Bayesian Networks

Louis Oliphant
oliphant@cs.wisc.edu
CS540 section 2

Remember

- chain rule – any joint probability can be represented as a sum of conditional probabilities

$$P(x_1, x_2, \ldots x_n) = \sum_i P(x_i | x_{i-1}, \ldots, x_1)$$

- conditional independence – A is independent of B given C

$$P(A, B|C) = P(A|C) P(B|C)$$

Motivation for Bayesian Networks

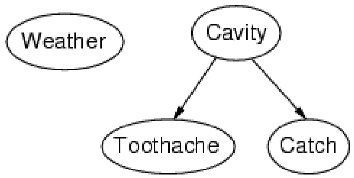
- Size of full joint probability table
  - N variables that can take on k possible values
  - size of full joint is k^N
- Would be nice to keep all information, but in a compacted form

Bayesian networks

- A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions
- Syntax:
  - a set of nodes, one per variable
  - a directed, acyclic graph (link ≈ "directly influences")
  - a conditional distribution for each node given its parents: $\mathbf{P}(X_i \mid \text{Parents}(X_i))$
- In the simplest case, conditional distribution represented as a conditional probability table (CPT) giving the distribution over X_i for each combination of parent values

Example

- Topology of network encodes conditional independence assertions:

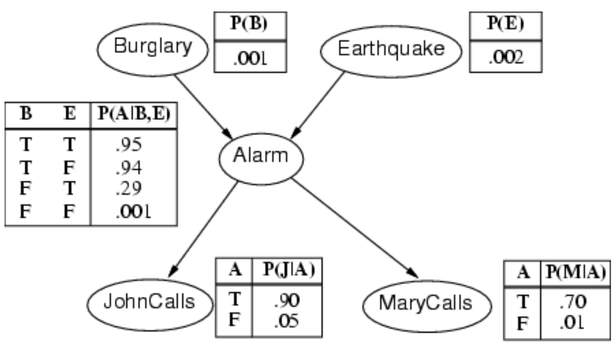


- *Weather* is independent of the other variables
- *Toothache* and *Catch* are dependent upon *Cavity* and they are conditionally independent from each other given *Cavity*

Example

- I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?
- Variables: *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*
- Network topology reflects "causal" knowledge:
  - A burglar can set the alarm off
  - An earthquake can set the alarm off
  - The alarm can cause Mary to call
  - The alarm can cause John to call

Example contd.



P(B)
.001

P(E)
.002

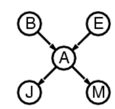
B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Compactness

- A CPT for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values

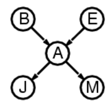


- Each row requires one number p for $X_i = true$ (the number for $X_i = false$ is just $1-p$)
- If each variable has no more than k parents, the complete network requires $O(n \cdot 2^k)$ numbers
- I.e., grows linearly with n, vs. $O(2^n)$ for the full joint distribution
- For burglary net, 1 + 1 + 4 + 2 + 2 = 10 numbers (vs. 2^5-1 = 31)

Semantics

The full joint distribution is defined as the product of the local conditional distributions:

$$\mathbf{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i \mid Parents(X_i))$$



e.g., $\mathbf{P}(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

$= \mathbf{P}(j \mid a)\, \mathbf{P}(m \mid a)\, \mathbf{P}(a \mid \neg b, \neg e)\, \mathbf{P}(\neg b)\, \mathbf{P}(\neg e)$

Two Parts to every Model

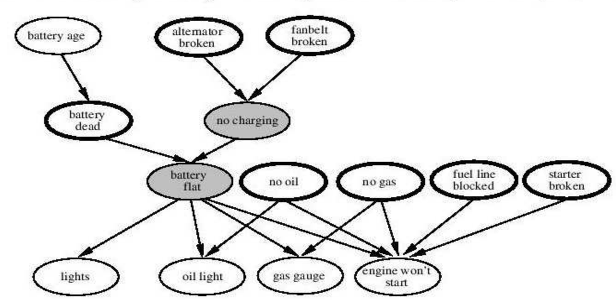
- Inference (classifying)
  - Inference by Enumeration (exact method)
  - Variable Elimination (exact method)
  - Sampling methods (approximation method) *(next time)*
- Induction (learning) *(next time)*
  - parameter learning
    - Given a Bayes Net graph, fill in the values in the CPTs using some training set of data
  - structure learning
    - Given some training set construct the Bayes Net topography

Inference in Bayes Networks

- Three types of nodes during inference
  - Query Nodes
    - These are the ones you want to know the probability distribution about
  - Evidence Nodes
    - These are the ones you know what their values are
  - Hidden Nodes
    - These are the ones you don't know anything about
- More powerful than simply having a fixed classification feature like with decision trees, neural nets, etc.
  - **<u>You can query about any node, or any set of nodes</u>**

Another Example

Initial evidence: engine won't start
Testable variables (thin ovals), diagnosis variables (thick ovals)
Hidden variables (shaded) ensure sparse structure, reduce parameters



Inference by Enumeration

```
ENUMERATIONASK(X,e,bn) returns a distribution over X
inputs: X, the query variable
        e, evidence specified as an event
        bn, a belief network specifying joint distribution P(X_1, ..., X_n)

  Q(x) ← a distribution over X
  for each value x_i of X do
       extend e with value x_i for X
       Q(x_i) ← ENUMERATEALL(VARS[bn], e)
  return NORMALIZE(Q(X))
-----------------------------------------------------------------
ENUMERATEALL(vars, e) returns a real number
  if EMPTY?(vars) then return 1.0
  else do
       Y ← FIRST(vars)
       if Y has value y in e
            then return P(y | Pa(Y)) × ENUMERATEALL(REST(vars), e)
            else return Σ_y P(y | Pa(Y)) × ENUMERATEALL(REST(vars), e_y)
                 where e_y is e extended with Y = y
```

Inference by Enumeration

Normalization Constant
(computed after the fact)

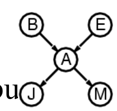
$$P(X|\mathbf{e}) = \alpha P(X,\mathbf{e}) = \alpha \sum_{\mathbf{y}} P(X,\mathbf{e},\mathbf{y})$$

Query Variables

Evidence Variables

Hidden Variables

Inference by Enumeration



Lets figure out what the probability is of a bu[rglary] given the fact that both John and Mary have called.

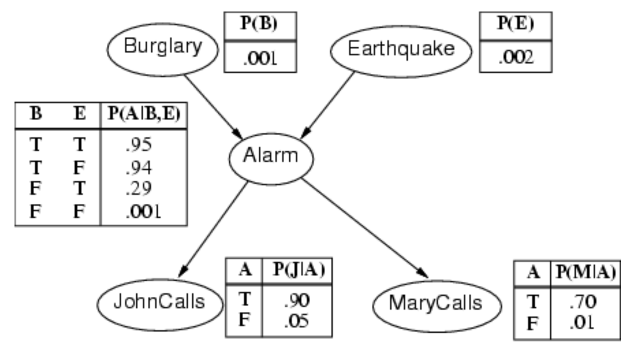
$$\mathbf{P(B|j,m) = \alpha P(B,j,m) = \alpha \sum_e \sum_a P(B,e,a,j,m)}$$

- Remember the Chain Rule. You can rewrite any joint probability as a product of conditional probabilities
- So figuring out P(b|j,m) (*half of the problem*)

$$\mathbf{P(b|j,m) = \alpha \sum_e \sum_a P(b)\ P(e)\ P(a|b,e)\ P(j|a)\ P(m|a)}$$

$$\mathbf{= \alpha P(b) \sum_e P(e) \sum_a P(a|b,e)\ P(j|a)\ P(m|a)}$$

Example



P(B)
.001

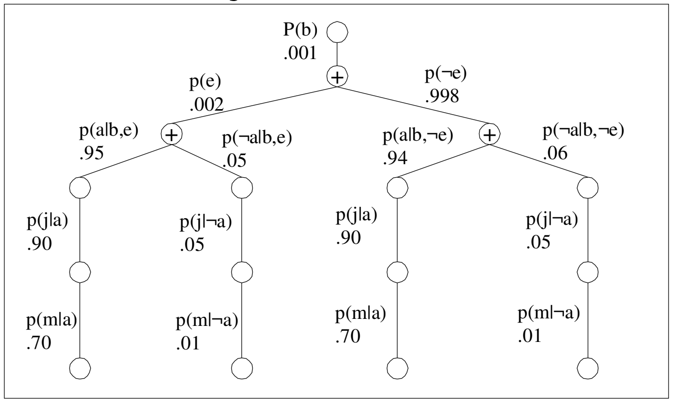
P(E)
.002

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

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Inference by Enumeration



Notice the repeated sub-structures. It would be nice to do them once and store the result. This would speed up computation.

Inference by Enumeration

- You try it:

  P(JohnCalls=true | Burglary=true)

  setup the equation for finding the probability.

Variable Elimination

Enumeration is inefficient: repeated computation
e.g., computes $P(J = true|a)P(M = true|a)$ for each value of e

Variable elimination: carry out summations right-to-left, storing intermediate results (<u>factors</u>) to avoid recomputation

$$\begin{aligned}
\mathbf{P}(B|J=true, M=true) \\
&= \alpha \underbrace{\mathbf{P}(B)}_{B} \Sigma_e \underbrace{P(e)}_{E} \Sigma_a \underbrace{\mathbf{P}(a|B,e)}_{A} \underbrace{P(J=true|a)}_{J} \underbrace{P(M=true|a)}_{M} \\
&= \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a \mathbf{P}(a|B,e) P(J=true|a) f_M(a) \\
&= \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a \mathbf{P}(a|B,e) f_J(a) f_M(a) \\
&= \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a f_A(a,b,e) f_J(a) f_M(a) \\
&= \alpha \mathbf{P}(B) \Sigma_e P(e) f_{\bar{A}JM}(b,e) \text{ (sum out } A) \\
&= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\
&= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
\end{aligned}$$

Variable Elimination

- Point-wise product between two factors

A	B	f_1(A,B)	B	C	f_2(B,C)	A	B	C	f_3(A,B,C)
T	T	.3	T	T	.2	T	T	T	.3x.2
T	F	.7	T	F	.8	T	T	F	.3x.8
F	T	.9	F	T	.6	T	F	T	.7x.6
F	F	.1	F	F	.4	T	F	F	.7x.4
						F	T	T	.9x.2
						F	T	F	.9x.8
						F	F	T	.1x.6
						F	F	F	.1x.4

- Summing out variable A

B	C	f_3(B,C)
T	T	.3x.2+.9x.2
T	F	.3x.8+.9x.8
F	T	.7x.6+.1x.6
F	F	.7x.4+.1x.4

Variable Elimination

```
function ELIMINATIONASK(X,e,bn) returns a distribution over X
   inputs: X, the query variable
           e, evidence specified as an event
           bn, a belief network specifying joint distribution P(X_1,...,X_n)
   if X ∈ e then return observed point distribution for X
   factors ← []; vars ← REVERSE(VARS[bn])
   for each var in vars do
      factors ← [MAKEFACTOR(var, e)|factors]
      if var is a hidden variable then factors ← SUMOUT(var,factors)
   return NORMALIZE(POINTWISEPRODUCT(factors))
```

Summary

- Bayesian networks provide a natural representation for (causally induced) conditional independence
- Topology + CPTs = compact representation of joint distribution
- Query, Evidence, and Hidden nodes
- Inference by Enumeration
- Variable Elimination