

Building Probabilistic Expert Systems

Probabilistic Expert Systems

- A model for diagnosis
- value of information
- conditional entropy

Building Expert Systems

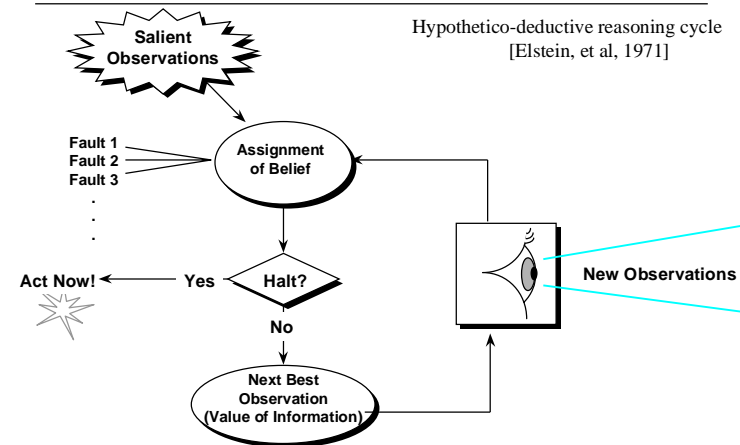
- Assessing Feasibility
- Model Assessment

(Friday:)

- Causal independence models
- Noise in probability assessment

Diagnostic Reasoning

Hypothetico-deductive reasoning cycle
[Elstein, et al, 1971]



Diagnosis

Assigning belief

Infer the $P(\text{Disease}|\text{Symptoms})$ from a joint distribution over the same

Recommending observations...

Making Decisions

One way to decide:

Examine effect on treatment *decisions*

Can we treat better after observing more information, than before?

A model for treatment: maximize expected utility of treatment decisions.

Assume:

A patient has exactly one of N diseases

The doctor treats the patient using one of M treatments.

The outcome can be measured using a utility function $U(D, T)$.

Best treatment decision maximizes:

$$\max_T \sum_{O,D} P\{D, O\} U(D, T)$$

Value of Information

Value of information:

Difference in the expected utility before and after an observation.

$$\sum_O P\{O\} \max_T \sum_D P\{D|O\} U(D,T) - \max_T \sum_{O,D} P\{D,O\} U(D,T)$$

Does this change in value if observation doesn't change the treatment?

Other VOI techniques

Ben-Bassat:

Expected 'distance' in joint distribution due to observation.

$$\sum_O \|P\{D|O\} - P\{D\}\|_M P\{O\}$$

Entropy

Select observation of maximum mutual information (expected change in entropy).

Entropy: $H(D) = -\sum_D P\{D\} \ln P\{D\}$

Change: $E[H(D|O) - H(D)] = \sum_O H(D|O) P\{O\} - H(D)$

Two Cool Things

Conditional KL distance:

$$\begin{aligned} E[H(D|O) - H(D)] &= \sum_O H(D|O)P\{O\} - H(D) \\ &= -\sum_{O,D} P\{D,O\} \ln P\{D|O\} + \sum_D P\{D\} \ln P\{D\} = -\sum_{O,D} P\{D,O\} \ln P\{D|O\} + \sum_{O,D} P\{D,O\} \ln P\{D\} \\ &= -\sum_O P\{O\} \sum_D P\{D|O\} \ln \frac{P\{D|O\}}{P\{D\}} \end{aligned}$$

Conditional Entropy

$$\begin{aligned} E[H(D|O)] &= -\sum_{D,O} P\{D,O\} \ln P\{D|O\} \\ &= -\sum_{D,O} P\{D,O\} (\ln P\{D,O\} - \ln P\{O\}) = H(D,O) - H(O) \end{aligned}$$

Application Development

Assessing Feasibility

Economic Model

User Model

Scope

Clarity and Telephone Test

Expertise

Building Models

What works...

Consulting

Soapbox I (Dilbert hat)

KI Experience:

"Hey look, we can create a knowledge base for X!"

Not a good business strategy

Revised strategy:

Suppose that you can solve the described problem perfectly by any means::

Does it make economic sense?

Does the solution integrate into the user work flow? (medical infomatics)

ONLY then, begin to think about assessing the feasibility of the project...

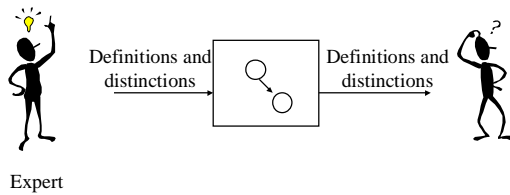
The most critical factors for success have nothing to do with the underlying technology.

Soapbox II: An expensive discovery

It is *easy* to construct belief network applications.

It is *hard* to construct commercially-viable belief network applications.

Expert Systems and Communication



Clarity Test

Are there clear distinctions in the domain?

Imagine that you can ask a clairvoyant any yes/no question using the distinctions in your domain.

Can the clairvoyant answer each question unambiguously with a yes/no answer?

Example:

Is Bill tall?

Is Bill over 1.75 meters tall?

Clarity Test

Two purposes:

Before knowledge acquisition:

Evaluate whether a domain *can* be captured.

During knowledge acquisition:

Continually use for encoding/checking distinctions.

Clarity Test

Telephone Test:

Type I:

Can the intended user call an expert and solve the problem over the phone?

Type II:

Can the intended user call an expert and solve the problem over the phone with the help of a few selected pictures?

The “type” of the telephone test depends on the intended user.

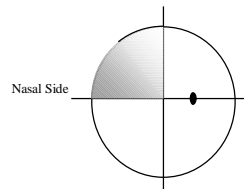
Example: Glaucoma

Easy question for a neuro-ophthamologist:

Does the patient exhibit a "nasal step?"

Easy question for a non-expert:

Does the visual field look like this?



Expertise Sources

Q1: What is the source of expertise for the project?

Q2: Can the expert solve the problem?

Evaluating Expertise:

Won't work

Program managers

Casual "experts"

Might work

Specifications

Best

Diagnostic experts (doctors, mechanics ...)

Expertise Sources

Evaluating Expertise (cont'd)

Might work

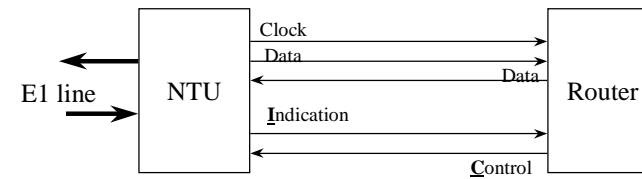
Design specifications

Networks do not fail according to specifications

The specification generally describes what the circuit should do when it is working, not when it fails.

Tend to over-model based on physics of the working device and under-model with respect to failures.

Specifications



Example: X.21

Spec: The C and I leads on an X.21 network are used to establish the data link.

Reality: You can cut the C and I wires and everything works just fine.

Expertise Sources

Might work (continued)

Design engineers

BIT on the GTCP 700 auxillary power unit. Detects approximately 8 problems.

Expert system: 50-60 common problems. Not all are the same as those detected built-in test.

If something was known to fail before hand, they would have designed it better...

Test program

Expertise not readily available? Become the expert!

Difficult/expensive to simulate some failure modes

Expertise Sources IV

Best

Diagnosis experts

Doctors: clinicians, not researchers

Mechanics

Sales engineers

Look for the "buck stops here" expert.

XXX is an expert if:

XXX is always busy.

Other techs say "let's go ask XXX."

Others defer to XXX.

Feasibility Checklist

Economic Model

If we magically solved this problem (using any approach) would we be rich?

User Model

Does the application fit into the user's work flow?

Clarity Test

Does the domain have clear distinctions?

Expertise

Is there a source of expertise?

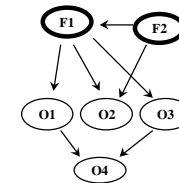
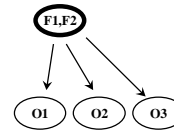
Scoping the Application

Every successful big application starts off as a successful small application.

Make assumptions.

Single Fault vs Multiple Fault.

Discrete-valued vs continuous-valued.



Single Fault Assumption

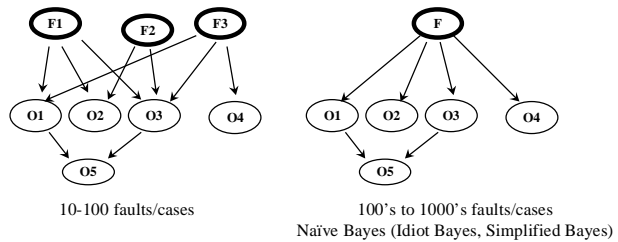
Good for:

Highly maintained systems

Serious faults without a common cause

Why? Heart attack or stroke: go to hospital immediately

Faster inference:



Knowledge Engineering Cycle

Incremental Development Methodology

Main Points

Components of the cycle

Testing is a large part of the development of real systems

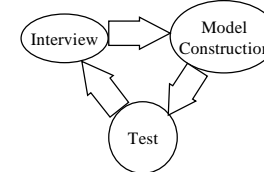
Managing the expert is an important factor in making a successful application.

Clarity matters a lot during knowledge acquisition.

"Exact" probabilities do not.

Knowledge Engineering Cycle

Knowledge Engineering Cycle



Interview:

Assess the knowledge from the expert (10-15%)

Model Construction/Optimization:

Construct the model, optimize the structure (30-40%)

Test:

Make sure the system works.

Demonstrate to the client that the system works.

> 50% of time for a real system.

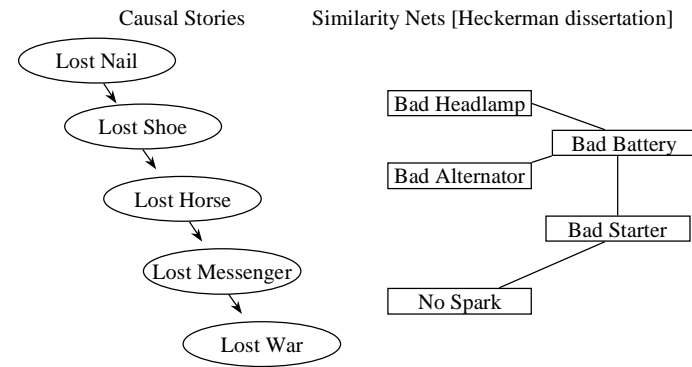
< 20% of time for a prototype.

The Interview

Objectives

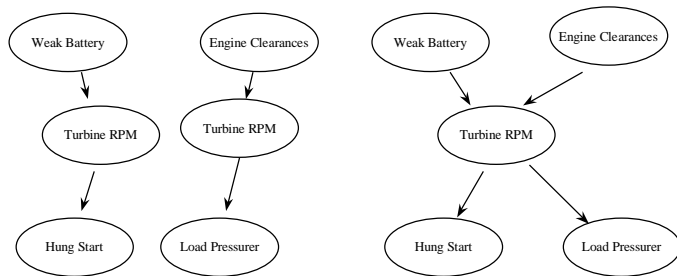
- Identify salient faults and observations.
- Assess variables and mutually exclusive states of variables.
- Assess belief network structure
- Assess probabilities

Feature Assessment Techniques



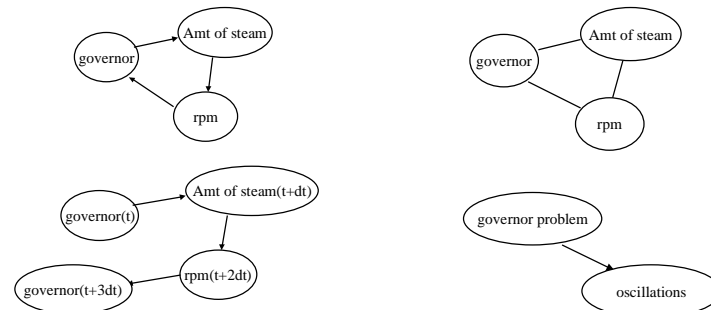
The Interview: Clarity

Clarity problems are the dominant source of structural problems in networks.



Temporal Aggregation

Variable has different meaning at different times
Probability distributions are defined on events, not variables.



Probability Assessments

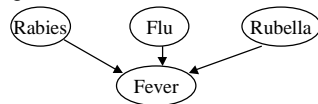
Friday:

Henrion paper: Don't stress about assessing "exact" probabilities for diagnostic models:

Priors only affect the initial phases of question asking and are quickly washed out.

Only orders of magnitude seem to matter for likelihoods.

Minimizing assessments:



Causal independence distributions