Machine Learning and Pervasive Computing

Stephan Sigg

Georg-August-University Goettingen, Computer Networks

22.06.2015

Overview and Structure

13.04.2015	Organisation
13.04.2015	Introduction
20.04.2015	Rule-based learning
27.04.2015	Decision Trees
04.05.2015	A simple Supervised learning algorithm
11.05.2015	–
18.05.2015	Excursion: Avoiding local optima with random search
25.05.2015	–
01.06.2015	High dimensional data
08.06.2015	Artificial Neural Networks
15.06.2015	k-Nearest Neighbour methods
22.06.2015	Probabilistic graphical models
29.06.2015	Topic models
06.07.2015	Unsupervised learning
13.07.2015	Anomaly detection, Online learning, Recom. systems

mvote.ugoe.de/2823

Outline

mvote.ugoe.de/2823

Probabilistic graphical models

Introduction

In the previous models, probabilistic inference was a prominent aspect.

We will now discuss probabilistic graphical models

Some of the classification approaches discussed earlier can be described by such models

mvote.ugoe.de/2823

Probabilistic graphical models

Introduction

In the previous models, probabilistic inference was a prominent aspect.

We will now discuss probabilistic graphical models

Some of the classification approaches discussed earlier can be described by such models

Benefits of probabilistic graphical models

→ Simple way to visualise the structure of a probabilistic model

→ Insights into properties of the model, including conditional independence

→ Graphical representation of complex computations might help to perform inference and learning

mvote.ugoe.de/2823

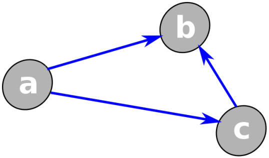
Probabilistic graphical models

Definition

A probabilistic graphical model comprises vertices connected by edges

Vertices represent random variables or groups of variables

Edges represent probabilistic relationships between variables



mvote.ugoe.de/2823

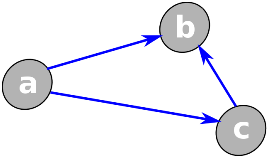
Probabilistic graphical models

Definition

A probabilistic graphical model comprises vertices connected by edges

Vertices represent random variables or groups of variables

Edges represent probabilistic relationships between variables

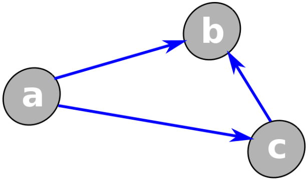


Probabilistic graphical model

The graph captures the way in which the joint distribution over all of the random variables can be decomposed into a product of factors each depending only on a subset of variables

mvote.ugoe.de/2823

Probabilistic graphical models



Example

Consider an arbitrary joint distribution $\mathcal{P}[a, b, c]$.

We can then write

$$\begin{aligned}\mathcal{P}[a, b, c] &= \mathcal{P}[b|a, c]\mathcal{P}[a, c] \\ &= \mathcal{P}[b|a, c]\mathcal{P}[c|a]\mathcal{P}[a]\end{aligned}$$

mvote.ugoe.de/2823

Probabilistic graphical models

Example

Similarly we can define a joint distribution

$$\mathcal{P}[x_1, \dots, x_n] = \mathcal{P}[x_n | x_1, \dots, x_{n-1}] \dots \mathcal{P}[x_2 | x_1] \mathcal{P}[x_1]$$

mvote.ugoe.de/2823

Probabilistic graphical models

Example

Similarly we can define a joint distribution

$$\mathcal{P}[x_1, \ldots, x_n] = \mathcal{P}[x_n | x_1, \ldots, x_{n-1}] \ldots \mathcal{P}[x_2 | x_1] \mathcal{P}[x_1]$$

These graphs are fully connected.
(One edge between every pair of nodes)

mvote.ugoe.de/2823

Probabilistic graphical models

Example

Similarly we can define a joint distribution

$$\mathcal{P}[x_1, \ldots, x_n] = \mathcal{P}[x_n | x_1, \ldots, x_{n-1}] \ldots \mathcal{P}[x_2 | x_1] \mathcal{P}[x_1]$$

These graphs are fully connected.
(One edge between every pair of nodes)

The actual absence of links in the graph covers intersting information about the properties of the class of distributions represented

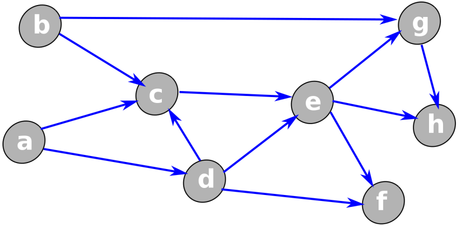
mvote.ugoe.de/2823

Probabilistic graphical models

Definition

A general distribution for a graph with n nodes is

$$\mathcal{P}[x] = \prod_{i=1}^{n} \mathcal{P}[x_i | \text{parents of vertex } x_i]$$



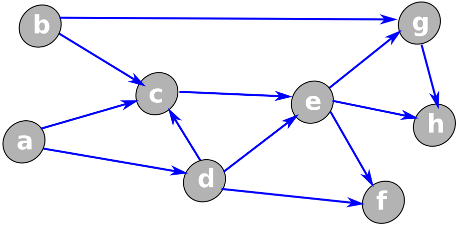
mvote.ugoe.de/2823

Probabilistic graphical models

Definition

A general distribution for a graph with n nodes is

$$\mathcal{P}[x] = \prod_{i=1}^{n} \mathcal{P}[x_i | \text{parents of vertex } x_i]$$



Remark: Bayesian networks are represented in this way

mvote.ugoe.de/2823

Outline

mvote.ugoe.de/2824

Bayesian decision 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

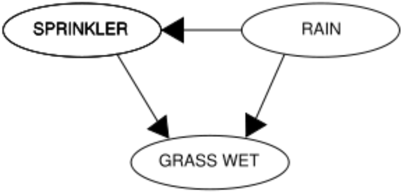
In fact, it is possible to represent uncertainty by probability

mvote.ugoe.de/2824

Example

SPRINKLER

RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN

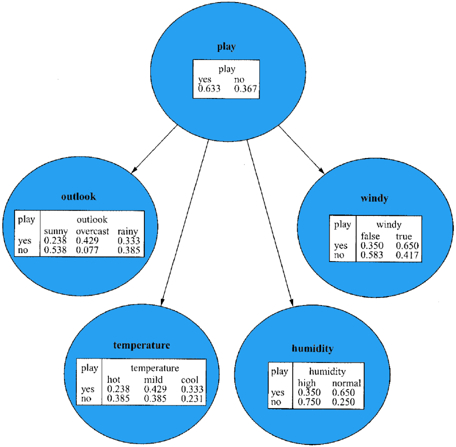
T	F
0.2	0.8

GRASS WET

SPRINKLER	RAIN	T	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

mvote.ugoe.de/2824

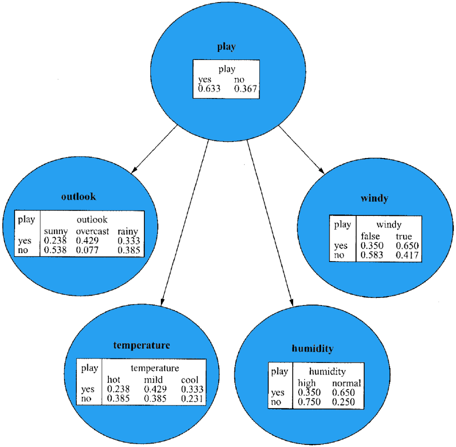
Bayesian Networks



mvote.ugoe.de/2824

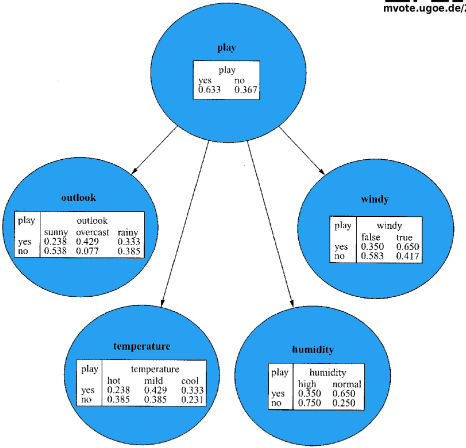
Bayesian Networks

Directed acyclic Graph with one vertex for each feature or class



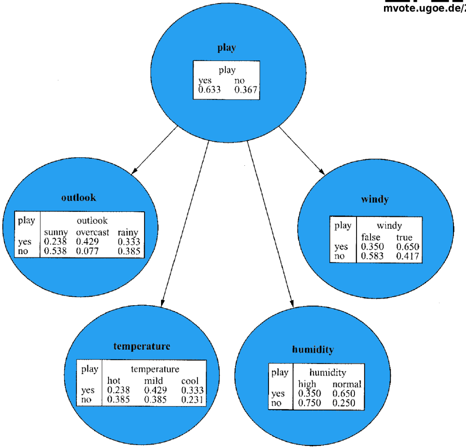
mvote.ugoe.de/2824

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



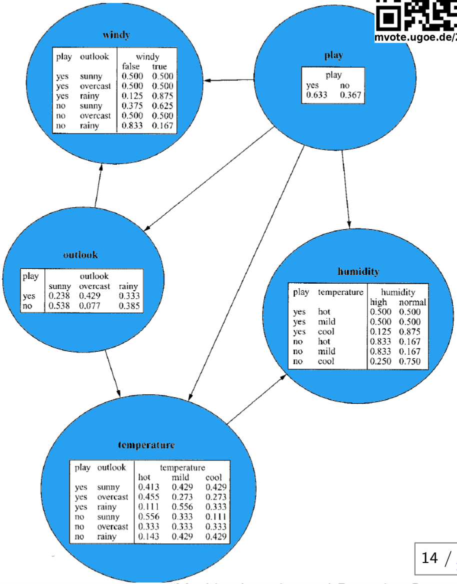
Left side of the distribution table in each node contains a column for every ingoing edge from a parent node

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



mvote.ugoe.de/2824

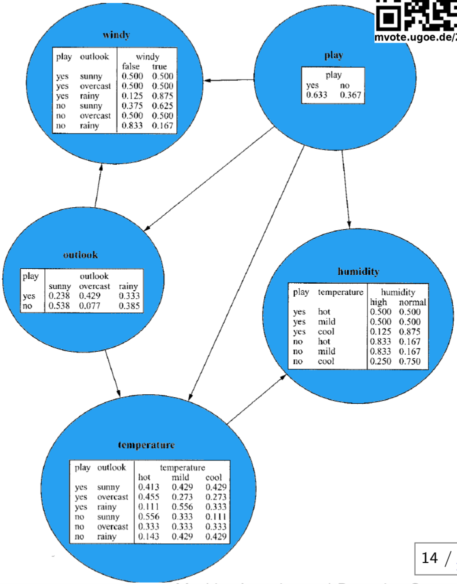
Prediction of class probabilities



mvote.ugoe.de/2824

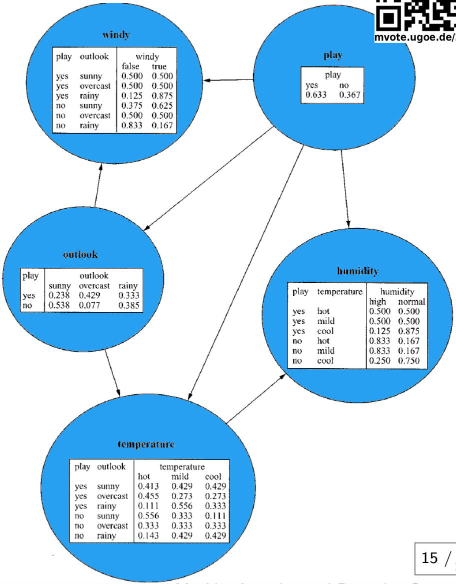
Prediction of class probabilities

For a particular sample, multiply all corresponding probabilities



mvote.ugoe.de/2824

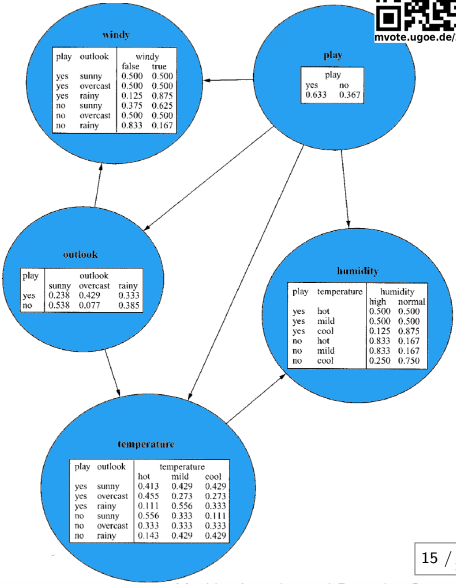
Example



mvote.ugoe.de/2824

Example

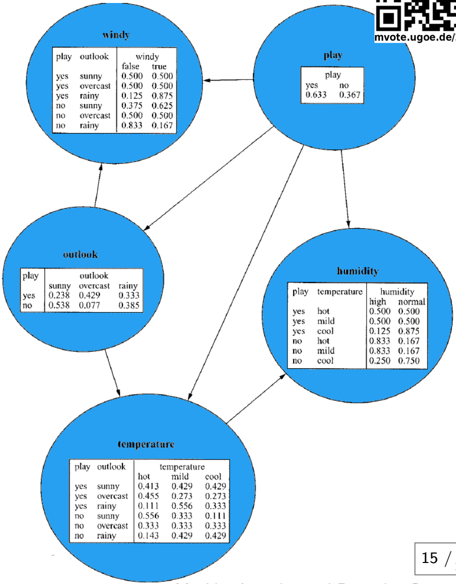
outlook	rainy
temperature	cool
humidity	high
windy	true



mvote.ugoe.de/2824

Example

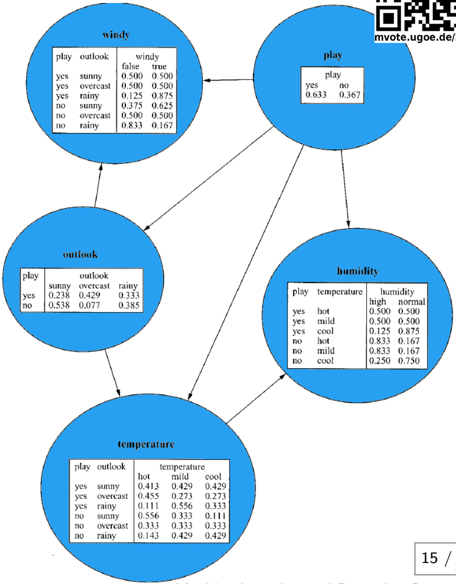
outlook	rainy
temperature	cool
humidity	high
windy	true
play = no	$0.367 \cdot 0.167 \cdot 0.385 \cdot 0.25 \cdot 0.429 = 0.0025$



mvote.ugoe.de/2824

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	$= 0.0077$



mvote.ugoe.de/2824

Example

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

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

mvote.ugoe.de/2824

Example

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

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

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

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

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.

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

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).

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

Data structures for fast learning

Learning Bayesian networks involves a lot of counting

mvote.ugoe.de/2824

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

mvote.ugoe.de/2824

Data structures for fast learning

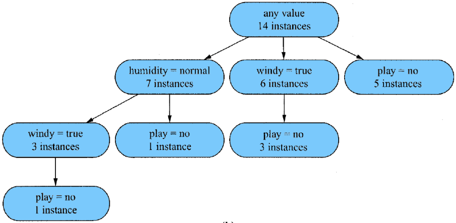
Learning Bayesian networks involves a lot of counting

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)

mvote.ugoe.de/2824

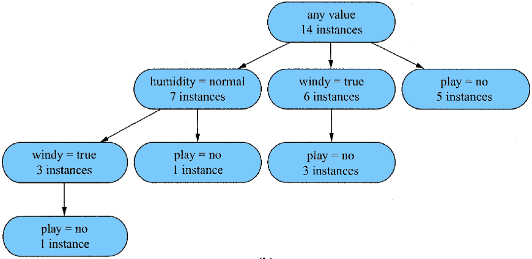
Data structures for fast learning

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)



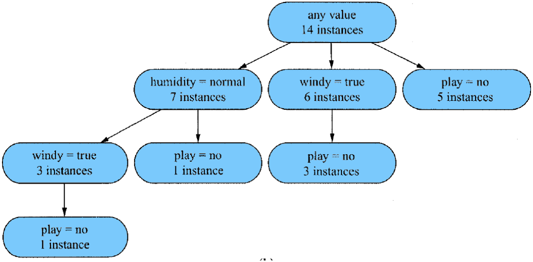
(b)

Data structures for fast learning

Example

Humidity	normal
Windy	true
Play	yes

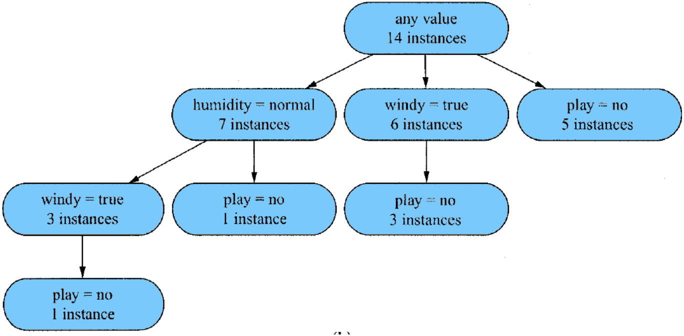
(No node in the tree but one occurrence of [normal-true-no]



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



mvote.ugoe.de/2825

Outline

Introduction

Bayesian Networks

Naïve Bayes

Bayesian Curve fitting

Hidden Markov models
- Evaluation
- Deconding
- Learning

Conditional random fields
- Conditional independence

mvote.ugoe.de/2825

Naïve Bayes

Naïve Bayes

Bayes Networks require indenpendency of events.

Often, this can not be guaranteed for real-world problems and events

→ Naïve Bayes is naïve in the sense that independence is assumed against one's better judgement

mvote.ugoe.de/2825

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					

mvote.ugoe.de/2825

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					

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

Likelihood of YES:

Likelihood of NO:

Naïve Bayes classificaiton

mvote.ugoe.de/2825

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: $\frac{5}{14} \cdot \frac{2}{14} \cdot \frac{3}{14} \cdot \frac{7}{14} \cdot \frac{14}{30} = 0.0026$

mvote.ugoe.de/2825

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

mvote.ugoe.de/2825

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

mvote.ugoe.de/2825

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:

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

mvote.ugoe.de/2825

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:

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

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

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

mvote.ugoe.de/2825

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.

mvote.ugoe.de/2825

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

mvote.ugoe.de/2825

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

mvote.ugoe.de/2825

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.

mvote.ugoe.de/2825

Naïve Bayes classificaiton

Example (Laplace estimator)

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

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

In practice, these small modifications make little difference given

mvote.ugoe.de/2825

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

mvote.ugoe.de/2825

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

mvote.ugoe.de/2825

Outline

mvote.ugoe.de/2825

Probabilistic graphical models

Bayesian Curve fitting

W Polynomial coefficients

$X = (x_1, \ldots, x_n)^T$ Input data

$Y = (y_1, \ldots, y_n)^T$ Observed data (Ground truth)

σ^2 Noise variance

α representation of the precision of the Gaussian prior over W

$$\mathcal{P}[Y, W] = \mathcal{P}[W] \prod_{i=1}^{n} \mathcal{P}[y_i | W]$$

(omitting deterministic parameters)

mvote.ugoe.de/2825

Probabilistic graphical models

Bayesian Curve fitting

W Polynomial coefficients

$X = (x_1, \ldots, x_n)^T$ Input data

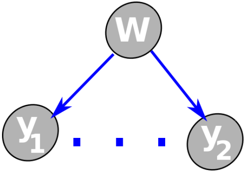
$Y = (y_1, \ldots, y_n)^T$ Observed data (Ground truth)

σ^2 Noise variance

α representation of the precision of the Gaussian prior over W

$$\mathcal{P}[Y, W] = \mathcal{P}[W] \prod_{i=1}^{n} \mathcal{P}[y_i | W]$$

(omitting deterministic parameters)



mvote.ugoe.de/2825

Probabilistic graphical models

Bayesian Curve fitting

$$\mathcal{P}[Y, W | X, \alpha, \sigma^2] = \mathcal{P}[W|\alpha] \prod_{i=1}^{n} \mathcal{P}[y_i | W, x_i, \sigma^2]$$

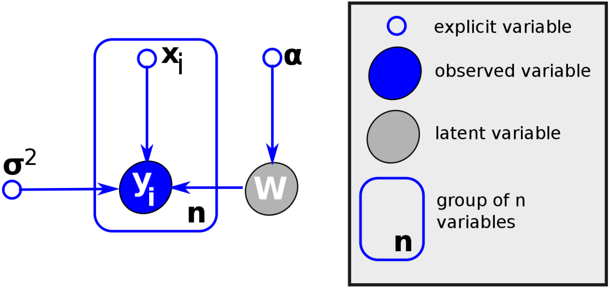


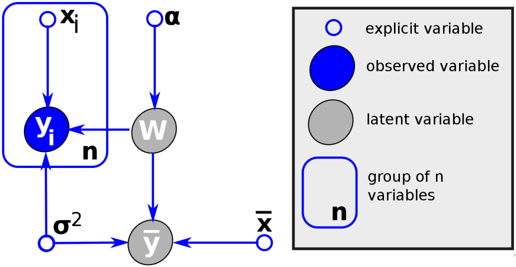
Plate notation

mvote.ugoe.de/2825

Probabilistic graphical models

Prediction of $\overline{y}$ given the model and a new sample $\overline{x}$ as

$$\mathcal{P}[\overline{y}, Y, W|\overline{x}, X, \alpha, \sigma^2] = \left[\prod_{i=1}^{n} \mathcal{P}[y_i|W, x_i, \sigma^2]\right] \mathcal{P}[W|\alpha]\mathcal{P}[\overline{y}|\overline{x}, W, \sigma^2]$$



mvote.ugoe.de/2825

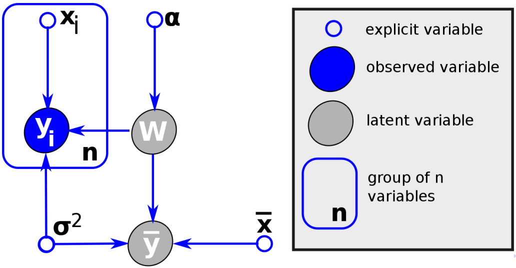
Probabilistic graphical models

Prediction of $\overline{y}$ given the model and a new sample $\overline{x}$ as

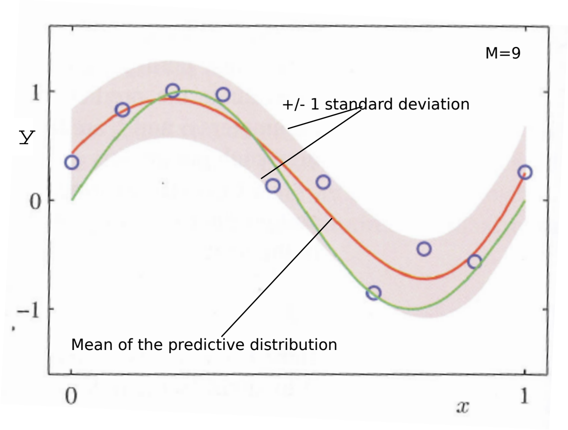
$$\mathcal{P}[\overline{y}, Y, W|\overline{x}, X, \alpha, \sigma^2] = \left[\prod_{i=1}^{n} \mathcal{P}[y_i|W, x_i, \sigma^2]\right] \mathcal{P}[W|\alpha]\mathcal{P}[\overline{y}|\overline{x}, W, \sigma^2]$$

Sum rule of probability leads to predictive distribution for $\overline{y}$:

$$\mathcal{P}[\overline{y}|\overline{x}, X, \alpha, Y, \sigma^2] \propto \int \mathcal{P}[\overline{y}, Y, W|\overline{x}, X, \alpha, \sigma^2]dW$$



Bayesian curve fitting



mvote.ugoe.de/2826

Outline

Markov chains

mvote.ugoe.de/2826

Markov processes

- Intensively studied
- Major branch in the theory of stochastic processes

A. A. Markov (1856 – 1922)

Extended by A. Kolmogorov to chains of infinitely many states

- 'Anfangsgründe der Theorie der Markoffschen Ketten mit unendlich vielen möglichen Zuständen' (1936) [1]

[1] A. Kolmogorov, *Anfangsgründe der Theorie der Markoffschen Ketten mit unendlich vielen möglichen Zuständen*, 1936.

mvote.ugoe.de/2826

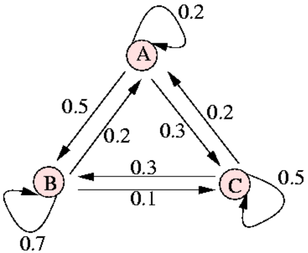
Markov chains

- Theory applied to a variety of algorithmic problems
- Standard tool in many probabilistic applications

Intuitive graphical representation

- Suitable for graphical illustration of stochastic processes

Popular for their simplicity and easy applicability to huge set of problems[2]



[2] William Feller, *An introduction to probability theory and its applications*, Wiley, 1968.

Markov chains

mvote.ugoe.de/2826

Independent trials of events

Dependent trials of events

Markov chains

mvote.ugoe.de/2826

Independent trials of events

- Set of possible outcomes of a measurement E_i associated with occurrence probability p_i
- Probability to observe sample sequence:
  - $P\{(E_1, E_2, \ldots, E_i)\} = p_1 p_2 \cdots p_i$

Dependent trials of events

Markov chains

mvote.ugoe.de/2826

Independent trials of events

- Set of possible outcomes of a measurement E_i associated with occurrence probability p_i
- Probability to observe sample sequence:
  - $P\{(E_1, E_2, \ldots, E_i)\} = p_1 p_2 \cdots p_i$

Dependent trials of events

- Probability to observe specific sequence $E_1, E_2, \ldots, E_i$ obtained by conditional probability:

$$P(E_i | E_1, E_2, \ldots, E_{i-1})$$

mvote.ugoe.de/2826

Markov chains

Independent random variables

Dependent random variables

mvote.ugoe.de/2826

Markov chains

Independent random variables

- Number of coin tosses until 'head' is observed
- Radioactive atoms always have same probability of decaying at next trial

Dependent random variables

mvote.ugoe.de/2826

Markov chains

Independent random variables

- Number of coin tosses until 'head' is observed
- Radioactive atoms always have same probability of decaying at next trial

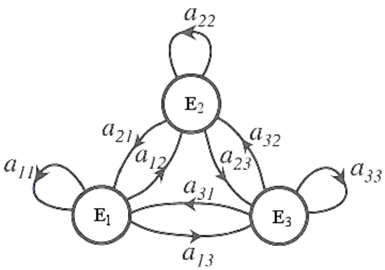
Dependent random variables

- Knowledge that no car has passed for five minutes increases expectation that it will come soon.
- Coin tossing:
  - Probability that the cumulative numbers of heads and tails will equalize at the second trial is $\frac{1}{2}$
  - Given that they did not, the probability that they equalize after two additional trials is only $\frac{1}{4}$

Markov property

In the theory of stochastic processes the described lack of meme
is connected with the Markov property.

mvote.ugoe.de/2826



Outcome depends exclusively on outcome of directly preceding trial

- Every sequence (E_i, E_j) has a conditional probability p_{ij}
- Additionally: Probability a_i of the event E_i

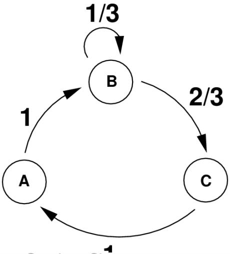
Markov chains

Markov chain

A sequence of observations $E_1, E_2, \ldots$ is called a Markov chain if the probabilities of sample sequences are defined by

$$P(E_1, E_2, \ldots, E_i) = a_1 \cdot p_{12} \cdot p_{23} \cdot \cdots \cdot p_{(i-1)i}.$$

and fixed conditional probabilities p_{ij} that the event E_i is observed directly in advance of E_j.



mvote.ugoe.de/2826

Markov chains

Described by probability a for initial distribution and matrix P of transition probabilities.

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots \\ p_{21} & p_{22} & p_{23} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

P is called a stochastic matrix

(Square matrix with non-negative entries that sum to 1 in each row)

Markov chains

p_{ij}^k denotes probability that E_j is observed exactly k observations after E_i was observed.

Calculated as the sum of the probabilities for all possible paths $E_i E_{i_1} \cdots E_{i_{k-1}} E_j$ of length k

We already know

$$p_{ij}^1 = p_{ij}$$

Consequently:

$$\begin{aligned} p_{ij}^2 &= \sum_\nu p_{i\nu} \cdot p_{\nu j} \\ p_{ij}^3 &= \sum_\nu p_{i\nu} \cdot p_{\nu j}^2 \end{aligned}$$

mvote.ugoe.de/2826

Markov chains

By mathematical induction:

$$p_{ij}^{n+1} = \sum_{\nu} p_{i\nu} \cdot p_{\nu j}^{n}$$

and

$$p_{ij}^{n+m} = \sum_{\nu} p_{i\nu}^{m} \cdot p_{\nu j}^{n} = \sum_{\nu} p_{i\nu}^{n} \cdot p_{\nu j}^{m}$$

Similar to matrix P we can create a matrix P^n that contains all p_{ij}^n

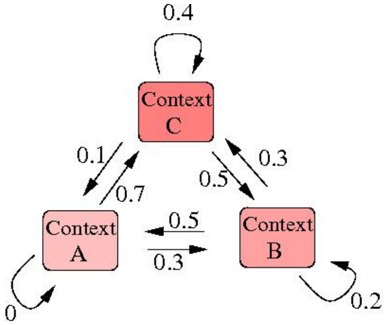
p_{ij}^{n+1} obtained from P^{n+1}: Multiply row i of P with column j of P^n

Symbolically: $P^{n+m} = P^n P^m$.

$$P^n = \begin{bmatrix} p_{11}^n & p_{12}^n & p_{13}^n & \cdots \\ p_{21}^n & p_{22}^n & p_{23}^n & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

mvote.ugoe.de/2826

Markov chains



	Context A	Context B	Context C
Context A	0	0.3	0.7
Context B	0.5	0.2	0.3
Context C	0.1	0.5	0.4

	Context A	Context B	Context C
Context A	0.22	0.41	0.37
Context B	0.13	0.34	0.53
Context C	0.29	0.33	0.38

	Context A	Context B	Context C
Context A	0.242	0.333	0.425
Context B	0.223	0.372	0.405
Context C	0.203	0.343	0.454

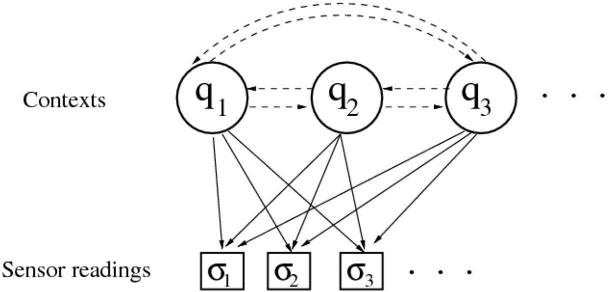
mvote.ugoe.de/2826

Hidden Markov Models

Make a sequence of decisions for a process that is not directly observable[3]

Current states of the process might be impacted by prior states

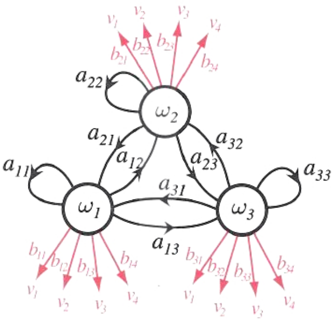
HMM often utilised in speech recognition or gesture recognition



[3] Richard O. Duda, Peter E. Hart and David G. Stork, *Pattern classification*, Wiley-interscience, 2001.

mvote.ugoe.de/2826

Hidden Markov Models



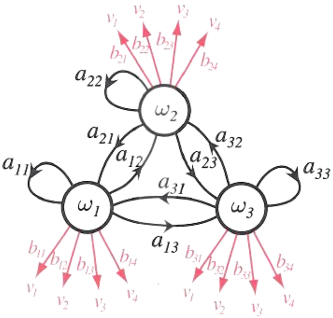
At every time step t the system is in an internal state $\omega(t)$

Additionally, we assume that it emits a (visible) symbol $v(t)$

Only access to visible symbols and not to internal states

mvote.ugoe.de/2826

Hidden Markov Models



Probability to be in state $\omega_j(t)$ and emit symbol $v_k(t)$:

$$P(v_k(t)|\omega_j(t)) = b_{jk}$$

Transition probabilities: $p_{ij} = P(\omega_j(t+1)|\omega_i(t))$

Emission probability: $b_{jk} = P(v_k(t)|\omega_j(t))$

mvote.ugoe.de/2826

Hidden Markov Models

Central issues in hidden Markov models:

Evaluation problem Determine the probability that a particular sequence of visible symbols V^n was generated by a given hidden Markov model

Decoding problem Determine the most likely sequence of hidden states ω^n that led to a specific sequence of observations V^n

Learning problem Given a set of training observations of visible symbols, determine the parameters p_{ij} and b_{jk} for a given HMM

Hidden Markov Models – Evaluation problem

Probability that model produces a sequence V^n:

$$P(V^n) = \sum_{\overline{\omega}^n} P(V^n|\overline{\omega}^n)P(\overline{\omega}^n)$$

Also:

$$P(\overline{\omega}^n) = \prod_{t=1}^{n} P(\omega(t)|\omega(t-1))$$

$$P(V^n|\overline{\omega}^n) = \prod_{t=1}^{n} P(v(t)|\omega(t))$$

Together:

$$P(V^n) = \sum_{\overline{\omega}^n} \prod_{t=1}^{n} P(v(t)|\omega(t))P(\omega(t)|\omega(t-1))$$

mvote.ugoe.de/2826

Hidden Markov Models – Evaluation problem

Probability that model produces a sequence V^n:

$$P(V^n) = \sum_{\overline{\omega}^n} \prod_{t=1}^{n} P(v(t)|\omega(t))P(\omega(t)|\omega(t-1))$$

Formally complex but straightforward

Naive computational complexity

- $\mathcal{O}(c^n n)$

Hidden Markov Models – Evaluation problem

Probability that model produces a sequence V^n:

$$P(V^n) = \sum_{\overline{\omega}^n} \prod_{t=1}^{n} P(v(t)|\omega(t))P(\omega(t)|\omega(t-1))$$

Computationally less complex algorithm:

- Calculate $P(V^n)$ recursively
- $P(v(t)|\omega(t))P(\omega(t)|\omega(t-1))$ involves only $v(t), \omega(t)$ and $\omega(t-1)$

$$\alpha_j(t) = \begin{cases} 0 & t = 0 \text{ and } j \neq \text{ initial state} \\ 1 & t = 0 \text{ and } j = \text{ initial state} \\ \left[\sum_i \alpha_i(t-1)p_{ij}\right] b_{jk} & \text{otherwise } {\scriptstyle (b_{jk} \text{ leads to observed } v(t))} \end{cases}$$

Hidden Markov Models – Evaluation problem

Forward Algorithm

Computational complexity: $O(c^2 n)$

Forward algorithm

initialise $t \leftarrow 0, p_{ij}, b_{jk}, V^n, \alpha_j(0)$
    for $t \leftarrow t+1$
        $j \leftarrow 0$
        for $j \leftarrow j+1$
            $\alpha_j(t) \leftarrow b_{jk} \sum_{i=1}^{c} \alpha_i(t-1) p_{ij}$
        until $j = c$
    until $t = n$
return $P(V^n) \leftarrow \alpha_j(n)$ for the final state
end

mvote.ugoe.de/2826

Hidden Markov Models – Decoding problem

Given a sequence V^n, find most probable sequence of hidden states

Enumeration of every possible path will cost $O(c^n)$

- Not feasible

mvote.ugoe.de/2826

Hidden Markov Models – Decoding problem

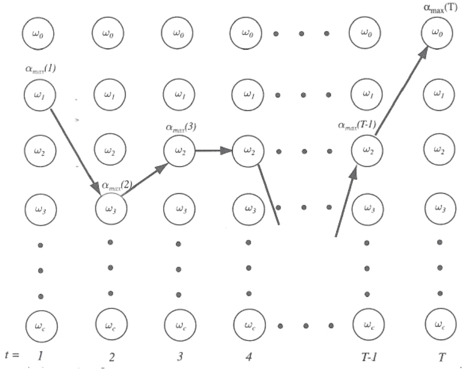
Given a sequence V^n, find most probable sequence of hidden states

Decoding algorithm

initialise: path $\leftarrow \{\}, t \leftarrow 0$
    for $t \leftarrow t+1$
        $j \leftarrow 0$;
        for $j \leftarrow j+1$
            $\alpha_j(t) \leftarrow b_{jk} \sum_{i=1}^{c} \alpha_i(t-1) p_{ij}$
        until $j = c$
        $j' \leftarrow arg\max_j \alpha_j(t)$
        append $\omega_{j'}$ to path
    until $t = n$
return path
end

mvote.ugoe.de/2826

Hidden Markov Models – Decoding problem



Computational time of the decoding algorithm

- $O(c^2 n)$

mvote.ugoe.de/2826

Hidden Markov Models – Learning problem

Determine the model parameters p_{ij} and b_{jk}

- Given: Training sample of observed values V^n

No method known to obtain the optimal or most likely set of parameters from the data

- However, we can nearly always determine a good solution by the forward-backward algorithm
- General expectation maximisation algorithm
- Iteratively update weights in order to better explain the observed training sequences

Hidden Markov Models – Learning problem

mvote.ugoe.de/2826

Probability that the model is in state $\omega_i(t)$ and will generate the remainder of the given target sequence:

$$\beta_i(t) = \begin{cases} 0 & t = n \text{ and } \omega_i(t) \text{ not final hidden state} \\ 1 & t = n \text{ and } \omega_i(t) \text{ final hidden state} \\ \sum_j \beta_j(t+1) p_{ij} b_{jk} & \text{otherwise } (b_{jk} \text{ leads to } v(t+1)) \end{cases}$$

Hidden Markov Models – Learning problem

$\alpha_i(t)$ and $\beta_i(t)$ only estimates of their true values since transition probabilities p_{ij}, b_{jk} unknown

Probability of transition between $\omega_i(t-1)$ and $\omega_j(t)$ can be estimated

- Provided that the model generated the entire training sequence V^n by **any** path

$$\gamma_{ij}(t) = \frac{\alpha(t-1)p_{ij}b_{jk}\beta_j(t)}{P(V^n|\Omega)}$$

Probability that model generated sequence V^n:

$$P(V^n|\Omega)$$

Hidden Markov Models – Learning problem

Calculate improved estimate for p_{ij} and b_{jk}

$$\overline{p_{ij}} = \frac{\sum_{t=1}^{n} \gamma_{ij}(t)}{\sum_{t=1}^{n} \sum_{k} \gamma_{ik}(t)}$$

$$\overline{b_{jk}} = \frac{\sum_{t=1, v(t)=v_k}^{n} \sum_{l} \gamma_{jl}(t)}{\sum_{t=1}^{n} \sum_{l} \gamma_{jl}(t)}$$

Start with rough estimates of p_{ij} and b_{jk}

Calculate improved estimates

Repeat until some convergence is reached

mvote.ugoe.de/2826

Hidden Markov Models – Learning problem

Forward-Backward algorithm

```
initialise p_ij, b_jk, V^n, convergence criterion Δ, t ← 0
    do t ← t + 1
        compute overline{p_ij(t)}
        compute overline{b_jk(t)}
        p_ij(t) ← overline{p_ij(t)}
        b_jk(t) ← overline{b_jk(t)}
    until max_{i,j,k}[p_ij(z) − p_ij(z − 1), b_jk(t) − b_jk(t − 1)] < Δ
                (convergence achieved)
return p_ij ← p_ij(t), b_jk ← b_jk(t)
end
```

$$\text{initialise } p_{ij}, b_{jk}, V^n, \text{convergence criterion } \Delta, t \leftarrow 0$$
$$\text{do } t \leftarrow t+1;\ \text{compute } \overline{p_{ij}(t)};\ \text{compute } \overline{b_{jk}(t)};\ p_{ij}(t) \leftarrow \overline{p_{ij}(t)};\ b_{jk}(t) \leftarrow \overline{b_{jk}(t)}$$
$$\text{until } \max_{i,j,k}[p_{ij}(z) - p_{ij}(z-1), b_{jk}(t) - b_{jk}(t-1)] < \Delta \text{ (convergence achieved)}$$
$$\text{return } p_{ij} \leftarrow p_{ij}(t),\ b_{jk} \leftarrow b_{jk}(t)$$

mvote.ugoe.de/2826

Outline

mvote.ugoe.de/2826

Probabilistic graphical models

Conditional independence between nodes of the graph

Consider variables a, b and c and assume the conditional distribution

$$\mathcal{P}[a|b,c] = \mathcal{P}[a|c]$$

Then: a is conditionally independent of b given c

mvote.ugoe.de/2826

Probabilistic graphical models

Conditional independence between nodes of the graph

Consider variables a, b and c and assume the conditional distribution

$$\mathcal{P}[a|b,c] = \mathcal{P}[a|c]$$

Then: a is conditionally independent of b given c

Notation: $a \perp\!\!\!\perp b \mid c$

mvote.ugoe.de/2826

Probabilistic graphical models

Conditional independence between nodes of the graph

Consider variables a, b and c and assume the conditional distribution

$$\mathcal{P}[a|b,c] = \mathcal{P}[a|c]$$

Then: a is conditionally independent of b given c

Notation: $a \perp\!\!\!\perp b \mid c$

Importance of conditional independence in probabilistic models

Conditional independence in probabilistic models for pattern recognition

- simplifies the structure of a model and
- the computations needed to perform inference and learning

Probabilistic graphical models

Conditional independence between nodes of the graph

Conditional independence can be read directly from the graph !

mvote.ugoe.de/2826

Probabilistic graphical models

Conditional independence between nodes of the graph

Conditional independence can be read directly from the graph !

Example

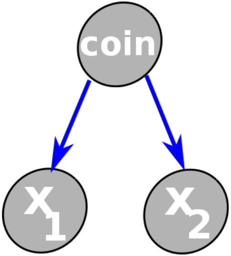
Assume a random experiment containing a biased and a fair coin.

Biased: $\mathcal{P}[\text{head}] = 0.8,\ \mathcal{P}[\text{tail}] = 0.2$

Fair: $\mathcal{P}[\text{head}] = \mathcal{P}[\text{tail}] = 0.5$

The experiment consists of two steps:

1. Choose which coin to toss
2. Toss the coin twice



mvote.ugoe.de/2826

Probabilistic graphical models

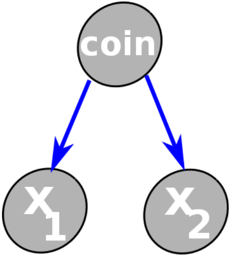
Conditional independence between nodes of the graph

Conditional independence can be read directly from the graph !

Example

If we are ignorant of which coin we chose, the result of the first toss impacts our expectation of what we see in the second toss:

→ e.g. if the first toss came out head, this will increase our expectation to see head also in the second toss



mvote.ugoe.de/2826

Probabilistic graphical models

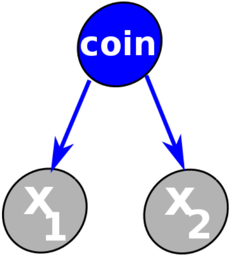
Conditional independence between nodes of the graph

Conditional independence can be read directly from the graph !

Example

However, if we were given information about which coin we chose, the x_1 and x_2 independent.

→ Since we know the distribution expected by both coins, knowledge of the outcome of x_1 does not change the expected outcome of x_2



mvote.ugoe.de/2826

Probabilistic graphical models

Conditional independence between nodes of the graph

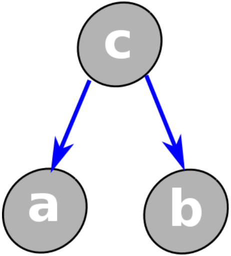
$$\mathcal{P}[a, b, c] = \mathcal{P}[a|c]\mathcal{P}[b|c]\mathcal{P}[c]$$

If none of the variables are observed, we can investigate whether a and b are independent by marginalizing both sides with respect to c:

$$\mathcal{P}[a, b] = \sum_c \mathcal{P}[a|c]\mathcal{P}[b|c]\mathcal{P}[c]$$

Since this does not factorize into $\mathcal{P}[a]\mathcal{P}[b]$ in general, we conclude

$$a \not\perp\!\!\!\perp b \mid \emptyset$$



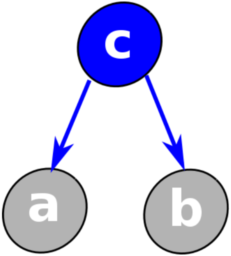
mvote.ugoe.de/2826

Probabilistic graphical models

Conditional independence between nodes of the graph

If, however, c is observed, we obtain

$$\begin{aligned}\mathcal{P}[a,b|c] &= \frac{\mathcal{P}[a,b,c]}{\mathcal{P}[c]} \\ &= \frac{\mathcal{P}[a|c]\mathcal{P}[b|c]\mathcal{P}[c]}{\mathcal{P}[c]} \\ &= \mathcal{P}[a|c]\mathcal{P}[b|c]\end{aligned}$$



And thus obtain the conditional independence property

$$a \perp\!\!\!\perp b \mid c$$

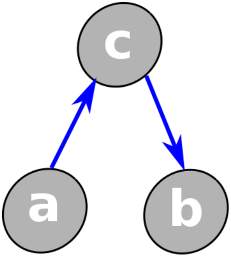
Probabilistic graphical models

Conditional independence between nodes of the graph

$$\mathcal{P}[a,b,c] = \mathcal{P}[a]\mathcal{P}[c|a]\mathcal{P}[b|c]$$

Marginalizing over c leads to

$$\begin{aligned}\mathcal{P}[a,b] &= \mathcal{P}[a]\sum_c \mathcal{P}[c|a]\mathcal{P}[b|c] \\ &= \mathcal{P}[a]\mathcal{P}[b|a]\end{aligned}$$



This does not factorize into $\mathcal{P}[a]\mathcal{P}[b]$ in general and therefore

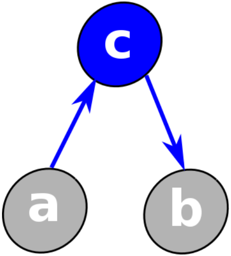
$$a \not\perp\!\!\!\perp b \mid \emptyset$$

mvote.ugoe.de/2826

Probabilistic graphical models

Conditional independence between nodes of the graph

$$\begin{aligned}\mathcal{P}[a, b|c] &= \frac{\mathcal{P}[a, b, c]}{\mathcal{P}[c]} \\ &= \frac{\mathcal{P}[a]\mathcal{P}[c|a]\mathcal{P}[b|c]}{\mathcal{P}[c]} \\ &= \mathcal{P}[a|c]\mathcal{P}[b|c]\end{aligned}$$



And therefore

$$a \perp\!\!\!\perp b \mid c$$

Probabilistic graphical models

Conditional independence between nodes of the graph

mvote.ugoe.de/2826

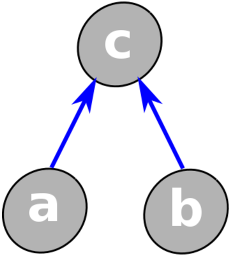
$$\mathcal{P}[a, b, c] = \mathcal{P}[a]\mathcal{P}[b]\mathcal{P}[c|a, b]$$

Marginalizing over c leads to

$$\mathcal{P}[a, b] = \mathcal{P}[a]\mathcal{P}[b]$$

So, in this case, we obtain

$$a \perp\!\!\!\perp b \mid \emptyset$$

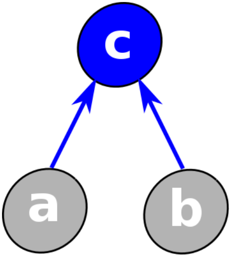


mvote.ugoe.de/2826

Probabilistic graphical models

Conditional independence between nodes of the graph

$$\mathcal{P}[a,b|c] = \frac{\mathcal{P}[a,b,c]}{\mathcal{P}[c]} = \frac{\mathcal{P}[a]\mathcal{P}[b]\mathcal{P}[c|a,b]}{\mathcal{P}[c]}$$



Which does not in general factorize into $\mathcal{P}[a|c]\mathcal{P}[b|c]$ and so

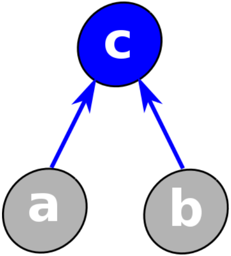
$$a \not\perp\!\!\!\perp b \mid c$$

mvote.ugoe.de/2826

Probabilistic graphical models

Conditional independence between nodes of the graph

$$\mathcal{P}[a,b|c] = \frac{\mathcal{P}[a,b,c]}{\mathcal{P}[c]} = \frac{\mathcal{P}[a]\mathcal{P}[b]\mathcal{P}[c|a,b]}{\mathcal{P}[c]}$$



Which does not in general factorize into $\mathcal{P}[a|c]\mathcal{P}[b|c]$ and so

$$a \not\perp\!\!\!\perp b \mid c$$

This rule applies also if, instead of c, any its descendants are observed !

Probabilistic graphical models

Conditional independence between nodes of the graph

D-separation

Consider a general directed graph in which A, B and C are arbitrary nonintersecting sets of nodes

A is d-separated from B by C when all possible paths from A to B contain a node such that either

a) the node is in the set C and the arrows meet head-to-tail or tail-to-tail

b) the node is not in the set C nor any of its descendants and the arrows meet head-to-head

mvote.ugoe.de/2826

Probabilistic graphical models

The concept of d-separation helps us to understand the probability distributions that are expressed by a particular graphical model:

Probabilistic graphical models

The concept of d-separation helps us to understand the probability distributions that are expressed by a particular graphical model:

We have seen above that the joint distribution of a graph is given as its factorization:

$$\mathcal{P}[x] = \prod_{i=1}^{n} \mathcal{P}[x_i | \text{parents of vertex } x_i]$$

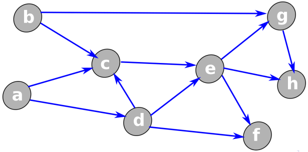
Probabilistic graphical models

The concept of d-separation helps us to understand the probability distributions that are expressed by a particular graphical model:

We have seen above that the joint distribution of a graph is given as its factorization:

$$\mathcal{P}[x] = \prod_{i=1}^{n} \mathcal{P}[x_i | \text{parents of vertex } x_i]$$

The graph literally filters those distributions which can express it in terms of the factorization implied by the graph.



mvote.ugoe.de/2826

Probabilistic graphical models

The concept of d-separation helps us to understand the probability distributions that are expressed by a particular graphical model:

We have seen above that the joint distribution of a graph is given as its factorization:

$$\mathcal{P}[x] = \prod_{i=1}^{n} \mathcal{P}[x_i | \text{parents of vertex } x_i]$$

The graph literally filters those distributions which can express it in terms of the factorization implied by the graph.

It can be shown that the set of distributions that pass the filter is precisely the set of distributions that fulfills the set of conditional independence properties defined by the d-separation property.

Probabilistic graphical models

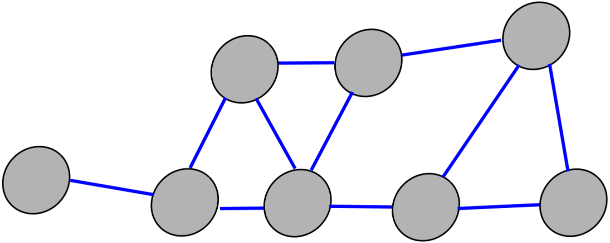
Undirected graphical models

Undirected graphical models

Also graphical models that are described by undirected graphs specify

a) a factorization

b) a set of conditional independence relations

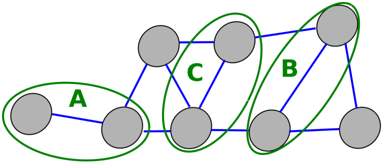


mvote.ugoe.de/2826

Probabilistic graphical models

Undirected graphical models

Assume three test of nodes A, B and C in such an undirected graph

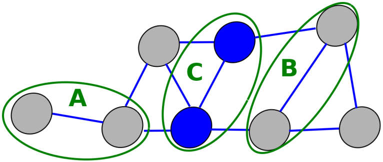


mvote.ugoe.de/2826

Probabilistic graphical models

Undirected graphical models

Assume three test of nodes A, B and C in such an undirected graph



Conditional independence in undirected graphs

$A \perp\!\!\!\perp B \mid C$ if all paths between A and B contain an observed node from the set C

$A \not\perp\!\!\!\perp B \mid C$ if at least one path between A and B does not contain any observed node.

Probabilistic graphical models

mvote.ugoe.de/2826

Factorization rule for undirected graphs

Two nodes a and b in a graph are conditionally independent (given all other nodes) if they are not connected by an edge

$\rightarrow$ Since there is no direct path between the nodes

Probabilistic graphical models

mvote.ugoe.de/2826

Factorization rule for undirected graphs

Two nodes a and b in a graph are conditionally independent (given all other nodes) if they are not connected by an edge

$\rightarrow$ Since there is no direct path between the nodes

Therefore, the joint distribution described by the graph is given by functions of the variables of the maximal cliques in the graph

mvote.ugoe.de/2826

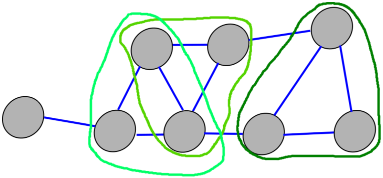
Probabilistic graphical models

Factorization rule for undirected graphs

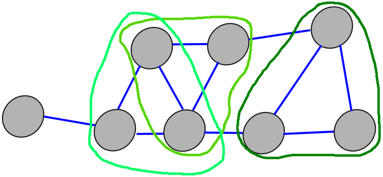
Two nodes a and b in a graph are conditionally independent (given all other nodes) if they are not connected by an edge

→ Since there is no direct path between the nodes

Therefore, the joint distribution described by the graph is given by functions of the variables of the maximal cliques in the graph



Probabilistic graphical models



The joint distribution is written as a product of potential functions $\phi_C(X_C)$ over the maximal cliques X_C of the graph:

$$\mathcal{P}[X] = \frac{1}{Z}\prod_C \phi_C(X_C)$$

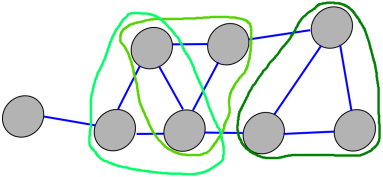
Here, Z is a normalisation constant given by

$$Z = \sum_X \prod_C \phi_C(X_C)$$

to ensure that the distribution $\mathcal{P}[X]$ is correctly normalised.

mvote.ugoe.de/2826

Probabilistic graphical models



The joint distribution is written as a product of potential functions $\phi_C(X_C)$ over the maximal cliques X_C of the graph:

$$\mathcal{P}[X] = \frac{1}{Z} \prod_C \phi_C(X_C)$$

Gibbs distribution

Here, Z is a normalisation constant given by

$$Z = \sum_X \prod_C \phi_C(X_C)$$

to ensure that the distribution $\mathcal{P}[X]$ is correctly normalised.

mvote.ugoe.de/2826

Probabilistic graphical models

Conditional random fields

Distinguishing between observed variables X and target variables Y, in the unnormalized measure

$$\mathcal{P}[X,Y] = \prod_C \phi_C(X_C)$$

we can define a conditional random field as

$$\begin{aligned} \mathcal{P}[Y|X] &= \frac{1}{Z(X)} \prod_C \phi_C(X_C) \\ Z(X) &= \sum_X \mathcal{P}[X,Y] \end{aligned}$$

mvote.ugoe.de/2826

Probabilistic graphical models

Conditional random fields

Distinguishing between observed variables X and target variables Y, in the unnormalized measure

$$\mathcal{P}[X,Y] = \prod_C \phi_C(X_C)$$

we can define a conditional random field as

$$\mathcal{P}[Y|X] = \frac{1}{Z(X)} \prod_C \phi_C(X_C)$$

$$Z(X) = \sum_X \mathcal{P}[X,Y]$$

Compared to the Bayesian models represented in directed graphs, the CRF removes from the model any dependency between the input variables x_i

mvote.ugoe.de/2826

Questions?

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

mvote.ugoe.de/2826

Literature

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