Welcome to study Expert Systems!

In this course, you will learn

- general knowledge of expert systems and their possibilities
- how to implement a system for your own problem
- how to compare different systems and select the best one
- several metaskills like new learning methods, groupworking, problem solving, information retrieval, ...

Course details

5 ects master-level course

Classes: Wed 14-16 D106/B 14.9.-30.11. 2005

Teacher: Wilhelmiina Hämäläinen
(Meeting times: Wed 13.00-14.00, whamalai@cs.joensuu.fi)

Prerequicites: basic knowledge of logics and probability calculus

Performance

- No exam, no demo sessions, but still we will learn!
- Prepare for 12 hours work/week as in normal courses!

1. **Participating classes** (you can drop at most 2/12 classes)
2. **Presentations** about main expert systems: A student pair gives a 30 min presentation about some expert system with some **lecture material** for other students. They should also design some **exercise task** about the topic and check the solutions. Each exercise task should take approximately 2 hours work.
3. **Exercises**: In the beginning of course exercises about introduction lectures and all student presentations. Weekly exercises take approximately 6 hours work in the beginning of course.
4. **Learning diary**: Each student writes a personal learning diary and returns a piece of it weekly. The lecturer will give feedback next week. The learning diaries are evaluated in the end of course.
5. **Projectworks** The projectworks are made in groups of three students (other than in the presentations). The projects are performed in several phases: specification (topic and data), design (how to implement), implementation and testing

Evaluation

Projectworks 40%
Presentations 20%
Learning diaries 20%
Exercises: 20%

Presentations will be evaluated by students and the teacher (according to how much you learnt)
For learning diaries, we will fix the detailed criteria together!

Learning diary

In the beginning:

- What are your expections?
- What do you know beforehand?
- What is your motivation?

Every week:

- What you have learnt and what you have not?
- Set yourself learning goals, what you should learn or find out! Check also, if you have reached your previous learning goals!
- All experiences, feelings, changes in motivation are also important!
- Explain the new or difficult things with your own words! Try to relate them to your previous knowledge, and invent new applications

In the end:

- Combine all pieces of knowledge into one and draw the final picture (by words or drawing a concept map)
- Evaluate your whole learning process during the course: Did you reach your goals? Are you satisfied or not?

In addition:

- You can write or draw anything you wish – the style is totally free
- Return your piece of learning diary weekly on paper, so that your teacher can give you individual feedback and answer your questions
- Save old learning diaries for final evaluation

Preliminary schedule

14.9. Introduction to Expert systems, course overview and work allocation
21.9. Basic types of ESs, rule-based, functional and probabilistic systems
28.9. ESs, student presentations
5.10. ESs, student presentations
12.10. ESs, student presentations
19.10. ESs, last student presentations. Comparing and evaluating systems (lecturer). Beginning projectworks.
26.10. Work specifications (topic + data), short presentations.
2.11. Work plans (method + how to implement), short presentations.
9.11. Detailed plans with class diagrams and main algorithms.
16.11. Implementations, short presentations?
23.11. About testing, designing test plans. Other evaluation (restrictions, time and space requirements etc.)
30.11. Supersession (4 h), final presentations

What are expert systems?

Several definitions:

"A computer system which emulates the decision-making ability of a human expert"

"A computer system capable of giving advice in a particular knowledge domain"

"A model of the expertise of the best practitioners of the field"

- application of *Artificial Intelligence* (simulates intelligent behaviour)
- roots in *Decision Analysis* (utility of decision)
- also related to *Machine Learning* (learning models from data) and *Data Mining* (discovering new patterns in data)
- performs at the level of an expert (to assist an expert or even instead of an expert)
- have been designed for a narrow problem domain
- can be implemented as an *intelligent agent* = a part of software, which monitors the system and offers help, when it considers it is needed

What they can do?

Several types of tasks

- Diagnosis: finding faults in the system or diseases of living organisms, e.g. printer fault detection system, automatic medical advicors in the net, ...
- Monitoring: continuous interpretetion of signals, e.g. breating-support machines in hospitals, controlling systems in industry
- Planning: producing a sequence of actions for achieving a particular goal (e.g. designing an ideal curriculum for a school, interactive guides for repairing a car, navigating an exploration robot in Mars)
- Instruction (intelligent tutoring systems): especially for practising physical skills like plane simulators, but nowadays in all levels, especially in distance learning
- Prediction: forecasting future events (e.g. advicing pricing according to market fluctuations)

Examples

A rule-based system for automobile trouble-shooting

The system consist of simple "if ... then" rules, with uncertainty measures. For example:

"If the result of trying the starter is the car cranks normally and a gas smell is not present while trying the starter Then the gas tank is empty with 90% confidence"

The on-line demo can be found on:
`http://www.expertise2go.com/webesie/car/`
The system is described on
`http://www.expertise2go.com/webesie/`

Bayesian automobile trouble shooting

The idea is to diagnosize the most probable reasons for automobile start-up problems and evaluate the probability that the engine works, when we fix a certain component. For example if we observe that the gauge is empty (whether or not there is fuel), we can calculate the probability that the car starts after we have filled the tank.

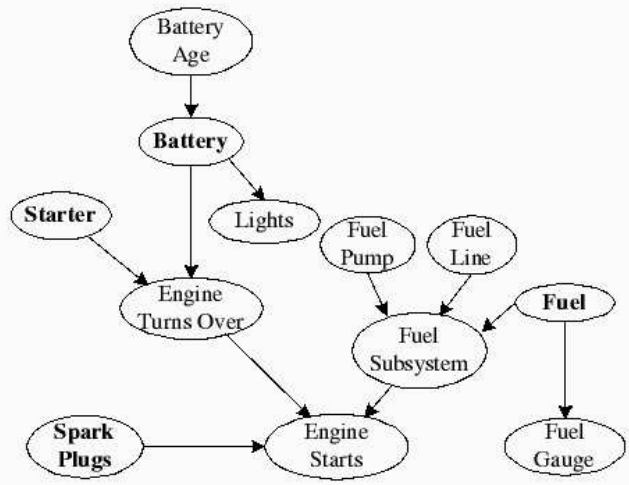
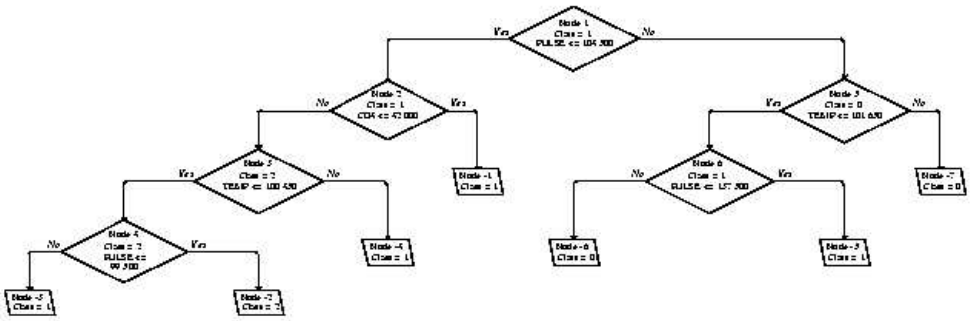


Figure 5: A Bayesian network for an automobile troubleshooting problem.

This example is from Breese, J.S., Heckerman, D. and Rommelse, K.: Troubleshooting Under Uncertainty. Technical report MSR-TR-94-07, 1994. `http://www.research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-94-07`

A decision tree for medical diagnosis

An HIV-positive patient has come to the emergency department for care. What is the urgency of the visit? The nurse should classify patients as 0=emergent, 1=urgent, 2=non-urgent.



This example is from Lewis, R.J.: An Introduction to Classification and Regression Tree (CART) Analysis. Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, California, 2000. `http://www.saem.org/download/lewis1.pdf`.

Why ESs are useful?

- help if expertise is scarse, expensive or unavailable – training new human experts is time-consuming and expensive
- save time, because they are fast and the same system can serve several users parallelly
- do not forget or make human errors
- are consistent: similar situations are always handled in the same way
- can combine knowledge of multiple human experts

However, ESs

- are totally dependent on their knowledge base
- usually manage only one way of reasoning
- don't have common sense!
- cannot adapt to new situations creatively like people can
- don't recognize when no answer exists or when the problem is outside teir area of expertise!

History

- medical software tools begun to emerge during 1970's, e.g. MYCIN (Stanford University 1976) for aiding physicians in diagnosing and treating patients with infectious blood diseases
- the first systems were simple rule-based systems defined by human experts
- Revolution in the late 80's: Bayesian networks
- in 1990's several new applications and methods
- nowadays "AI-tools" – general make-it-yourself expert systems
- Evolution: rule-based systems $\Rightarrow$ Uncertain rules $\Rightarrow$ Bayesian networks, neural networks, decision trees, ...
- Tendency from manually constructed "ad hoc models" to *machine learning*, systems, which can learn themselves

System construction

Terminology

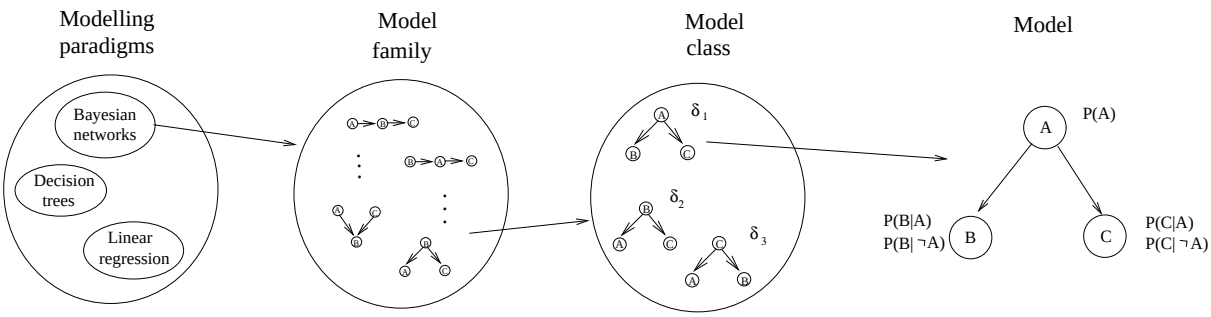
A **model** $\mathcal{M} = (S, \theta)$ = a model structure S and assigned parameter values θ. I.e. it is an instance of a given model structure.

The **model structure** S is a structure which determines the parameters required for constructing a model. Typically the structure consists of variables and their relations (e.g. conditional dependencies).

The **model parameters** θ are assigned numerical values like probabilities in Bayesian network or linear coefficients (above α, β, γ) in linear regression.

Modelling paradigm – the general modelling principles used. The modelling paradigm consist of basic definitions, assumptions, and techniques for constructing and using certain kind of models.

Hierarchy of concepts



Modelling paradigms like Bayesian networks, decision trees and linear regression describe the general modelling principles used.

A model family is a set of model structures in the given modelling paradigm. E.g. in Bayesian networks modelling paradigm a model family consists of different graph structures with variable nodes and conditional dependencies (edges) between variables.

A model class is a set of models with a fixed structure but different parameters θ_i. In the case of Bayesian networks the parameters are prior probabilities associated to root nodes and conditional probabilities associated to non-root nodes.

A model is an item in a model class with a fixed structure and fixed parameters.

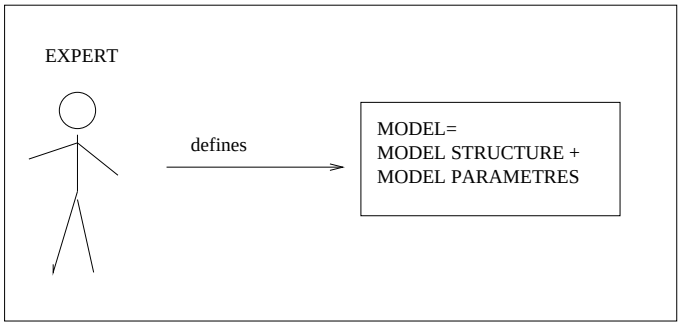
Construction process (from the implementor's point of view)

1. Understanding the domain: interview human experts and read literature. What are the system functionalities? Special utilities or risks?
2. Data management: what and how to collect? Preprocessing?
3. Define the model structure (see below). Especially causal relations are important!
4. Define the model parametres, usually from data.
5. Verification and evaluation: Test the system! How sensitive it is for small changes in the data? (=sensitivity analysis)

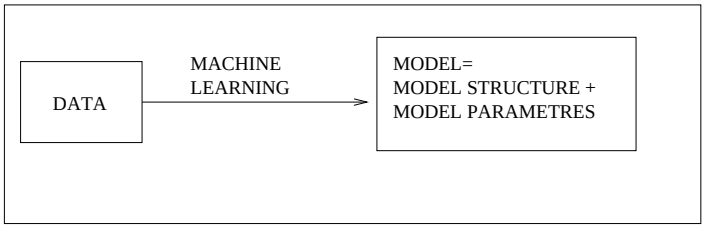
Four approaches

4 APPROACHES FOR ES CONSTRUCTION

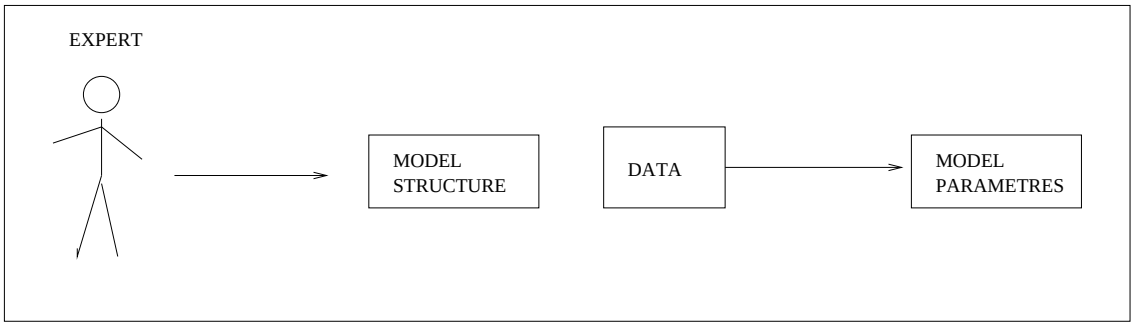
1)



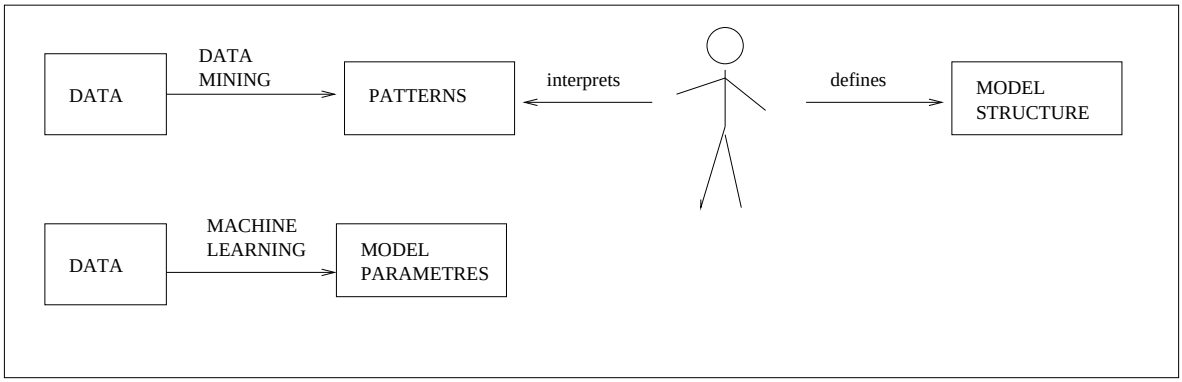
2)



3)



4)



1. Traditional approach: the expert defines both the model structure and the parametres
   - simple and fast method
   - expert knows the general causalities quite well
   - numeric measures (parametres) are hard to estimate
2. Machine Learning-approach: the whole model is defined by ML techniques
   - simple model structures can be learnt fast, but learning a graph-structure is generally NP-hard problem! (intractable for large data sets)
   - several model structures can be equally plausible – recognizing causal relationships is very hard!
   - accurate numeric parametres, if the data set was good (large and representative)

3. Compromise approach: The expert defines the model structure, and the parametres are learnt from data
   - combines the expert's knowledge on general causalities and computational means for defining parametres
   - the data and expert's knowledge can be contrary!
4. Iterative approach (See: Descriptive and predictive modelling by WH): 1) Data mining reveals patterns in the data. An expert defines the model structure according to these patterns. 2) The model parametres are learnt from data.
   - data and model structure are now consistent
   - can be applied to new and poorly known domains

Task: What other advantages and disadvantages do you invent in each of the approaches?

Remarks

- Likelihood $\neq$ importance (utility): e.g. the most probable medical diagnosis is:"There is nothing wrong in you, you just have a cold", but the 3rd probable (not reported) diagnosis is lung cancer.
- Don't test with the same data set you have used for model construction – any model will then work perfectly! $\Rightarrow$ distinct training and test sets
- In every modelling paradigm there is some *bias* – they suit better for some problems than others. Become conscious of the bias and select the best one!