Lecture 17: Uncertainty 2

Victor R. Lesser
CMPSCI 683
Fall 2010

Today’s Lecture

• How belief networks can be a “Knowledge Base” for probabilistic knowledge.

•How to construct a belief network.

•How to answer probabilistic queries, such as P (Hypothesis | Evidence), using belief networks.

Review of Key Issues *with respect to* Probability Theory

- Basic probability statements include prior probabilities and conditional probabilities over simple and complex propositions.
  - Product rule, Marginalization(summing out) and conditioning
- The axioms of probability specify constraints on reasonable assignments of probabilities to propositions.
  - An agent that violates the axioms will behave irrationally in some circumstances.
- The joint probability distribution specifies the probability of each complete assignment of values to random variables
  - *It is usually far too large to create or use.*

Defining Things in Terms of Joint Probability Distribution

- $P(A \wedge B) = P(A,B)$
- $P(A \vee B) = P(A) + P(B) - P(A,B)$

- $P(A|B) = P(A,B)/P(B)$ when $P(B) > 0$, or
- $P(A,B) = P(A|B)\ P(B)$ (the product rule)
  - $P(A,B,C,D,..) = P(A|B,C, D,..)\ P(B,C, D,..)$

- $P(A) = \sum_i P(A,B_i)$ -- marginalization or summing out

- $P(A) = \sum_i P(A \mid B_i)\ P(B_i)$ -- conditioning

Review of Key Issues *with respect to* Baye' Rule

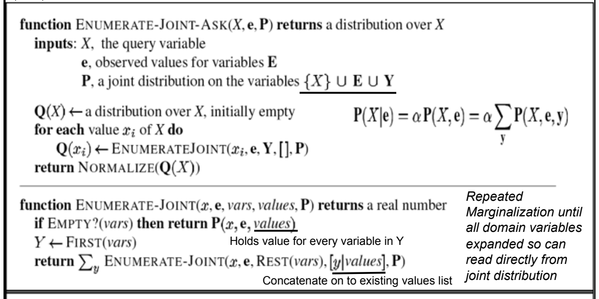
- *Bayes' rule allows unknown probabilities to be computed from known, stable ones.*
- In the general case, combining many pieces of evidence may require assessing a large number of conditional probabilities.
- *Conditional independence brought about by direct causal relationships in the domain allows Bayesian updating to work effectively even with multiple pieces of evidence.*

Bayes' Rule

- Conditional probability from its inverse. $P(B \mid A) = \dfrac{P(A \mid B)P(B)}{P(A)}$
  - Bayes' rule is typically written as: $P(B \mid A)=\alpha P(A \mid B)P(B)$
- Condition on background knowledge E: $P(B|A,E) = (P(A|B,E)\ P(B|E)) / P(A|E)$
- Can also be expressed as : $P(B|E_1,E_2) = (P(E_1, E_1|B)\ P(B)) / P(E_1, E_1)$
  - by seeing $\{E_1, E_2\}$ as A
- *With conditional independence, Bayes' rule becomes:* $P(B|E_1, E_2) = \alpha P(B)\ P(E_1|B)\ P(E_2|B)$
  - $P(E_1, E_2|B) = P(E_1|B)\ P(E_2|B)$ *conditional independence*
  - *Incremental evidence accumulation "*$P(B)\ P(E_1|B)$*" for* $P(B|E_1)$

Probabilistic reasoning

- Can be performed using the joint probability distribution: $\mathbf{P}(X|\mathbf{e}) = \underbrace{\alpha\,\mathbf{P}(X,\mathbf{e})}_{\text{Conditioning}} = \underbrace{\alpha \sum_{\mathbf{y}} \mathbf{P}(X,\mathbf{e},\mathbf{y})}_{\text{Marginalization}}$
- Problem: *How to represent the joint probability distribution compactly to facilitate inference.*
- *We will use a belief network as a data structure to represent the conditional independence relationships between the variables in a given domain.*



```
function ENUMERATE-JOINT-ASK($X$, $\mathbf{e}$, $\mathbf{P}$) returns a distribution over $X$
   inputs: $X$, the query variable
           $\mathbf{e}$, observed values for variables $\mathbf{E}$
           $\mathbf{P}$, a joint distribution on the variables $\underline{\{X\} \cup \mathbf{E} \cup \mathbf{Y}}$

   $\mathbf{Q}(X) \leftarrow$ a distribution over $X$, initially empty
   for each value $x_i$ of $X$ do
      $\mathbf{Q}(x_i) \leftarrow$ ENUMERATEJOINT($x_i$, $\mathbf{e}$, $\mathbf{Y}$, [], $\mathbf{P}$)
   return NORMALIZE($\mathbf{Q}(X)$)

function ENUMERATE-JOINT($x$, $\mathbf{e}$, vars, values, $\mathbf{P}$) returns a real number
   if EMPTY?(vars) then return $\mathbf{P}(x, \mathbf{e}, \underline{values})$
   $Y \leftarrow$ FIRST(vars)
   return $\sum_y$ ENUMERATE-JOINT($x$, $\mathbf{e}$, REST(vars), $\underline{[y|values]}$, $\mathbf{P}$)
```

$$\mathbf{P}(X|\mathbf{e}) = \alpha\,\mathbf{P}(X,\mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X,\mathbf{e},\mathbf{y})$$

Holds value for every variable in Y

Concatenate on to existing values list

Repeated Marginalization until all domain variables expanded so can read directly from joint distribution

Figure 13.4 An algorithm for probabilistic inference by enumeration of the entries in a full joint distribution.

Belief Networks

A major advance in making probabilistic reasoning systems practical for AI has been the development of **belief networks** (also called **Bayesian/probabilistic networks**).

The main purpose of the belief network is to encode the *conditional independence relations* in a domain.

- real domains have a lot of structure due to causality

This makes it possible to specify a complete probabilistic model using far fewer (and more natural/available) probabilities while keeping probabilistic interference tractable.

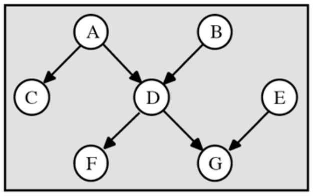
- Considered one of the major advances in AI
  - puts diagnostic and classification reasoning on a firm theoretical foundation
  - makes possible large applications

Belief (or Bayesian) networks

- Set of nodes, one per variable
- Directed acyclic graph (DAG): link represents “direct” influence
- *Conditional probability tables (CPTs): P (Child | Parent$_1$, ..., Parent$_n$)*

Bayesian network example

Example



X = {A=a, B=b, C=c, D=d, E=e, F=f, G=g)

$$p(X) = p(a)p(b)p(c|a)p(d|a,b)p(e)p(f|d)p(g|d,e).$$

P(a,b,c,d,e,f,g)= P(c|a,b,c,d,e,f,g)P(a,b,d,e,f,g)

Productize order c,d,f,g

Conditional independence in BNs

- Each node is conditionally independent of **its non-descendants, given its parents.**
  - Says nothing about other dependencies
- Causality is intricately related to conditional independence.
- Conditional independence is one type of knowledge that we use.

The semantics of belief networks

- Any joint distribution can be decomposed into a product of conditionals:

$$P(X_1, X_2, ..., X_n) = P(X_n|X_{n-1}, ...,X_1)P(X_{n-1}, ...,X_1) = \Pi\, P(X_i|X_{i-1}, ..., X_1)$$

- *Value of belief networks is in "exposing" conditional independence relations that make this product simpler:*

$$P(X_1, X_2, ..., X_n) = \Pi\, P(X_i \mid Parents(X_i))$$

Earthquake example (Pearl)

- You have a new burglar alarm installed.
- *It is reliable about detecting burglary, but responds to minor earthquakes.*
- The neighbors (John, Mary) promise to call you at work when they hear the alarm
  - John always calls when he hears the alarm, but confuses alarm with phone ringing (and calls then also)
  - Mary likes loud music and sometimes misses alarm!
  - Assumption: John and Mary don't perceive burglary directly; they do not feel minor earthquakes
- Given evidence about who has and hasn't called, estimate the probability of burglary.

Conditional probability tables

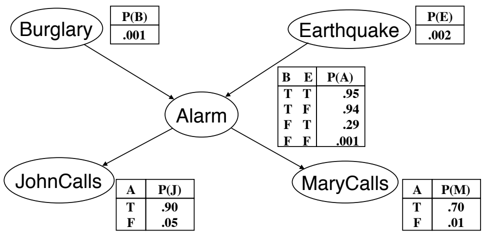
Probability Alarm goes off when burglary and earthquake

Burglary	Earthquake	P(A=True | B,E)	P(A=False | B,E)
True	True	0.950	0.050
True	False	0.940	0.060
False	True	0.290	0.710
False	False	0.001	0.999

How much data is needed to represent a particular problem? How can we minimize it?

Earthquake Example, *Cont'd*

Belief network with probability information:



P(B)
.001

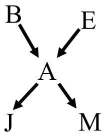
P(E)
.002

B	E	P(A)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J)
T	.90
F	.05

A	P(M)
T	.70
F	.01

Earthquake example cont.



Priors: P(B), P(E)

CPTs: P(A|B,E), P(J|A), P(M|A)

10 parameters in Belief Network but 31 parameters in the 5-variable Joint Distribution

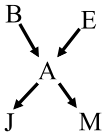
$$P(X_1, X_2, ..., X_n) = \Pi \, P(X_i \mid Parents(X_i))$$

$$\mathbf{P(B,E,A,J,M)= P(B)P(E)P(A|B,E)P(J|A)P(M|A)}$$

Earthquake example cont

- Suppose you need:

  $P(J,E) = \Sigma \, P(J,m,a,b,E)$



- P(J,m,a,b,E) = P(J|m,a,b,E) P(m|a,b,E) P(a|b,E) P(b|E) P(E)
- Conditional independence saves us: P(J,m,a,b,E) = P(J|a) P(m|a) P(a|b,E) P(b) P(E)

Ignorance /Laziness in Example

- Not included
  - Mary is currently listening to music
  - telephone ringing and confusing John
- Factor summarized in
  - Alarm → John calls
  - Alarm → Mary calls
- Approximating Situation
  - eliminating hard-to-get information
  - reducing computational complexity

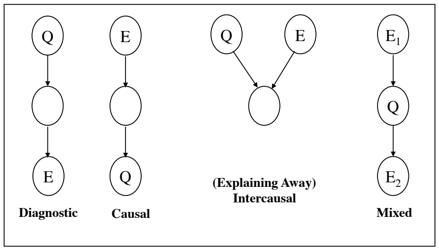
How would they be fit in to bel

Inference in Belief Networks

- BNs are fairly expressive and easily engineered representation for knowledge in probabilistic domains.
- They facilitate the development of inference algorithms.
- They are particularly suited for parallelization
- Current inference algorithms are efficient and can solve large real-world problems.

Reasoning in Belief Networks

Simple examples of 4 patterns of reasoning that can be handled by belief networks. *E* represents an evidence variable; *Q* is a query variable.

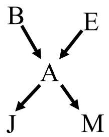


P(Q|E) =?

Types of tasks and queries

- **Diagnostic inferences** (from effects to causes).
  - Given that ***JohnCalls***, infer that *P(Burglary | JohnCalls*) = 0.016

    normalized Sum (E,A,M) P(B,e,a,J,m)
- **Causal inferences** (from causes to effects).
  - Given that ***Burglary***, infer that *P(JohnCalls | Burglary*) = 0.86 and *P*(*MaryCalls* |*Burglary*)= 0.67.



P(B,E,A,J,M)= P(B)P(E)P(A|B,E) P(J|A)P(M|A)

$$\mathbf{P}(X|\mathbf{e}) = \alpha\,\mathbf{P}(X,\mathbf{e}) = \alpha\sum_{\mathbf{y}}\mathbf{P}(X,\mathbf{e},\mathbf{y})$$

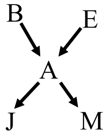
How to do *P(Burglary |JohnCalls)*

- *Bayes Rule* $k*P(J|B)*P(B)$
- *Marginalization* $k*Sum_A P(J,Alarm|B)*P(B)$
- $P(s_i,s_j|d) = P(s_i| s_j,d) P(s_j|d)$ $k*Sum_A P(J|A,B)*P(A|B) *P(B)$
- Case 1: a node is conditionally independent of non-descendants given its parents

  $k*Sum_A P(J|A)*P(A|B) *P(B)$
- *Marginalization* $k*Sum_A P(J|A)*Sum_E P(A,E|B) *P(B)$
- $P(s_i,s_j|d)=P(s_i| s_j,d) P(s_j|d)$ $k*Sum_A P(J|A)*Sum_E P(A|B,E)*P(B|E) *P(B)$
- *case 1 P(B|E)=P(B)* $k*Sum_A P(J|A)*Sum_E P(A|B,E)*P(B) *P(B)$
- *Can read everything off the CPT's*

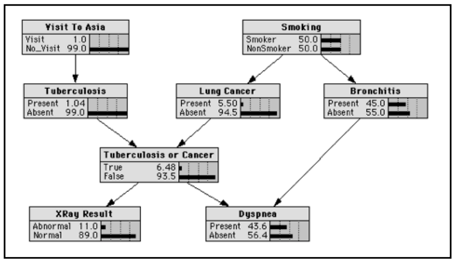
Types of tasks and queries cont.

- **Intercausal inferences** (between causes of a common effect).
  - Given *Alarm*, we have *P(Burglary |Alarm*) = 0.376. But if we add the evidence that *Earthquake* is true, then *P(Burglary |Alarm* ∧ *Earthquake*) goes down to 0.003.
  - *Even though burglaries and earthquakes are independent, the presence of one makes the other less likely.* This pattern of reasoning is also known as **explaining away.**

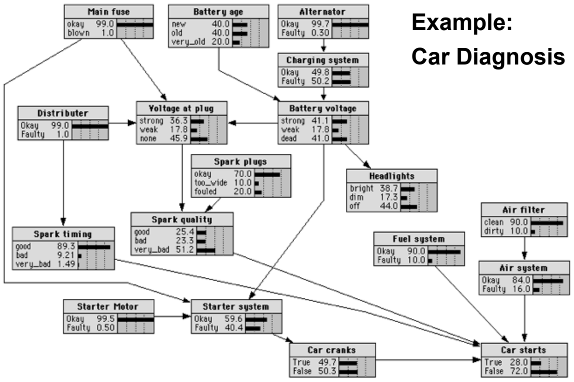


- **Mixed inferences** (combining two or more of the above).
  - Setting the effect *JohnCalls* to true and the cause *Earthquake* to false gives *P(Alarm |JohnCalls* ∧ ¬*Earthquake*) = 0.03

Chest clinic example



Example: Car Diagnosis



Other types of queries

- **Most probable explanation (MPE) or most likely hypothesis:**

  The instantiation of *all* the remaining variables U with the highest probability given the evidence

  $$\text{MPE}(U \mid e) = \text{argmax}_u\ P(u,e)$$

- **Maximum a posteriori (MAP):**

  The instantiation of *some* variables V with the highest probability given the evidence

  $$\text{MAP}(V \mid e) = \text{argmax}_v\ P(v,e)$$

  Note that the assignment to A in MAP(A|e) might be completely different from the assignment to A in MAP({A,B} | e) because of summing over non-specified variables, e.g., B.

- Other queries: probability of an arbitrary logical expression over query variables, decision policies, information value, seeking evidence, information gathering planning, etc.

Representation of Conditional Probability Tables

- Canonical distributions
- Deterministic nodes
  - No uncertainty in decision

    If x_1=a and x_2=b ⇒ x_3=c
- Noisy - OR
  - Generalization of logical/OR
  - Each cause *(parent)* has an independent chance of causing the effect
  - All possible causes are listed
    - Otherwise add "miscellaneous cause"
  - Inhibition of causality independent among causes
  - O(*k*) vs O(2^k) parameters need to specify *P(H/C_i)*
  - *P(~H|C_1, ... C_n) = product of (1-P(H|C_i)) for all C_i=T*
  - Reduce CPT significantly

Example of Noisy-OR

P(Fever=T/Cold=T) = .4
P(Fever=T/Flu=T) = .8
P(Fever=T/Malaria=T) = .9

P(~H/C_1, … C_n) = product of (1-P(H/C_i))
for all C_i=T; if all false (C_i=F) then 0

Cold	*Flu*	*Malaria*	*P(Fever)*	*P(¬Fever)*
F	F	F	0.0	1.0
F	F	T	0.9	0.1
F	T	F	0.8	0.2
F	T	T	0.98	0.02=0.2 x 0.1
T	F	F	0.4	0.6
T	F	T	0.94	0.06=0.6 x 0.1
T	T	F	0.88	0.12 = 0.6 x 0.2
T	T	T	0.988	0.012 = 0.6 x 0.2 x 0.1

Next Lecture

- Construction of Belief Network
- Inference in Belief Networks
- Belief propagation

Overall Grades

- A (86-91) 5
- A- (84-79) 9
- B+ (75-64) 13
- Below B+ 9

Review of Long Questions on Exam –A

- A (22 points) Sketch out an algorithm for bi-directional A*. As part of the sketch you should discuss why your algorithm will always find the minimal cost solution.
- In order to do this problem, I would need to have both a heuristic admissible function that worked for both directions and obviously a well defined goal and start state (for example in route finding problem the city I am starting at the and city that I am going to) and appropriate operators for going in both directions. Obviously if you were doing the route finding you could do the search in both directions using the same operators and heuristic function. Additionally the cost g between two directly connected nodes should be the same no matter what direction you are coming from. *There are two issues that must be resolved. First is how do I make a decision about which direction to next proceed. I would have two open lists one for each direction. I would choose for the node to next expand which has the smallest f value on either list. This way if I expand a node on backward search which is the smallest f and it is the initial state I have found the lowest cost solution and vice versa. I also have to understand how to handle the situation where in expanding a node one or more of its successors is on the other direction's open list. In that case you can combine the two paths and generate a new node on the open list of the node that was a complete path with appropriate cost. Like A* generating a complete solution does not mean you can immediately terminate the search, you need to wait until this solution is taken off the open list to make the decision that this is the minimal cost path. However, if the node was on the other agent's closed list then you could immediately stop.*

Review of Long Questions on Exam –B.1

- B.1 (8 points) Sketch out very briefly how this problem can be translated into an N-SAT problem in order to perform a stochastic search. You do not need to do the full translation!

- *For each node (state) in the graph there would be three literals. For example CT-red, CT-blue and CT-yellow. You would then have clauses indicating the one and only one of those literals is true. Similar to the mapping of the n-queens problem. You would then have clauses indicating the constraints among nodes. For instance there would be clauses indicating that if CT-red is true then MA-red needs to be false and RI red needs to be false; this would require multiple 2-literal clauses to express this ((not CT-red) OR (not MA-red)) AND ((not CT-red) OR (not RI-red))*

Review of Long Questions on Exam –B.2

- B.2 (8 points) How would you formulate it as a systematic constraint satisfaction search? Give representative examples of the different types of constraints.

- *I would have a variable associated with each node (e.g, CT) in the map whose domain of values include red, blue and yellow. I would then have a set of pairwise constraints (such as CT not equal to MA) for each node in the map that is directly connected with another node. I would use min-conflict heuristic search paradigm.*

Review of Long Questions on Exam –B3.3

- B.3 (8 points) If you had a larger graph, coloring problem, let us say the entire map of the US which has 50 states, which search approach (systematic or stochastic) would you use. Briefly explain your reasoning!

- *I don't think there is an obvious answer since using the mini-conflict heuristic search at least for the N-queens problems is in the same ballpark as a stochastic search. I would see first whether I could find a good and cheap way to generate a heuristic starting solution. Probably, if that was the case, I would go with the systematic search otherwise stochastic search.*