuit\spadesuit\heartsuit\heartsuit\heartsuit\}) = \ldots$$

Entropy (Shannon, 1948)

Examples

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

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

$$
\begin{array}{c}
\{\spadesuit\spadesuit\spadesuit\heartsuit\} \\
/ \quad \backslash \\
\{\heartsuit\}\ \{\spadesuit\spadesuit\spadesuit\}
\end{array}
$$

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

Feature Selection

f_1

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

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

$0 \quad 0$

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

f_2

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

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

$0 \quad 0{,}6$

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

Feature Selection

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

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

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

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

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

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
```

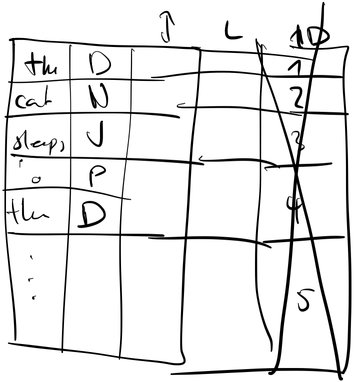
ID3

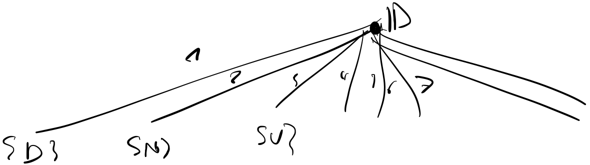
J. Ross Quinlan (1986). »Induction of Decision Trees«. In: *Machine Learning* 1.1, pp. 81–106. DOI: 10.1007/BF00116251

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)


the
D
cat
N
sleeps
V
.
P
the
D
J
L
ID
1
2
3
4
5


ID
1
2
3
4
5
6
7
{D}
{N}
{V}

Subsection 1

Example: Spam Classification

Data set

- 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
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

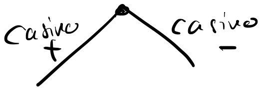
Learning Algorithm

First step: Use the full data set

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

Learning Algorithm

First step: Use the full data set



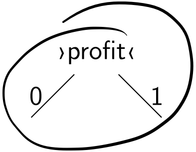
$$
\begin{aligned}
H(\text{full data set}) &= 1 \\
H(\text{›casino‹} = 1) &= 0.9991 \\
H(\text{›casino‹} = 0) &= 0.9985
\end{aligned}
$$

Learning Algorithm

First step: Use the full data set

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

Learning Algorithm



First step: Use the full data set

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

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

›profit‹

0 ›text‹ 1

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

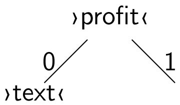
›profit‹ = 0

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

›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

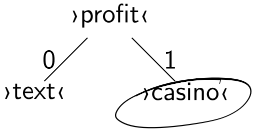
›profit‹ = 0

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

›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

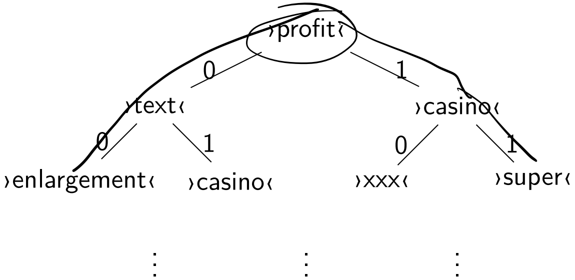
$$\begin{aligned} H(\text{data set}) &= 0.99403 \\ H(\text{›casino‹} = 1) &= 0.9910 \\ H(\text{›casino‹} = 0) &= 0.9963 \\ IG(\text{›casino‹}) &= 0.00029 \\ IG(\text{›text‹}) &= 0.01151 \end{aligned}$$

›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

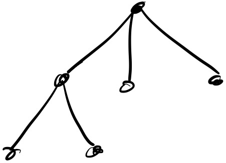
Next step: Use the data set *after* application of the first two layers of selected features



Section 1

Summary

Summary



- Decision Tree
  - Prediction model
    - Transparent: Easy to apply by humans
    - Easy to implement: Follow the path from root to leaf
  - 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! ☺

References I

- Manning, Christopher D./Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts and London, England: MIT Press.
- Quinlan, J. Ross (1986). »Induction of Decision Trees«. In: *Machine Learning* 1.1, pp. 81–106. DOI: 10.1007/BF00116251.
- — (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- Shannon, Claude E. (1948). »A mathematical theory of communication«. In: *The Bell System Technical Journal* 27.3, pp. 379–423.