Syllabus

Mon. 9.4.	(1)	0. Introduction
		A. Fundamentals
Mon. 16.4.	(2)	A. Bayesian and Markov Networks
		B. Inference
Mon. 23.4.	(3)	B.1. Exact Inference
Mon. 30.4.	(4)	B.2. Exact Inference / Join Tree
Mon. 7.5.	(5)	(ctd.)
Mon. 14.5.	(6)	B.3. Approximate Inference / Sampling
Mon. 21.5.		— pentecoste break —
Mon. 28.5.	(7)	B.4. Approximate Inference / Propagation
Mon. 4.6.	(8)	B.5 Variational Inference
		C. Learning
Mon. 11.6.	(9)	C.1. Parameter Learning
Mon. 18.6.	(10)	C.2. Parameter Learning with Missing Values
Mon. 25.6.	(11)	C.3. Structure Learning / PC Algorithm
Mon. 2.7.	(12)	C.4. Structure Learning / Search Methods
		D. Representation
Mon. 9.7.	(13)	D. Bayesian and Markov Networks revisited

Inference in propositional logic

propositional logic:

- we can infer new statements from a given set of statements
- example:

$$\left.\begin{array}{l}\text{it rains} \rightarrow \text{the lawn is wet} \\ \text{the lawn is not wet}\end{array}\right\} \Rightarrow \text{it does not rain}$$

- knowledge base is often represented as a set of rules $A \rightarrow B$.
- there exist efficient algorithms for inference (resolution mechanism, rule chaining, etc.; see, e.g., [CGH97, p. 28ff])
- variables are mapped to $\{0, 1\}$ (truth values).

How can we add information about uncertainties?

Let's look at 3 different mechanisms:

1. Rule-based inference with uncertainties (Fuzzy logic)
2. Inference using functional dependencies (most of supervised machine learning)
3. Inference using the joint probability distribution (Bayesian networks)

1. Rule-based inference with uncertainties (1/3)

propositional logic with uncertainties:

- variables are mapped to $[0, 1]$ (certainties).
- rules are associated with a certainty $x \in [0, 1]$ (uncertain implications; generalized rules):

$$A \to B \text{ with certainty } x.$$

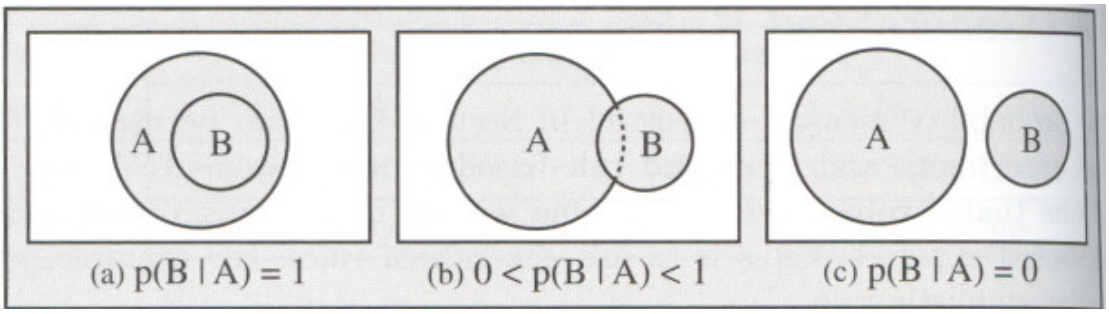


Figure 1: Types of uncertain implications [CGH97, p. 86].

1. Rule-based inference with uncertainties (2/3)

- combination: how can certainties be combined?

  I take a cup of coffee $\to$ I will stay awake during lecture with certainty 0.5

  I take a walk $\to$ I will stay awake during lecture with certainty 0.8

  I take a cup of coffee and I take a walk

  $\Rightarrow$ I will stay awake during lecture with certainty . . . ?

- chaining: how can certainties be chained?

  I take a cup of coffee $\to$ I will stay awake during lecture with certainty 0.5

  I stay awake during lecture $\to$ I will get good marks in the exam with certainty 0.8

  I take a cup of coffee

  $\Rightarrow$ I will get good marks in the exam with certainty . . . ?

1. Rule-based inference with uncertainties (3/3)

- abduction: how can certainties be used in modus tollens?

  I take a cup of coffee $\rightarrow$ I will stay awake during lecture with certainty 0.5

  I do not stay awake during lecture.

  $\Rightarrow$ I did not take a cup of coffee with certainty . . . ?

- Functions for combining and chaining certainties require ad-hoc adjustments and assumptions and thus are problematic (see [Nea90]).

- Systems using uncertain implications:
  - PROSPECTOR [DHN76]
  - MYCIN [BS84]

2. Inference using functional dependencies (1/4)

- The main task in data mining / statistical modelling is predictive modelling, i.e., the prediction of
  - the value of a continuous target variable (regression) or
  - the state of a categorical target variable (classification)

  based upon the knowledge of the values or states of other variables (predictor variables).

- In case of a categorical target variable, one often is more interested in predicting the **probabilities of the different states** instead of the **most-probable state** only (e.g., if cost info is available etc.).

2. Inference using functional dependencies (2/4)

- Mathematically the prediction model is represented as a function from the domains of predictor variables in the domain of the target variable or $[0, 1]$ respectively. If X_i are the domains of the predictor variables ($i \in I$) and Y is the domain of the target variable, then a model is described by

$$f : \prod_{i \in I} X_i \to Y \quad \text{[continuous target with domain } Y\text{]}$$

or

$$f : \prod_{i \in I} X_i \to [0, 1]^Y \quad \text{[probabilities for the states } Y \text{ of a categorical target]}$$

- To make the task of learning such functional dependencies from data feasible, the class of admissible functions is restricted (linear models, neural nets, decision trees, etc.).

- Inference is done by evaluating the function f for given values of the predictor variables.

2. Inference using functional dependencies (3/4)

- Example: detection fraudulent transactions in banking.

  $X_1 = [0, 24]$: Time of day of transcation.

  $X_2 = [0, 10000]$: Amount of transaction (in EUR).

  $Y = \{\text{yes}, \text{no}\}$: transaction is fraudulent.

$$f(x_1, x_2) := \frac{1}{1 + e^{10 + 10 \cdot sin^2(x_1 \cdot \pi / 24) - 0.0005 \cdot x_2}}$$

f gives $p(Y = \text{yes})$.

As Y is binary, trivially

$$p(Y = \text{no}) = 1 - p(Y = \text{yes})$$

For example,

$$f(10, 100) = 4.2 \cdot 10^{-9}$$
$$f(2, 10000) = 0.0034$$

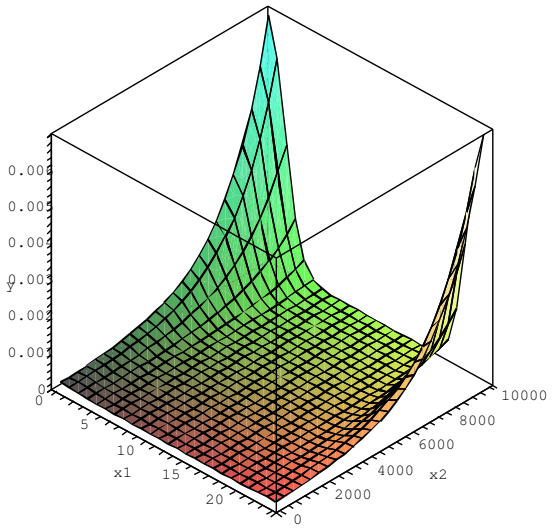


Figure 2: Predicted probability y for given predictors x_1, x_2.

2. Inference using functional dependencies (4/4)

Limitations:

- a separate model for each variable (target) has to be learned
- special methods to deal with varying constellations of predictor variables have to be used (missing values)
- learning algorithms have to be derived for each family of functions.

3. Inference using the Joint Probability Distribution

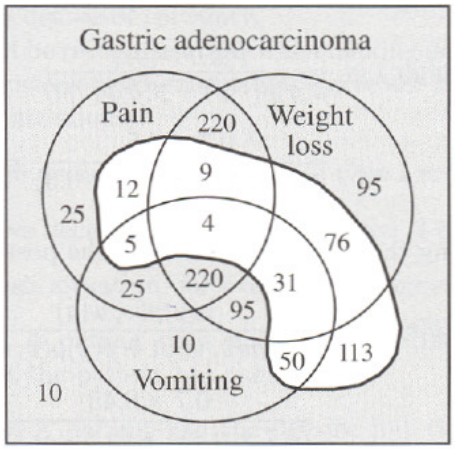


Figure 3: Number of patients classified by one disease (Adeno) and three symptoms [CGH97, p. 81].

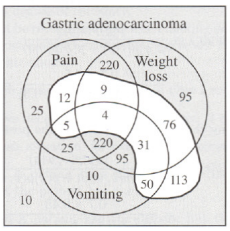
Pain	Y				N			
Weightloss	Y		N		Y		N	
Vomiting	Y	N	Y	N	Y	N	Y	N
Adeno Y	220	220	25	25	95	95	10	10
N	4	9	5	12	31	76	50	113

We can infer probabilties for patients suffering from adeno from observed symptoms (V = vomiting, W = weightloss, P = pain):

- If the state of none of the symptoms is known, then

$$p(\text{adeno}=Y) = \frac{700}{1000} = 0.7$$

3. Inference using the Joint Probability Distribution



Pain	Y				N			
Weightloss	Y		N		Y		N	
Vomiting	Y	N	Y	N	Y	N	Y	N
Adeno Y	220	220	25	25	95	95	10	10
N	4	9	5	12	31	76	50	113

Figure 3: Number of patients classified by one disease (Adeno) and three symptoms [CGH97, p. 81].

We can infer probabilties for patients suffering from adeno from observed symptoms (V = vomiting, W = weightloss, P = pain):

- If $V = Y$, and we know there are no other symptoms (W = N, P = N), then

$$p(\text{adeno=}Y|V = Y, W = N, P = N) = \frac{10}{10 + 50} = 0.17$$

- If $V = Y$, and we do not know the states of the other symptoms, then

$$p(\text{adeno=}Y|V = Y) = \sum_{w,q} p(\text{adeno=}Y, W = w, P = q|V = Y)$$
$$= \frac{220 + 25 + 95 + 10}{224 + 30 + 126 + 60} = \frac{350}{440} = 0.80$$

3. Inference using the Joint Probability Distribution (2/3)

In general, it is a bad idea to use the observed frequencies as estimation for the full joint probability distribution (JPD):

- If the number of variables increases, it becomes infeasible to store the JPD (e.g., for 100 binary variables the JPD has $2^{100} \approx 10^{30}$ cells).
- As the JPD has so many parameters, it suffers extremely from overfitting.
- Domain experts often can tell us in advance, that certain variables cannot be related to some other variables.

$\Rightarrow$ Idea: break the JPD down in several smaller conditional probability distributions of a subset of related variables (i.e., factor the JPD).

3. Inference using the Joint Probability Distribution (3/3)

Bayesian Network:

structure: The graph structure encodes the factors of the JPD. Each node represents a variable x, nodes pointing at it represent variables y_i ($i \in \text{fanin}(x)$) on which x depends.

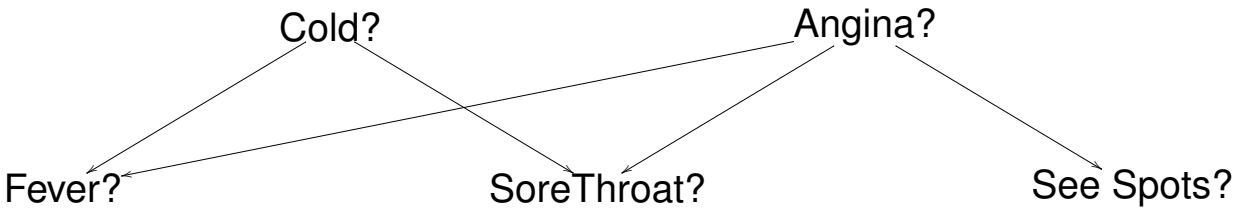


Figure 4: Example Bayesian Network.

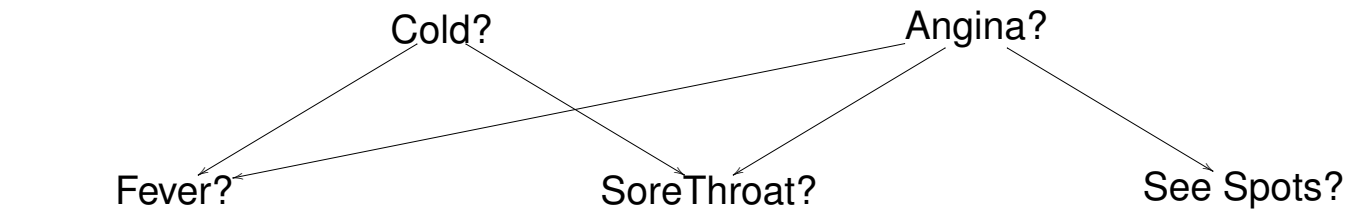
Inference using the Joint Probability Distribution (3/3)

Bayesian Network:

structure: The graph structure encodes the factors of the JPD. Each node represents a variable x, nodes pointing at it represent variables y_i ($i \in \text{fanin}(x)$) from which x depends.

parameters: to each node is attached a conditional probability table

$$p(x|(y_i)_{i\in\text{fanin}(x)})$$



Cold	Y		N	
Angina	Y	N	Y	N
Fever Y	0.8	0.4	0.3	0.01

Cold	Y		N	
Angina	Y	N	Y	N
Sore Th. Y	0.9	0.2	0.7	0.02

Angina	Y	N
Spots Y	0.2	0.001

Figure 4: Example Bayesian Network.

1. What is a bayesian network?
2. Overview
3. Organizational stuff

Exact Inference

For inference, evidence E – i.e., the knowledge of the states of some of the variables – is entered into the net and propagated, s.t. the marginals of the resulting node tables give $p(x|E)$.

To compute this efficiently, special graph structures as the join tree have to be derived.

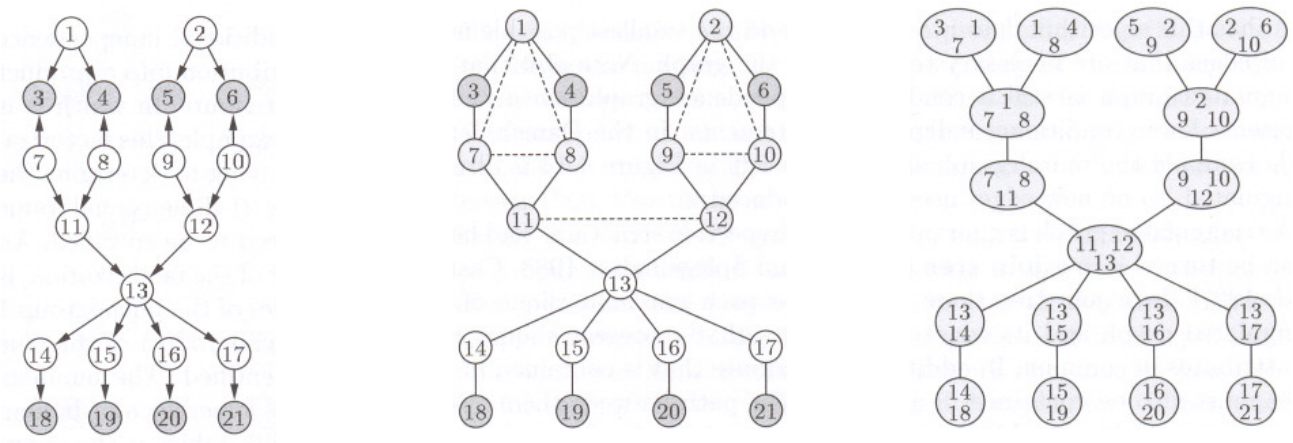


Figure 5: a) Example bayesian network, b) triangulated moral graph, c) join tree [BK02, p. 127, 129].

Approximate Inference

For large or dense networks, exact inference is too expensive to compute.

Approximative results can be established based on stochastic simulation.

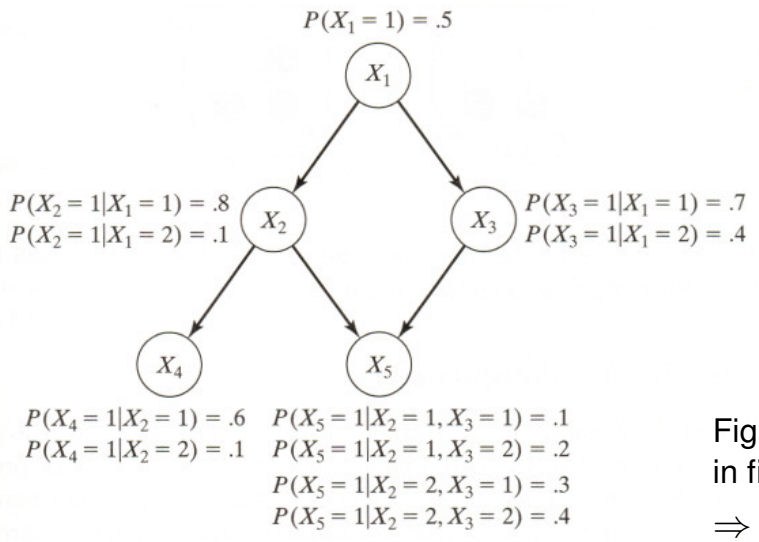


Figure 6: Example bayesian network [Nea03, p. 211].

Case	X_1	X_2	X_3	X_4	X_5
1	1	2	1	2	2
2	1	2	2		
3	1	2	1	2	1
4	2	1	1	1	
5	2	2	1	2	2
6	2	1	2		
7	1	1	1	2	1

Figure 7: Sample drawn from bayesian network in fig. 6.

$\Rightarrow$

$$\hat{p}(X_1 = 1|X_3 = 1, X_4 = 2) = \frac{3}{4}$$

Bayesian Network Analysis

Beneath inference, some other tasks can be accomplished with bayesian networks, as, e.g., generating explanations for specific observations by computing the configuration having maximal probability.

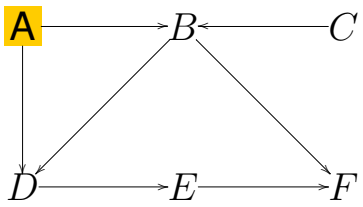


Figure 8: Example bayesian network.

We observe $A = a$. Then we can compute the configuration $B = b, C = y, D = d, \ldots$ with

$$p(A = a, B = b, C = c, D = d, \ldots)$$

maximal among all $p(A = a, \ldots)$. Thus, $B = b, C = c, D = d, \ldots$ is the most likely explanation for observing $A = a$.

Manually Building Models

In many application domains, domain knowledge can be used to specify the structure and / or the parameters of bayesian networks.

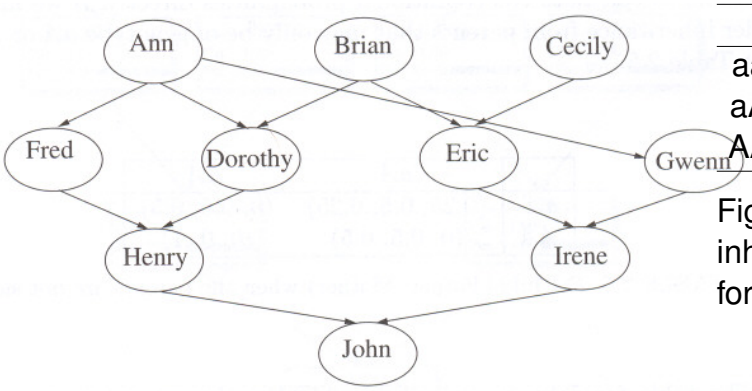


Figure 9: Genealogical structure for the horses in the studfarm example [Jen01, p. 47].

	aa	aA	AA
aa	(1, 0, 0)	(0.5, 0.5, 0)	(0, 1, 0)
aA	(0.5, 0.5, 0)	(0.25, 0.5, 0.25)	(0, 0.5, 0.5)
AA	(0, 1, 0)	(0, 0.5, 0.5)	(0, 0, 1)

Figure 10: p(Child | Father, Mother) for genetic inheritance. The numbers are the probabilities for (aa, aA, AA) [Jen01, p. 47].

Learning Parameters

If we know which variables may influence with others (the structure), but not the exact quantities (the parameters), and we have data, we can estimate the parameters from data.

If data is rare, we need background knowledge to estimate parameters (bayesian estimation).

Case	Cold	Angina	Fever	Sore Throat	See Spots
1	Y	N	Y	N	N
2	N	Y	Y	N	Y
3	Y	N	N	Y	N
4	Y	Y	N	N	N
5	N	N	Y	Y	Y
6	N	Y	Y	Y	N
7	Y	Y	N	N	Y
⋮	⋮	⋮	⋮	⋮	⋮

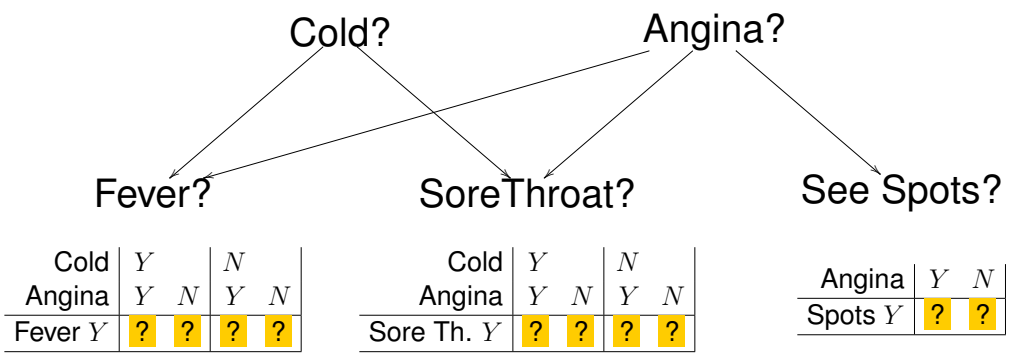


Figure 11: Bayesian network structure with unknown parameters.

Learning Structure

Learning structure requires

- the specification of a model selection criterion as well as
- a search procedure over a subspace of possible graph structures.

Case	Cold	Angina	Fever	Sore Throat	See Spots
1	Y	N	Y	N	N
2	N	Y	Y	N	Y
3	Y	N	N	Y	N
4	Y	Y	N	N	N
5	N	N	Y	Y	Y
6	N	Y	Y	Y	N
7	Y	Y	N	N	Y
⋮	⋮	⋮	⋮	⋮	⋮

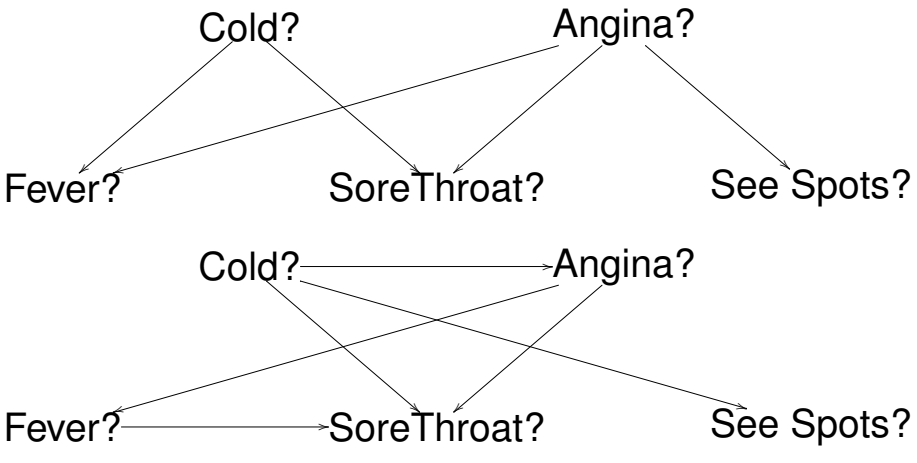


Figure 12: Different bayesian network structures.

Syllabus

Mon. 9.4. (1) 0. Introduction

A. Fundamentals

Mon. 16.4. (2) A. Bayesian and Markov Networks

B. Inference

Mon. 23.4. (3) B.1. Exact Inference
Mon. 30.4. (4) B.2. Exact Inference / Join Tree
Mon. 7.5. (5) (ctd.)
Mon. 14.5. (6) B.3. Approximate Inference / Sampling
Mon. 21.5. — pentecoste break —
Mon. 28.5. (7) B.4. Approximate Inference / Propagation
Mon. 4.6. (8) B.5 Variational Inference

C. Learning

Mon. 11.6. (9) C.1. Parameter Learning
Mon. 18.6. (10) C.2. Parameter Learning with Missing Values
Mon. 25.6. (11) C.3. Structure Learning / PC Algorithm
Mon. 2.7. (12) C.4. Structure Learning / Search Methods

D. Representation

Mon. 9.7. (13) D. Bayesian and Markov Networks revisited

1. What is a bayesian network?
2. Overview
3. Organizational stuff

Text books

- Finn V. Jensen, Thomas Nielsen (^{2}2007):
  Bayesian networks and decision graphs, Springer.
- Richard E. Neapolitan (2003):
  Learning Bayesian Networks, Prentice Hall.
- Daphne Koller, Nir Friedman (2009):
  Probabilistic Graphical Models: Principles and Techniques, MIT Press.
- Adnan Darwiche (2009):
  Modeling and Reasoning with Bayesian Networks, Cambridge University Press.
- Enrique Castillo and José Manuel Gutiérrez and Ali S. Hadi (1997):
  Expert Systems and Probabilistic Network Models, Springer.
- Christian Borgelt and Rudolf Kruse (2002):
  Graphical Models, Wiley.

Bayesian Networks Software

open source:

- libpgm (https://github.com/CyberPoint/libpgm, 3/2015)
- PyOpenPNL (https://github.com/PyOpenPNL/PyOpenPNL, 4/2017), wrapping OpenPNL (C++, http://sourceforge.net/projects/openpnl), v1.0, 7/2006, V. Eruhimov & K.Murp
- pomegranate (https://github.com/jmschrei/pomegranate, v0.4.0, 3/2016)
- Python Bayesian Network Toolbox (https://github.com/achille/pbnt, 2005)
- BayesPy (https://bayespy.org, v0.5.11, 9/2017)
- OpenBugs (http://openbugs.info, v3.2.3, 3/2014, D. Spiegelhalter).

open source, based on commercial software:

- Bayes Net Toolbox (Matlab, https://github.com/bayesnet/bnt, 10/2007, K. Murphy), based on Matlab.

commercial software:

- Bayes Server (https://www.bayesserver.com, v7.25, 9/2017)
- Hugin (http://www.hugin.com), BayesiaLab, ...

Exercises and tutorials

- There will be a weekly sheet with two exercises handed out **each Monday** in the lecture.
  1st sheet will be handed out **Wed. 11.4.** in the tutorial.
- Solutions to the exercises can be submitted until **next Monday noon**,
  1st sheet is **due Mon. 16.4.**
- Exercises will be corrected.
- Tutorials **each Wednesday 16–18**,
  1st tutorial at **Wed. 11.4.**
- Successful participation in the tutorial gives up to 10% bonus points for the exam.

Exam and credit points

- There will be a written exam at end of term (2h, 4 problems).
  - The exam is open-book.
- The course gives 6 ECTS (2+2 SWS).
- The course can be used in
  - Angewandte Informatik MSc. / Informatik / Gebiet KI & ML
  - Data Analytics MSc. / Methodological Specialization
  - IMIT MSc. / Informatik / Gebiet KI & ML
  - Wirtschaftsinformatik MSc. / Business Intelligence
  - as well as in any BSc. program.

References

[BK02] Christian Borgelt and Rudolf Kruse. *Graphical Models*. Wiley, New York, 2002.

[BS84] B. G. Buchanan and E. H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Wiley, New York, 1984.

[CGH97] Enrique Castillo, José Manuel Gutiérrez, and Ali S. Hadi. *Expert Systems and Probabilistic Network Models*. Springer, New York, 1997.

[DHN76] Richard O. Duda, Peter E. Hard, and N. Nilsson. Subjective bayesian methods for rule-based inference systems. In *Proceedings of the 1976 National Computer Conference (AFIPS)*, volume 45, pages 1075–1082, 1976.

[Jen01] Finn V. Jensen. *Bayesian networks and decision graphs*. Springer, New York, 2001.

[Nea90] Richard E. Neapolitan. *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*. Wiley, New York, 1990.

[Nea03] Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.