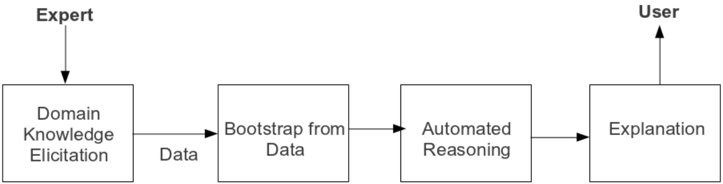
Building Bayesian Network Based Expert Systems From Rules

Saravanan Thirumuruganathan Dr.Manfred Huber

University of Texas At Arlington

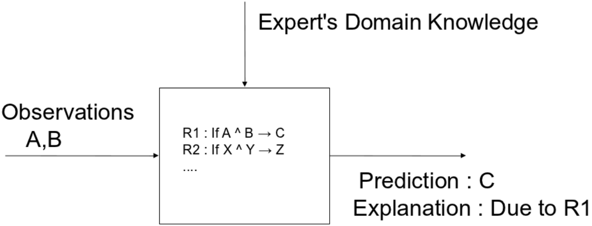
Requirements for Expert Systems

- Modeling of domains with uncertain information
- Bootstrappable from data.
- Automated and theoretically rigorous reasoning.
- Human interaction
  - Elicit domain knowledge from experts
  - Explain predictions to end-users.



Rule based Expert Systems

- Convenient for experts to specify domain knowledge.
- Easy to generate explanation for predictions using expert specified rules.

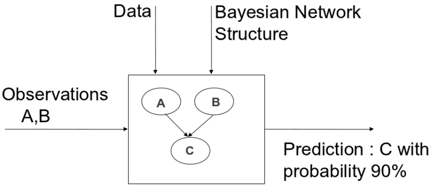


Certainty Factors

- Certainty Factors are one way to incorporate uncertainty into rule based systems.
- If Observation Then Hypothesis (CF1)
  - Change in belief in hypothesis due to observation.
- Pros : Convenient to express domain knowledge and generate explanations.
- Cons : Independence assumptions, inability to handle partial observations or additional data

Bayesian Network based Expert Systems

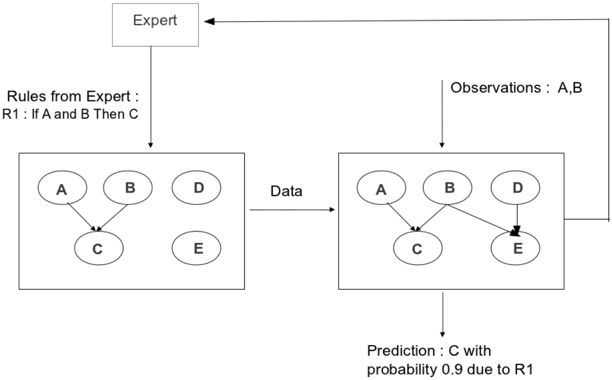
- Probabilistic graphical model
- Theoretically rigorous and automated reasoning
- Hard to explain prediction to users.



Proposed System

- Experts specify domain knowledge using rules
- Bootstrap Bayesian Network from rules
- Learn complete Bayesian Network from data
- Make probabilistically correct predictions
- Explain prediction using rules

Proposed System

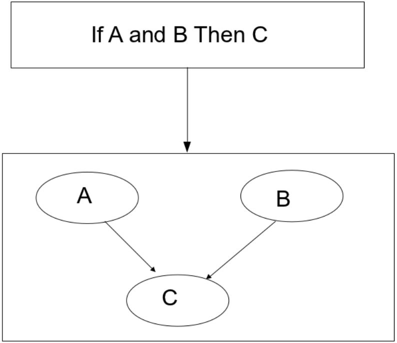


Challenges

- Bootstrapping Bayesian Network
  - Given expert specified rules , how to construct partial Bayesian Network structure and parameters.
- Learning complete Bayesian Network
  - How to learn complete Bayesian Network structure and parameters given additional data
- Generating explanations for predictions
  - How to generate user understandable explanations using expert specified rules.
- Identifying incorrect rules
  - How to identify rules that might potentially be incorrect.

Bootstrapping Bayesian Network Structure

- Rules define a dependency between antecedents and consequents
- Add edges between antecedent and consequent.



Bootstrapping Bayesian Network Parameters

- Heckerman's probabilistic interpretation of Certainty Factors
  - Rules need not independent or exhaustive.
- Rules with Certainty Factors are closer to evidence than probability
- Treat rules from an evidence framework.
  - Antecedents provide evidence to consequent
  - Even when few antecedents does not hold, the rule provide some residual evidence.
- Probability bounds instead of unique probability values

Augmenting Rules with Strength Parameter

- Rules in rule base need not be exhaustive
- Maximize information extracted from rules by making them more expressive.
- Strength parameter for antecedents.
  - R1 : If A (S1) and B (S2) Then C (CF1)
- Allows estimation of evidence for related rules not specified by expert
- If A (0.7) and B (0.8) Then C (0.9)
  - If $\neg$A and B Then C ($0.3 * 0.9 = 0.27$)
    - 0.27 is the residual evidence provided by R1.
  - If A and $\neg$B Then C ($0.2 * 0.9 = 0.18$)
  - If $\neg$A and $\neg$B Then C ($0.3 * 0.2 * 0.9 = 0.054$)

Constraints on Probability Values

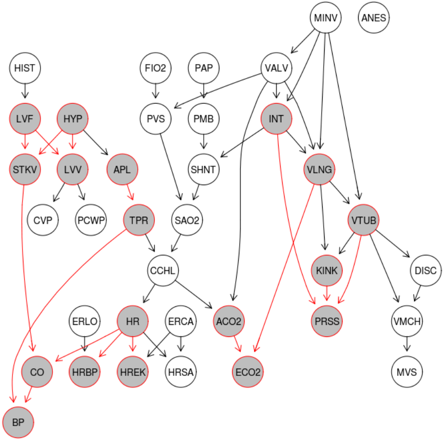
- Rules provide probability bounds instead of unique probability value.
- Strength parameter allows estimation of residual evidence of related rules.
- Rules for X and $\neg X$ provide both lower and upper bounds.
- Related rules provide constraints on possible probability values.
  - Rules which are subset of others
    - If A and B and C Then D, If B and C Then D
    - $\mathrm{P(D|BC) = P(D|ABC)P(A){+}P(D|\neg\ ABC)P(\neg\ A)}$
  - Overlapping Rules
    - If A Then C, If B Then C
    - $\mathrm{P(C|B) = P(C|AB)P(A) + P(C|\neg AB)P(\neg A)}$
    - $\mathrm{P(C|A) = P(C|AB)P(B) + P(C|A\neg B)P(\neg B)}$

Determining CPT of Partial Bayesian Network

- Given a set of bounds, provide a consistent assignment for CPT
- Rules and related rules
  - If A (0.7) and B (0.8) Then C [0.9,1]
  - If $\neg$A and B Then C [0.27,1]
  - If A and $\neg$B Then C [0.18,1]
  - If $\neg$A and $\neg$B Then C [0.054,1]
- Linear Programming formulation
  - Maximize $P(C|AB)+P(C|\neg AB)+P(C|A\neg B) + P(C|\neg A\neg B) + P(\neg C|AB)+P(\neg C|\neg AB) + P(\neg C|A\neg B) + P(\neg C|\neg A\neg B)$
  - subject to
    - $P(C|AB)+P(\neg C|AB) = 1$ etc
    - $1 - P(C|AB) + \delta_{P(C|AB)} = P(C|AB) - 0.9$ etc

Learning Complete Bayesian Network

- Complete network structure and parameters can be learnt from data
- Any network learning algorithm suffices
- Requires significantly less data



Generating Explanations

- Step 1 : Find rules matching prediction
- Step 2 : Sort them by rule's Bayes Factor
  - $\textit{Bayes Factor (If A Then B)} = \frac{P(B|A)}{1-P(B|A)}$
- Step 3 : Generate explanation based on rule with highest Bayes Factor
- Step 4 : Show evidence added by additional rules
- Step 5 : Classify past data with no matching rules under misc

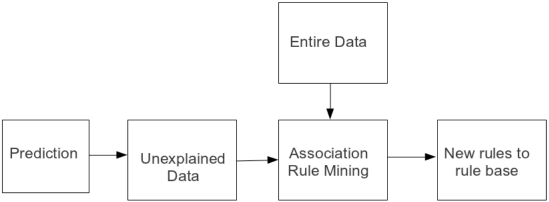
Identifying Incorrect Rules

- Strength, Certainty factors or even entire rule might be wrong
- Detection
  - Learned probabilty violates rule's probability bounds beyond tolerance values
  - Structure learning with different subsets of rules
- If LVF = FALSE (0.8) And HYP = FALSE (0.8) Then STKV = NORMAL (0.9)

LVF	HYP	CF	Bound for STKV=Normal	LP Assignment	Final Probability
F	F	0.9	[0.85,1]	0.93	0.89
F	T	0.18	[0.54,1]	0.77	0.51
T	F	0.18	[0.54,1]	0.77	0.04
T	T	0.04	[0.13,1]	0.57	0.01

Suggesting New Rules

- Inability to explain predictions using existing rules
- Association rule mining on unexplained past data
- Suggest rules with high support and confidence.
- Find rules that can better explain predictions than rules in rule base



Conclusion

- Framework to accept domain knowlege from rules, perform automated reasoning and generate user understandable explanation
- Used evidence framework to convert potentially imprecise certainty factors to probability bounds.
- Consistent assignment using Linear Programming
- Learn complete network using significantly less data.
- Explain predictions using rules.