Machine Learning and Pervasive Computing

Stephan Sigg

Georg-August-University Goettingen, Computer Networks

26.11.2014

Overview and Structure

22.10.2014	Organisation
22.10.3014	Introduction (Def.: Machine learning, Supervised/Unsupervised, Examples)
29.10.2014	Machine Learning Basics (Toolchain, Features, Metrics, Rule-based)
05.11.2014	A simple Supervised learning algorithm
12.11.2014	Excursion: Avoiding local optima with random search
19.11.2014	–
26.11.2014	Bayesian learner
03.12.2014	–
10.12.2014	Decision tree learner
17.12.2014	k-nearest neighbour
07.01.2015	Support Vector Machines
14.01.2015	Artificial Neural networks and Self Organizing Maps
21.01.2015	Hidden Markov models and Conditional random fields
28.01.2015	High dimensional data, Unsupervised learning
04.02.2015	Anomaly detection, Online learning, Recom. systems

Outline

Bayesian decision theory

With probability theory, the probability of events can be estimated by repeatedly generating events and counting their occurrences

When, however, an event only very seldom occurs or is hard to generate, other methods are required

Example:

Probability that the Arctic ice cap will have disappeared by the end of this century

In such cases, we would like to model uncertainty

In fact, it is possible to represent uncertainty by probability

Conditional probability

Conditional probability

The conditional probability of two events χ_1 and χ_2 with $P(\chi_2) > 0$ is denoted by $P(\chi_1|\chi_2)$ and is calculated by

$$P(\chi_1|\chi_2) = \frac{P(\chi_1 \cap \chi_2)}{P(\chi_2)}$$

$P(\chi_1|\chi_2)$ describes the probability that event χ_2 occurs in the presence of event χ_2.

Bayesian decision theory

With the notion of conditional probability we can express the effect of observed data $\overrightarrow{a} = a_1, \ldots, a_N$ on a probability distribution of $\overrightarrow{b}$: $P(\overrightarrow{b})$.

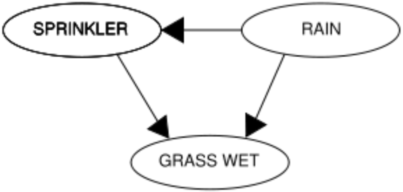
Thomas Bayes described a way to evaluate the uncertainty of $\overrightarrow{b}$ after observing $\overrightarrow{a}$

$$P(\overrightarrow{b}|\overrightarrow{a}) = \frac{P(\overrightarrow{a}|\overrightarrow{b})P(\overrightarrow{b})}{P(\overrightarrow{a})}$$

$P(\overrightarrow{a}|\overrightarrow{b})$ expresses how probable a value for $\overrightarrow{a}$ is given a fixed choice of $\overrightarrow{b}$

Example

RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99



RAIN T	RAIN F
0.2	0.8

SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Bayesian curve fitting

In the classification problems considered before, we were given $\overrightarrow{x}$ and $\overrightarrow{y}$ together with a new sample x_{M+1}

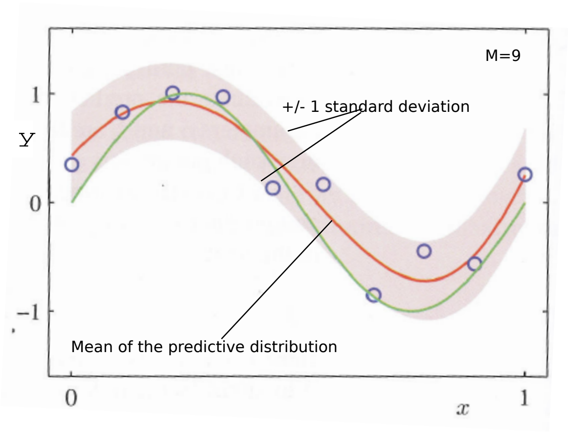
The task is to find a good estimation of the value y_{M+1}

This means that we want to evaluate the predictive distribution

$$p(y_{M+1}|x_{M+1}, \overrightarrow{x}, \overrightarrow{y})$$

To account for measurement inaccuracies, typically a probability distribution (e.g. Gauss) is underlying the sample vector $\overrightarrow{x}$

Bayesian curve fitting



Outline

Naïve Bayes classificaiton

WiFi			Accelerometer			Audio			Light			At work	
	yes	no		yes	no		yes	no		yes	no	yes	no
<3 APs	3	7	walking	4	8	quiet	8	5	outdoor	4	7	16	14
[3, 5]	5	5	standing	1	4	medium	6	3	indoor	12	7		
>5 APs	8	2	sitting	11	2	loud	2	6					

Naïve Bayes classificaiton

WiFi	yes	no	Accelerometer	yes	no	Audio	yes	no	Light	yes	no	At work yes	At work no
<3 APs	3	7	walking	4	8	quiet	8	5	outdoor	4	7	16	14
[3, 5]	5	5	standing	1	4	medium	6	3	indoor	12	7		
>5 APs	8	2	sitting	11	2	loud	2	6					

WiFi	Accelerometer	Audio	Light	At work
4 APs	sitting	medium	indoors	**???**

Likelihood of YES:

Likelihood of NO:

Naïve Bayes classificaiton

WiFi	yes	no	Accelerometer	yes	no	Audio	yes	no	Light	yes	no	At work yes	At work no
<3 APs	3	7	walking	4	8	quiet	8	5	outdoor	4	7	16	14
[3, 5]	5	5	standing	1	4	medium	6	3	indoor	12	7		
>5 APs	8	2	sitting	11	2	loud	2	6					

WiFi	Accelerometer	Audio	Light	At work
4 APs	sitting	medium	indoors	**???**

Likelihood of YES: $\frac{5}{16} \cdot \frac{11}{16} \cdot \frac{6}{16} \cdot \frac{12}{16} \cdot \frac{16}{30} = 0.032$

Likelihood of NO: $\frac{5}{14} \cdot \frac{2}{14} \cdot \frac{3}{14} \cdot \frac{7}{14} \cdot \frac{14}{30} = 0.0026$

Naïve Bayes classificaiton

WiFi	Accelerometer	Audio	Light	At work
4 APs	sitting	medium	indoors	**???**

Likelihood of YES: $\frac{5}{16} \cdot \frac{11}{16} \cdot \frac{6}{16} \cdot \frac{12}{16} \cdot \frac{16}{30} = 0.032$

Likelihood of NO: $\frac{5}{14} \cdot \frac{2}{14} \cdot \frac{3}{14} \cdot \frac{7}{14} \cdot \frac{14}{30} = 0.0026$

Probability of YES:

Probability of NO:

Naïve Bayes classificaiton

WiFi	Accelerometer	Audio	Light	At work
4 APs	sitting	medium	indoors	**???**

Likelihood of YES: $\frac{5}{16} \cdot \frac{11}{16} \cdot \frac{6}{16} \cdot \frac{12}{16} \cdot \frac{16}{30} = 0.032$

Likelihood of NO: $\frac{5}{14} \cdot \frac{2}{14} \cdot \frac{3}{14} \cdot \frac{7}{14} \cdot \frac{14}{30} = 0.0026$

Probability of YES: $\frac{0.032}{0.032+0.0026} \approx 0.925$

Probability of NO: $\frac{0.0026}{0.0026+0.032} \approx 0.075$

Naïve Bayes classificaiton

Likelihood of YES: $\frac{5}{16} \cdot \frac{11}{16} \cdot \frac{6}{16} \cdot \frac{12}{16} \cdot \frac{16}{30} = 0.032$

Likelihood of NO: $\frac{5}{14} \cdot \frac{2}{14} \cdot \frac{3}{14} \cdot \frac{7}{14} \cdot \frac{14}{30} = 0.0026$

Probability of YES: $\frac{0.032}{0.032+0.0026} \approx 0.925$

Probability of NO: $\frac{0.0026}{0.0026+0.032} \approx 0.075$

This is due to bayes rule:

$$\mathcal{P}[\text{Hypothesis}|\text{Evidence}] = \frac{\mathcal{P}[\text{Evidence}|\text{Hypothesis}]\mathcal{P}[\text{Hypothesis}]}{\mathcal{P}[\text{Evidence}]}$$

Naïve Bayes classificaiton

Likelihood of YES: $\frac{5}{16} \cdot \frac{11}{16} \cdot \frac{6}{16} \cdot \frac{12}{16} \cdot \frac{16}{30} = 0.032$

Likelihood of NO: $\frac{5}{14} \cdot \frac{2}{14} \cdot \frac{3}{14} \cdot \frac{7}{14} \cdot \frac{14}{30} = 0.0026$

Probability of YES: $\frac{0.032}{0.032+0.0026} \approx 0.925$

Probability of NO: $\frac{0.0026}{0.0026+0.032} \approx 0.075$

This is due to bayes rule:

$$\mathcal{P}[\text{Hypothesis}|\text{Evidence}] = \frac{\mathcal{P}[\text{Evidence}|\text{Hypothesis}]\mathcal{P}[\text{Hypothesis}]}{\mathcal{P}[\text{Evidence}]}$$

$$\mathcal{P}[\text{work}|\text{Evidence}] = \frac{\mathcal{P}[\text{E}_1|\text{work}]\mathcal{P}[\text{E}_2|\text{work}]\mathcal{P}[\text{E}_3|\text{work}]\mathcal{P}[\text{E}_4|\text{work}]\mathcal{P}[\text{work} = \text{YES}]}{\mathcal{P}[\text{Evidence}]}$$

$$\mathcal{P}[\text{work}|\text{E}] = \frac{\mathcal{P}[\text{5 APs}|\text{work}]\mathcal{P}[\text{sitting}|\text{work}]\mathcal{P}[\text{medium}|\text{work}]\mathcal{P}[\text{indoors}|\text{work}]\mathcal{P}[\text{work}]}{\mathcal{P}[\text{Evidence}]}$$

Naïve Bayes classificaiton

The name Naïve Bayes stems from the fact that

1. the method is based on Bayes' rule
2. it naïvely assumes independence among events

Note that it is only valid to multiply probabilities given the class when the events are independent.

Naïve Bayes classificaiton

The name Naïve Bayes stems from the fact that

1. the method is based on Bayes' rule
2. it naïvely assumes independence among events

Note that it is only valid to multiply probabilities given the class when the events are independent.

However, even though the latter assumption is unrealistic in real settings, the performance of Naïve Bayes on real data is good.

Naïve Bayes classificaiton

Be careful with impossible events!

In the case that an attribute value does not occur in the training set in conjuction with every class value:

Assume: Walking always associated with 'NO'
($\rightarrow \mathcal{P}[\text{walking}|\text{yes}] = 0$)

Then: $\mathcal{P}[\text{yes}|\text{E}] = 0$

Naïve Bayes classificaiton

Solution (Laplace estimator)

Add small constant $\frac{\mu}{n}$ to all numerators and compensate by adding μ to each of the n denominators:

$$\frac{5}{16} \cdot \frac{11}{16} \cdot \frac{6}{16} \cdot \frac{12}{16}$$

$$\rightarrow \frac{5+\frac{\mu}{4}}{16+\mu} \cdot \frac{11+\frac{\mu}{4}}{16+\mu} \cdot \frac{6+\frac{\mu}{4}}{16+\mu} \cdot \frac{12+\frac{\mu}{4}}{16+\mu}$$

In practice, these small modifications make little difference given that there are sufficient training examples.

Naïve Bayes classificaiton

Example (Laplace estimator)

Add 1 to all numerators and compensate by adding 4 to each of the 4 denominators:

$$\begin{aligned} & \frac{5}{16} \cdot \frac{11}{16} \cdot \frac{6}{16} \cdot \frac{12}{16} \\ \rightarrow & \frac{6}{20} \cdot \frac{12}{20} \cdot \frac{7}{20} \cdot \frac{16}{20} \end{aligned}$$

In practice, these small modifications make little difference given

Naïve Bayes classificaiton

Example (Laplace estimator)

Add 1 to all numerators and compensate by adding 4 to each of the 4 denominators:

$$\frac{5}{16} \cdot \frac{11}{16} \cdot \frac{6}{16} \cdot \frac{12}{16}$$
$$\rightarrow \frac{6}{20} \cdot \frac{12}{20} \cdot \frac{7}{20} \cdot \frac{16}{20}$$

Likelihood of YES: $\frac{6}{20} \cdot \frac{12}{20} \cdot \frac{7}{20} \cdot \frac{16}{20} \cdot \frac{16}{30} = 0.022$

Likelihood of NO: $\frac{6}{18} \cdot \frac{3}{18} \cdot \frac{4}{18} \cdot \frac{8}{18} \cdot \frac{14}{30} = 0.0026$

In practice, these small modifications make little difference given

Naïve Bayes classificaiton

Example (Laplace estimator)

Add 1 to all numerators and compensate by adding 4 to each of the 4 denominators:

$$\frac{5}{16} \cdot \frac{11}{16} \cdot \frac{6}{16} \cdot \frac{12}{16}$$
$$\rightarrow \frac{6}{20} \cdot \frac{12}{20} \cdot \frac{7}{20} \cdot \frac{16}{20}$$

Likelihood of YES: $\frac{6}{20} \cdot \frac{12}{20} \cdot \frac{7}{20} \cdot \frac{16}{20} \cdot \frac{16}{30} = 0.022$

Likelihood of NO: $\frac{6}{18} \cdot \frac{3}{18} \cdot \frac{4}{18} \cdot \frac{8}{18} \cdot \frac{14}{30} = 0.0026$

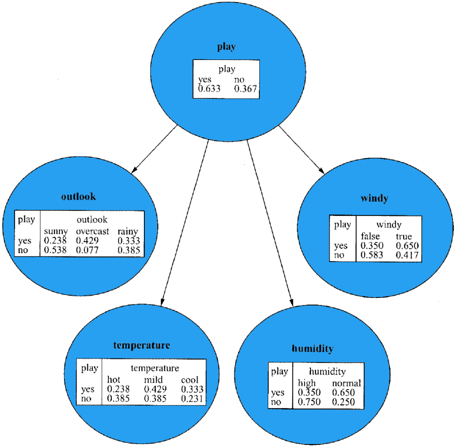
Probability of YES: $\frac{0.022}{0.022+0.0026} \approx 0.894$

Probability of NO: $\frac{0.0026}{0.0026+0.022} \approx 0.105$

In practice, these small modifications make little difference given

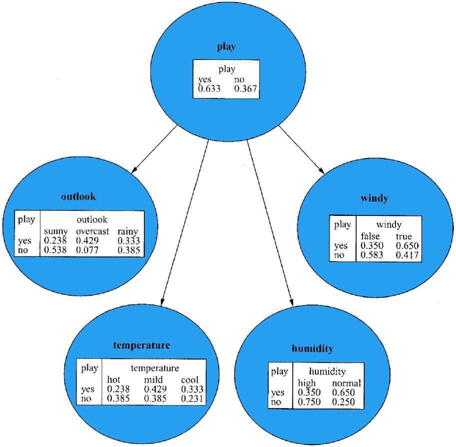
Outline

Bayesian Networks



Bayesian Networks

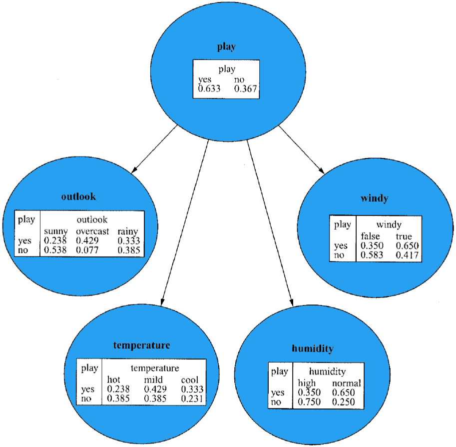
Concise and theoretically well founded way of representing probability distributions in a graphical manner



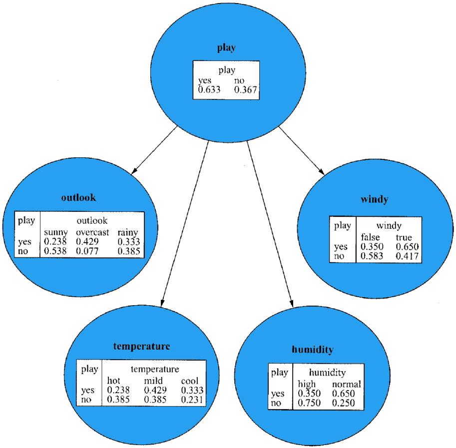
Bayesian Networks

Concise and theoretically well founded way of representing probability distributions in a graphical manner

Directed acyclic Graph with one vertex for each feature or class

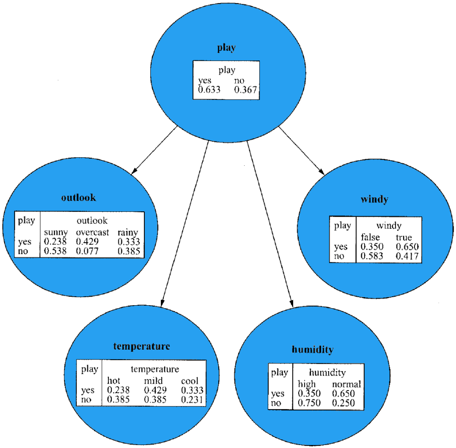


Left side of the distribution table in each node contains a column for every ingoing edge from a parent node

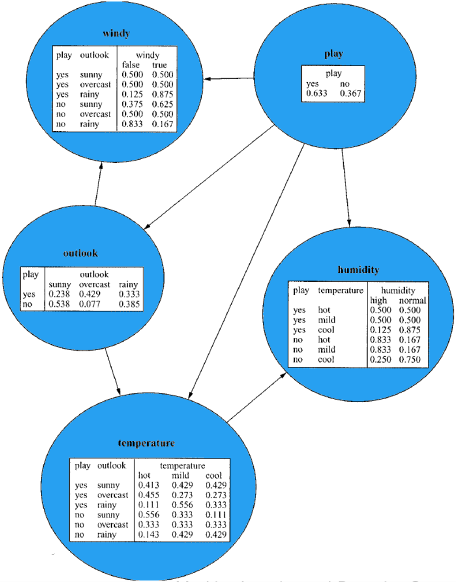


Left side of the distribution table in each node contains a column for every ingoing edge from a parent node

Each row defines a probability distribution over the values of a node's attribute

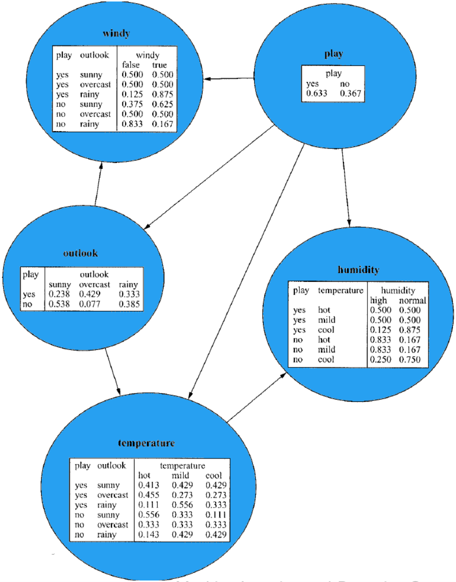


Prediction of class probabilities

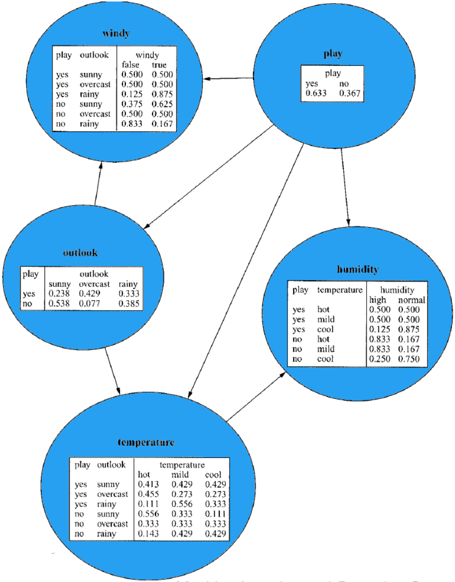


Prediction of class probabilities

For a particular sample, multiply all corresponding probabilities

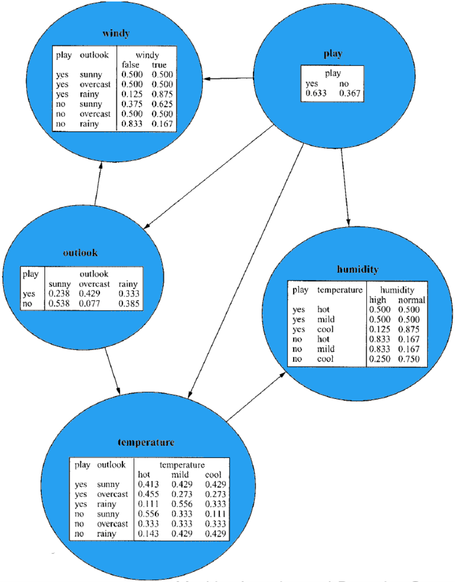


Example


windy
play outlook windy
false true
yes sunny 0.500 0.500
yes overcast 0.500 0.500
yes rainy 0.125 0.875
no sunny 0.375 0.625
no overcast 0.500 0.500
no rainy 0.833 0.167
play
play
yes no
0.633 0.367
outlook
play outlook
sunny overcast rainy
yes 0.238 0.429 0.333
no 0.538 0.077 0.385
humidity
play temperature humidity
high normal
yes hot 0.500 0.500
yes mild 0.500 0.500
yes cool 0.125 0.875
no hot 0.833 0.167
no mild 0.833 0.167
no cool 0.250 0.750
temperature
play outlook temperature
hot mild cool
yes sunny 0.413 0.429 0.429
yes overcast 0.455 0.273 0.273
yes rainy 0.111 0.556 0.333
no sunny 0.556 0.333 0.111
no overcast 0.333 0.333 0.333
no rainy 0.143 0.429 0.429

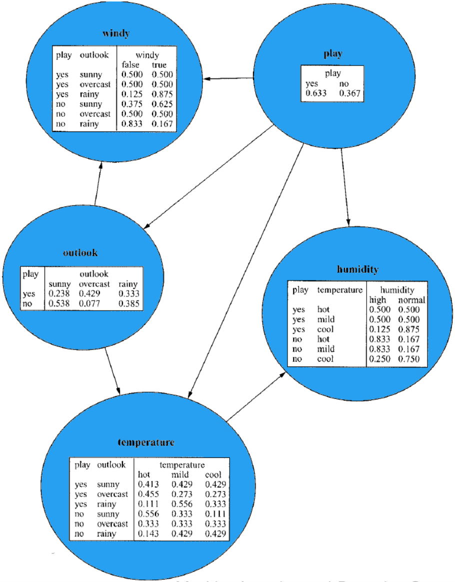
Example

outlook	rainy
temperature	cool
humidity	high
windy	true



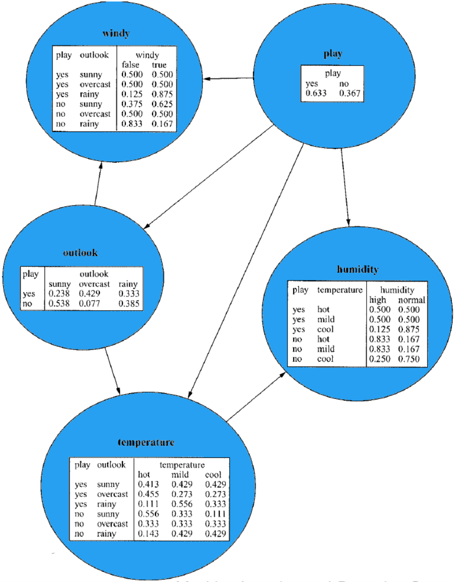
Example

outlook	rainy
temperature	cool
humidity	high
windy	true
play = no	$0.367 \cdot 0.167 \cdot 0.385 \cdot 0.25 \cdot 0.429 = 0.0025$



Example

outlook	rainy
temperature	cool
humidity	high
windy	true
play = no	$0.367 \cdot 0.167 \cdot 0.385 \cdot 0.25 \cdot 0.429 = 0.0025$
play = yes	$= 0.0077$



Example

$$\text{play} = \text{no} \quad 0.367 \cdot 0.167 \cdot 0.385 \cdot 0.25 \cdot 0.429 = 0.0025$$

$$\text{play} = \text{yes} \quad = 0.0077$$

Example

$$\text{play} = \text{no} \quad 0.367 \cdot 0.167 \cdot 0.385 \cdot 0.25 \cdot 0.429 = 0.0025$$

$$\text{play} = \text{yes} \quad = 0.0077$$

$$\mathcal{P}[\text{play} = \text{no}] \quad \frac{0.0025}{0.367+0.167+0.385+0.25+0.429} = 0.245$$

Example

$$\text{play} = \text{no} \quad 0.367 \cdot 0.167 \cdot 0.385 \cdot 0.25 \cdot 0.429 = 0.0025$$

$$\text{play} = \text{yes} \quad = 0.0077$$

$$\mathcal{P}[\text{play} = \text{no}] \quad \frac{0.0025}{0.367+0.167+0.385+0.25+0.429} = 0.245$$

$$\mathcal{P}[\text{play} = \text{yes}] \quad \frac{0.0077}{0.875+0.333+0.111+0.5+0.633} = 0.755$$

Example

$$\text{play} = \text{no} \quad 0.367 \cdot 0.167 \cdot 0.385 \cdot 0.25 \cdot 0.429 = 0.0025$$

$$\text{play} = \text{yes} \quad = 0.0077$$

$$\mathcal{P}[\text{play} = \text{no}] \quad \frac{0.0025}{0.367+0.167+0.385+0.25+0.429} = 0.245$$

$$\mathcal{P}[\text{play} = \text{yes}] \quad \frac{0.0077}{0.875+0.333+0.111+0.5+0.633} = 0.755$$

Remark Multiplication of all probabilities is valid due to conditional independence: Multiplication is valid provided that each node is independent from parents

Conditional indepencence

Multiplication follows result of chain rule in probability theory (joint probability of m variables can be decomposed into its product):

$$\mathcal{P}[a_1, a_2, \ldots, a_n] = \prod_{i=1}^{n} \mathcal{P}[a_i | a_{i-1}, \ldots, a_1]$$

Since the Bayesian network is an acyclic graph, nodes can be ordered to give all ancestors of a node a_i indices smaller than i
Then, due to conditional indepencence:

$$\mathcal{P}[a_1, a_2, \ldots, a_n] = \prod_{i=1}^{n} \mathcal{P}[a_i | a_{i-1}, \ldots, a_1] = \prod_{i=1}^{n} \mathcal{P}[a_i | a_i\text{'s parents}]$$

Learning Bayesian Networks

In order to learn/train a Bayesian network we require

1. A function to evaluate a given network
2. A method to search through the space of possible networks

Learning Bayesian Networks

In order to learn/train a Bayesian network we require

1. A function to evaluate a given network
2. A method to search through the space of possible networks

Evaluate a given network

Learning Bayesian Networks

In order to learn/train a Bayesian network we require

1. A function to evaluate a given network
2. A method to search through the space of possible networks

Evaluate a given network

Probability assigned to given instance is multiplied over all instances.

Learning Bayesian Networks

In order to learn/train a Bayesian network we require

1. A function to evaluate a given network
2. A method to search through the space of possible networks

Evaluate a given network

Probability assigned to given instance is multiplied over all instances.

To avoid very small numbers, the log likelihood is computed:

Log likelihood sum of the logarithms of the probabilities

Learning Bayesian Networks

In order to learn/train a Bayesian network we require

1. A function to evaluate a given network
2. A method to search through the space of possible networks

Search through the space of possible networks

Vertices are predefined by features and classes

Network structure is learned by a search over the space spanned by all possible edges

Learning Bayesian Networks

In order to learn/train a Bayesian network we require

1. A function to evaluate a given network
2. A method to search through the space of possible networks

Search through the space of possible networks

Vertices are predefined by features and classes

Network structure is learned by a search over the space spanned by all possible edges

Caveat: Log likelihood rewards adding of further edges (Network will overfit).

Learning Bayesian Networks

In order to learn/train a Bayesian network we require

1. A function to evaluate a given network
2. A method to search through the space of possible networks

Search through the space of possible networks

Vertices are predefined by features and classes

Network structure is learned by a search over the space spanned by all possible edges

Caveat: Log likelihood rewards adding of further edges (Network will overfit).

Solution 1 Adding a penalty for the complexity of the network

Solution 2 Use cross-validation to estimate the goodnesss of a fit

Popular methods to evaluate the quality of a network

Akaike Information Criterion (AIC)

$$\text{AIC score} = -(\text{Log likelihood}) + K$$

K Number of independent estimates in all probability tables

N Number of instances in the data

Popular methods to evaluate the quality of a network

Akaike Information Criterion (AIC)

$$\text{AIC score} = -(\text{Log likelihood}) + K$$

MDL metric

$$\text{MDL score} = -(\text{Log likelihood}) + \frac{K}{2} \log N$$

- K Number of independent estimates in all probability tables
- N Number of instances in the data

Algorithms to learn Bayesian networks

A simple and fast algorithm to learn Bayesian networks is called the K2 algorithm

K2 algorithm

Init: Given ordering of the featuers (vertices)

Iteratively: Process each node in turn by greedily adding edges from previously processed nodes

In each step: Add the edge that maximizes the network's score

Until: no further improvement $\rightarrow$ turn to the next node

Algorithms to learn Bayesian networks

A simple and fast algorithm to learn Bayesian networks is called the K2 algorithm

K2 algorithm

Init: Given ordering of the featuers (vertices)

Iteratively: Process each node in turn by greedily adding edges from previously processed nodes

In each step: Add the edge that maximizes the network's score

Until: no further improvement $\rightarrow$ turn to the next node

Overfitting: Can be avoided by restricting the maximum number of parents for each node

Algorithms to learn Bayesian networks

A simple and fast algorithm to learn Bayesian networks is called the K2 algorithm

K2 algorithm

Init: Given ordering of the featuers (vertices)

Iteratively: Process each node in turn by greedily adding edges from previously processed nodes

In each step: Add the edge that maximizes the network's score

Until: no further improvement $\rightarrow$ turn to the next node

Overfitting: Can be avoided by restricting the maximum number of parents for each node

Multistarts: Solution reached dependent on initial ordering

Data structures for fast learning

Learning Bayesian networks involves a lot of counting

Data structures for fast learning

Learning Bayesian networks involves a lot of counting

In order to avoid redundant computations,
all-dimensions (AD) trees might be employed

Data structures for fast learning

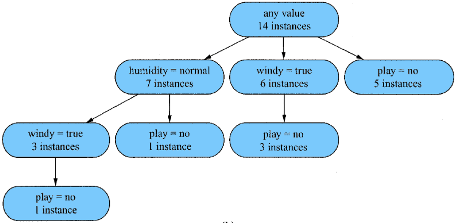
Learning Bayesian networks involves a lot of counting

In order to avoid redundant computations,
all-dimensions (AD) trees might be employed

Creation of such tree for each node in the Bayes network

Humidity	Windy	Play	Count
high	true	yes	1
high	true	no	2
high	false	yes	2
high	false	no	2
normal	true	yes	2
normal	true	no	1
normal	false	yes	4
normal	false	no	0

(a)



(b)

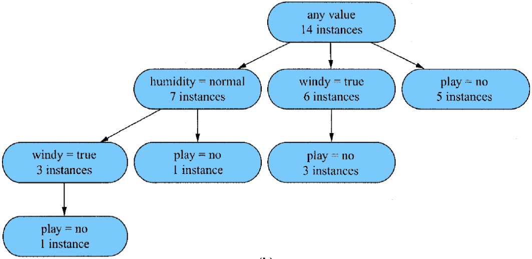
Data structures for fast learning

All possible combinations can be directly read from the tree

→ Node count is low since some information is implicit

Humidity	Windy	Play	Count
high	true	yes	1
high	true	no	2
high	false	yes	2
high	false	no	2
normal	true	yes	2
normal	true	no	1
normal	false	yes	4
normal	false	no	0

(a)



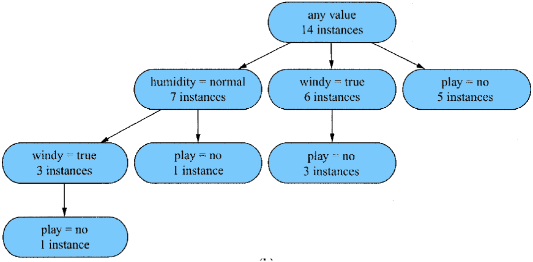
(b)

Data structures for fast learning

Example

Humidity	normal
Windy	true
Play	yes

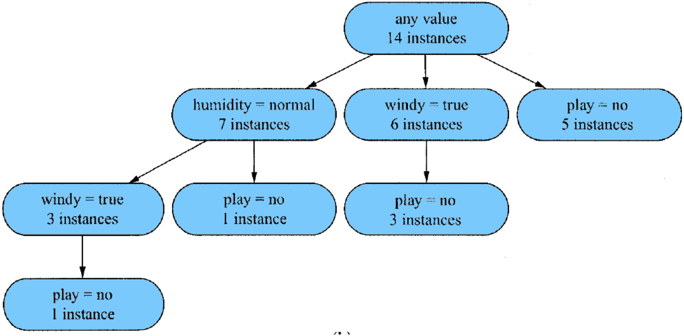
(No node in the tree but one occurrence of [normal-true-no]



Data structures for fast learning

AD trees pay off only if the data contains many instances (e.g. thousands)

Therefore, usually a cutoff parameter k is employed that specifies whether or not an AD tree is created for a specific node



Outline

Questions?

Stephan Sigg
stephan.sigg@cs.uni-goettingen.de

Literature

- C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.
- R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.