Knowledge Engineering with Bayesian Network

Yuqing Tang

Doctoral Program in Computer Science
The Graduate Center
City University of New York
ytang@cs.gc.cuny.edu

December 6, 2010

Outline

1. Introduction
2. Knowledge engineering with Bayesian Networks
3. Summary

Introduction

Knowledge Engineering with Bayesian Network (KEBN)

- KEBN: Overview
- The BN Knowledge Engineering Process
- Model construction
  - Variables and values
  - Graph Structure
  - Probabilities
  - Preferences
- Evaluation

Outline

1 Introduction

2 Knowledge engineering with Bayesian Networks

3 Summary

Knowledge engineering with Bayesian Networks

(Laskey, 1999)

- Objective: Construct a model to perform a defined task
- Participants: Collaboration between domain expert(s) and BN modelling expert(s), including use of automated methods.
- Process: iterate until "done"
  - Define task objective
  - Construct model
  - Evaluate model

KEBN

Production of Bayesian/decision nets for

Decision making: Which policy carries the least risk of failure?

Forward Prediction: Hypothetical or factual. Who will win the election?

Retrodiction/Diagnosis: Which illness do these symptoms indicate?

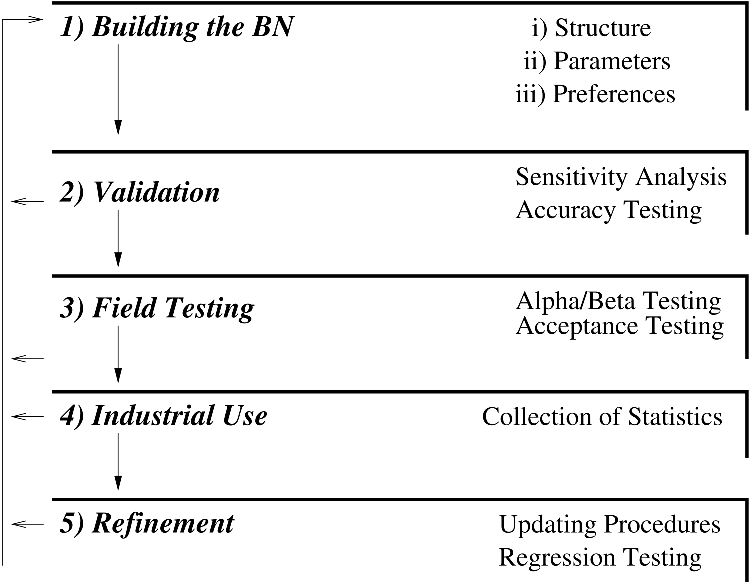
Monitoring/control: Do containment rods needs to be inserted here at Chernobal?

Explanation: Why did the patient die? Which cause exerts the greater influence?

Sensitivity Analysis: What range of probs/utilities make no difference to X?

Information value: What's the differential utility for changing precision of X to ϵ?

KEBN lifecycle model

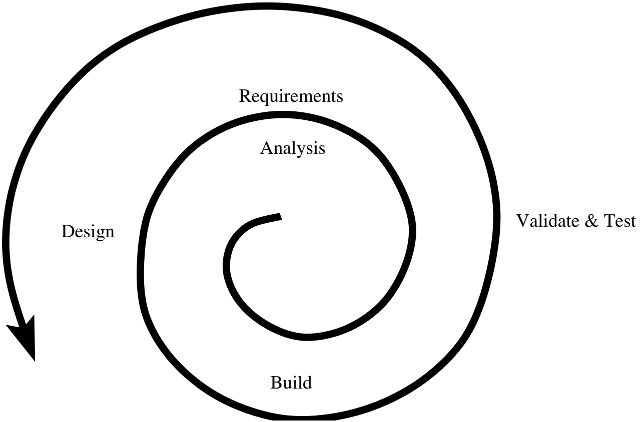


Notes on Lifecycle Model

- Phase 1: Building Bayesian Networks.
  - Major network components: structure, parameters and utilities.
  - Elicitation: from experts, learned with data mining methods, or some combination of the two.
- Phase 2: Evaluation.
  - Networks need to be validated for: predictive accuracy, respecting known temporal order of the variables and respecting known causal structure.
  - Use statistical data (if available) or expert judgement.
- Phase 3: Field Testing.
  - Domain expert use BN to test usability, performance, etc.
- Phase 4: Industrial Use.
  - Requires a statistics collection regime for on-going validation and/or refinement of the networks.
- Phase 5: Refinement.
  - Requires a process for receiving and incorporating change i requests
  - Includes regression testing to verify that changes do not undermine established performance.

KEBN spiral model

From Laskey & Mahoney (2000)
Idea (from Boehm, Brooks): prototype-test cycle



KEBN tasks

For Bayesian Networks, identifying:

- What are the variables? What are their values/states?
- What is the graph structure? What are the direct causal relationships?
- What are the parameters (probabilities)? Is there local model structure?

When building decision nets, identifying:

- What are the available actions/decisions?
- What are the utility nodes & their dependencies?
- What are the preferences (utilities)?

The major methods are:

- Expert elicitation
- Automated learning from data
- Adapting from data

Identifying the Variables

Which are the most important variables?

- "Focus" or "query" variables
  - variables of interest
- "Evidence" or "observation" variables
  - What sources of evidence are available?
- "Context" variables
  - Sensing conditions, background causal conditions
- "Controllable" variables
  - variables that can be "set", by intervention

Start with query variables and spread out to related variables. NB: Roles of variables may change.

Variable values/states

- Variable values must be exclusive and exhaustive
  - Naive modelers sometimes create separate (often Boolean) variables for different states of the same variable
- Types of variables
  - Binary (2-valued, including Boolean)
  - Qualitative
  - Numeric discrete
  - Numeric continuous
- Dealing with infinite and continuous variable domains
  - Some BN software (e.g. Netica) requires that continuous variables be discretized
  - Discretization should be based on differences in effect on related variables (i.e. not just be even sized chunks)

Graphical structure

Goals in specifying graph structure

- Minimize probability elicitation: fewer nodes, fewer arcs, smaller state spaces
- Maximize fidelity of model
  - Sometimes requires more nodes, arcs, states
  - Tradeoff between more accurate model and cost of additional modelling
  - Too much detail can decrease accuracy
- Drawing arcs in causal direction is not "required" BUT
  - Increases conditional independence
  - Results in more compact model
  - Improves ease of probability elicitation
- If mixing continuous and discrete variables
  - Exact inference algorithms only for the case where discrete variables are ancestors, not descendants of continous variables

Relationships between variables I

Types of qualitative understanding can help determine local/global structure

- Causal relationships
  - Variables that could cause a variable to take a particular state
  - Variables that could prevent a variable taking a particular state
- Enabling variables
  - Conditions that permit, enhance or inhibit operation of a cause
- Effects of a variable
- Associated variables
  - When does knowing a value provide information about another variable?
- Dependent and independent variables
  - D-separation tests
  - Which pairs are directly connected?
    Probabilities dependent regardless of all other variables?

Relationships between variables II

Matilda – software tool for visual exploration of dependencies (Boneh, 2002)

- Temporal ordering of variables
- Explaining away/undermining
- Causal non-interaction/additivity
- Causal interaction
  - Positive/negative Synergy
  - Preemption
  - Interference/XOR
- Screening off: causal proximity
- Explanatory value
- Predictive value

Probabilities

- The parameters for a BN are a set of conditional probability distributions of child values given values of parents
- One distribution for each combination of values of parent variables
- Assessment is exponential in the number of parent variables
- The number of parameters can be reduced by taking advantage of additional structure in the domain (called local model structure)

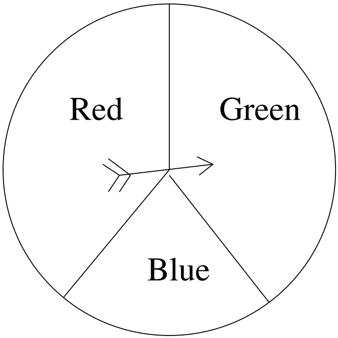
Probability Elicitation

- Discrete variables
  - Direct elicitation: $p = 0.7$
  - Odds (esp. for very small probs): 1 in $10,000$
  - Qualitative assessment: "very high probability"
    - Use scale with numerical and verbal anchors (van der Gaag et al., 1999)
    - Do mapping separately from qualitative assessment
- Continuous variables
  - bi-section method
    - Elicit median: equally likely to be above and below
    - Elicty 25th percentile: bisects interval below median
    - Continue with other percentiles till fine enough discriminations
- Often useful to fit standard functional form to expert's judgements
- Need to discreteize for most BN software

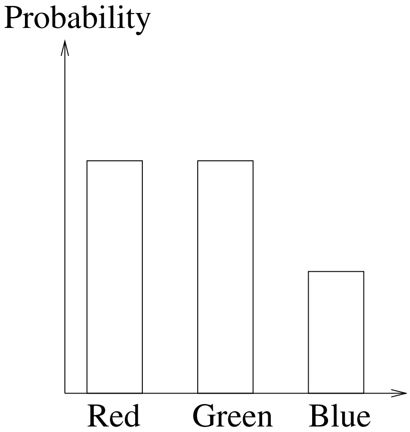
Probability elicitation

Graphical aids are known to be helpful

- pie charts
- histograms



(a)



(b)

Probability elicitation (cont.)

- Combination of qualitative and quantitative assessment
- Automated correction of incoherent probabilities (Hope, Korb & Nicholson, 2002)
  - Minimizing squared deviations from original estimates
- Automated maxentropy fill of CPTs (Hope, Korb & Nicholson, 2002)
- Automated normalization of CPTs (Hope, Korb & Nicholson, 2002)
- Use of lotteries to force estimates (also useful for utility elicitation)

Local model structure

Not every cell in CPT is independent from every other cell. Examples:

- Deterministic nodes
  - It is possible to have nodes where the value of a child is exactly specified (logically or numerically) by its parents
- Linear relationships:

$$X + i = a_0 X_0 + \ldots a_n X_n + \epsilon_i$$

- Logit model (binary, 2 parents):

$$log_2 \frac{P(X_2|X_0, X_1)}{P(\neg X_2|X_0, X_1)} = a + bX_0 + cX_1 + dX_1 X_2$$

- Partitions of parent state space
- Independence of causal influence
- Contingent substructures

Elicitation by Partition

(See Heckerman, 1991)

- Partition state set of parents into subsets
  - set of subsets is called a partition
  - each subset is a partition element
- Elicit one probability distribution per partition element
- Child is independent of parent given partition element
- Examples
  - $P(reportedLoc, sensor\text{-}type, weather)$ independent of sensor type given $weather = sunny$
  - $P(fever = high\ disease)$ is the same for $disease \in \{flu, measles\}$.

Independence of Causal Influence (ICI)

- Assumption: causal influences operate independently of each other in producing effect
  - Probability that C1 causes effect does not depend on whether C2 is operating
  - Excludes synergy or inhibition
- Examples
  - Noisy logic gates (Noisy-OR, Noisy-AND, Noisy-XOR)
  - Noisy adder
  - Noisy max
  - General noisy deterministic function

Noisy-OR nodes

- Adds some uncertainty to logical OR.
  Example: *Fever* true if and only if *Cold*, *Flu*, or *Malaria* is true.
  Assumptions:
  - each cause has an independent chance of causing the effect.
  - all possible causes are listed
  - inhibitors are independent
    E.g.: whatever inhibits *Cold* from causing *Fever* is independent of whatever inhibits *Flu* from causing a *Fever*.
- Inhibitors summarised as "noise parameters".

Noisy-OR parameters

Example

Given $P(Fever|Cold) = 0.4$, $P(Fever|Flu) = 0.8$, and $P(Fever|Malaria) = 0.9$, then noise parameters are $P(\neg Fever|Cold) = 0.6$, $P(\neg Fever|Flu) = 0.2$ and $P(\neg Fever|Malaria) = 0.1$ respectively. Probability that output node is *False* is the product of the noise parameters for all the input nodes that are true.

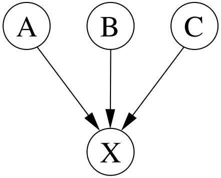
Cold	*Flu*	*Mal*	$P(Fev)$	$P(\neg Fev)$
F	F	F	0.0	1.0
F	F	T	0.9	0.1
F	T	F	0.8	0.2
F	T	T	$1 - 0.02$	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	0.6
T	F	T	$1 - 0.06$	$0.06 = 0.6 \times 0.1$
T	T	F	$1 - 0.12$	$0.12 = 0.6 \times 0.2$
T	T	T	$1 - 0.012$	$0.012 = 0.6 \times 0.2 \times 0.1$

Savings: for a binary noisy-OR node with 9 parents (10 nodes in total)

- CPT requires $1024 = 2^{10}$ parameters;
- noisy-OR requires 11 parameters

Classification Tree Representation

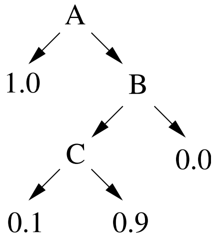
(Boutillier et al. 1996). Example: Suppose node X has 3 parents, A, B, C (all nodes Boolean).



(a)

A	B	C	P(X|A,B,C)
T	T	T	1.0
T	T	F	1.0
T	T	T	1.0
T	F	F	1.0
F	T	T	0.1
F	T	F	0.9
F	F	T	0.0
F	F	F	0.0

(b)



(c)

Savings: $CPT = 8$, tree rep $= 4$ parameters.

Object-oriented BNs

- Facilitate network construction wrt both structure and probabilities
- Allow representation of commonalities across variables
- Inheritance of priors and CPTs

Not widely used.

Decision Analysis

Since 1970s there have been nice software packages for decision analysis:

- Eliciting actions
- Eliciting utilities
- Eliciting probabilities
- Building decision trees
- Sensitivity analysis, etc.

See: Raiffa's Introduction to Decision Analysis (an excellent book!)
Main differences from KEBN:

- Scale: tens vs thousands of parms!
- Structure: trees reflect state-action combinations, not causal structure, prediction, intervention

Eliciting Decision Networks

- Action nodes: What actions can be taken in domain?
- Utility node(s):
  - What unit(s) will "utile" be measured in?
  - Are there difference aspects to the utility that should each be represented in a separate utility node?
- Graph structure:
  - Which variables can decision/actions affects?
  - Does the action/decision affect the utility?
  - What are the outcome variables that there are preferences about?

Model Evaluation

- Elicitation review
  - Review variable and value definition
    - clarity test, agreement on definitions, consistency
  - Review graph and local model structure
  - Review probabilities
    - compare probabilities with each other
- Sensitivity analysis (Laskey, 1993)
  - Measures effect of one variable on another
- Case-based evaluation
  - Run model on test of test cases
  - Compare with expert judgement or “ground truth”
- Validation methods using data (if available)
  - Predictive Accuracy
  - Expected value
  - Kullback-Leibler divergence
  - (Bayesian) Information reward

The need to prototype!

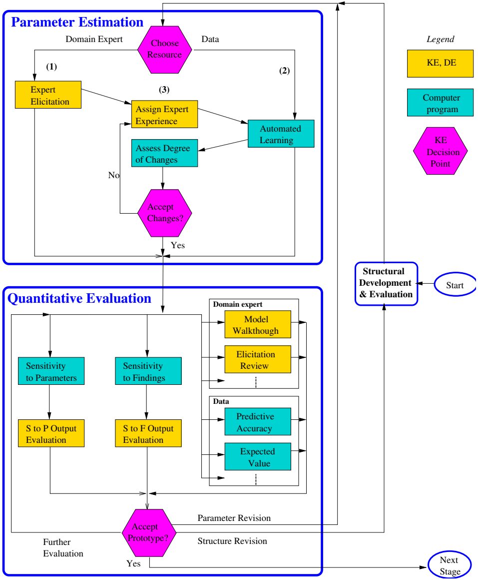
Why prototype?

- It's just the best software development process overall (Brooks). Organic growth of software:
  - tracks the specs
  - has manageable size (at least initially)
- Attacks the comprehensiveness vs. intelligibility trade-off from the right starting point.
- Few off-the-shelf models; prototyping helps us fill in the gaps, helps write the specs

Prototypes

- Initial prototypes minimize risk
  - Don't oversell result
  - Employ available capabilities
  - Simplify variables, structure, questions answered
  - Provide working product for assessment
- Incremental prototypes
  - Simple, quick extension to last
  - Attacks high priority subset of difficult issues
  - Helps refine understanding of requirements/approach

More recent KEBN methodolgies



When BNs are not appropriate

- If the problem is a "one-off", for which there is no data available and any model built won't be used again.
  Bayesian networks might still be used in a one-off modeling process without going through all the KENB knowledge engineering process.
- There are no domain experts, nor useful data.
- If the problem is very complex or not obviously decomposable, it may not be worth attempting to analyze into a Bayesian network.
- If the problem is essentially one of learning a function from available data, and a "black bloc" model is all that is required
  - Artificial neural network, or
  - Other standard machine learning technique

Outline

1 Introduction

2 Knowledge engineering with Bayesian Networks

3 Summary

Summary

- Various BN structures are available to compactly and accurately represent certain types of domain features.
- There is an interplay between elements of the KE process: variable choice, graph structure and parameters.
- No standard knowledge engineering process exists as yet.
- Integration of expert elicitation and automated methods still in early stages.
- There are few existing tools for supporting the BN KE process.
  - We at Monash are developing some! (e.g. VerbalBN, Matilda)

Acknowledgments

Lecture 9 is extracted from `http://www.csse.monash.edu.au/courseware/cse458/L5-4.pdf`, and composed of materials from [Korb and Nicholson, 2003, Chapter 9] with the instructor's own interpretations. The instructor takes full responsibility of any mistakes in the slides.

References I

K. Korb and A. E. Nicholson.
Bayesian Artificial Intelligence.
Chapman & Hall /CRC, 2003.