CSCI 5582
Artificial Intelligence

Lecture 15

Jim Martin

Today 10/19

- Review
- Belief Net Computing
- Sequential Belief Nets

Review

- Normalization
- Belief Net Semantics

Normalization

- What do I know about

P(~A| something) and P(A|same something)

They sum to 1

Normalization

- What if I have this…
  P(A, Y)/P(Y) and P(~A, Y)/P(Y)

  And I can compute the numerators but not the demoninator?

  Ignore it and compute what you have, then normalize

  P(A|Y) = P(A,Y)/(P(A,Y)+P(~A,Y))

  P(~A|Y) = P(~A,Y)/(P(A,Y)+P(~A,Y))

Normalization

- Alpha * <0.12, 0.08> = <0.6, 0.4>

Bayesian Belief Nets

- A compact notation for representing conditional independence assumptions and hence a compact way of representing a joint distribution.
- Syntax:
  - A directed acyclic graph, one node per variable
  - Each node augmented with local conditional probability tables

Bayesian Belief Nets

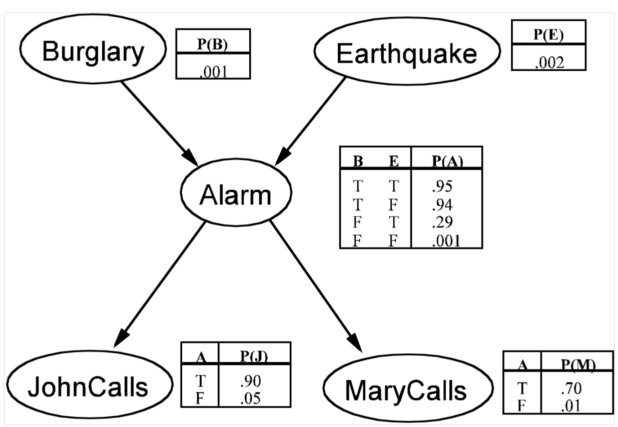
- Nodes with no incoming arcs (root nodes) simply have priors associated with them
- Nodes with incoming arcs have tables enumerating the
  - P(Node|Conjunction of Parents)
  - Where parent means the node at the other end of the incoming arc

Bayesian Belief Nets: Semantics

- The full joint distribution for the N variables in a Belief Net can be recovered from the information in the tables.

$$P(X1,...XN) = \prod_{i=1}^{N} P(Xi \mid Parents(Xi))$$

Alarm Example



Alarm Example

- P(J^M^A^~B^~E)=

  P(J|A)*P(M|A)*P(A|~B^~E)*P(~B)*P(~E)
  0.9 * 0.7 * .001 * .999 * .998

- In other words, the probability of atomic events can be read right off the network as the product of the probability of the entries for each variable

Events

- P(M ^J^E^B^A)+
  P(M^J^E^B^~A)+
  P(M^J^E^~B^A)+
  …

Chain Rule Basis

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

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

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

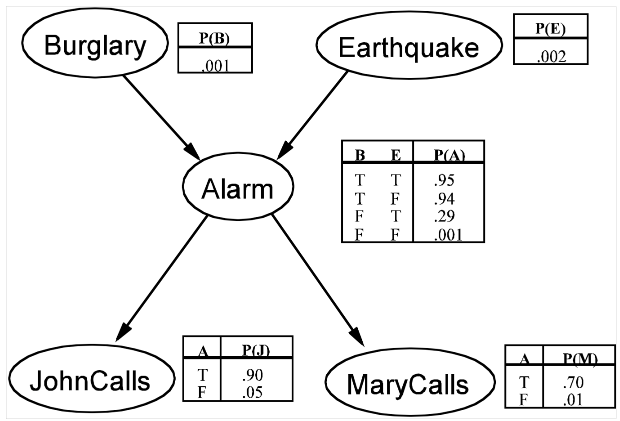
P(A|B,E)P(B,E)

P(B|E)P(E)

Chain Rule Basis

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

Alarm Example



Details

- Where do the graphs come from?
  - Initially, the intuitions of domain experts
- Where do the numbers come from?
  - Hopefully, from hard data
  - Sometimes from experts intuitions
- How can we compute things efficiently?
  - Exactly by not redoing things unnecessarily
  - By approximating things

Computing with BBNs

- Normal scenario
  - You have a belief net consisting of a bunch of variables
    - Some of which you know to be true (evidence)
    - Some of which you're asking about (query)
    - Some you haven't specified (hidden)

Example

- Probability that there's a burglary given that John and Mary are calling
- P(B|J,M)
  - B is the query variable
  - J and M are evidence variables
  - A and E are hidden variables

Example

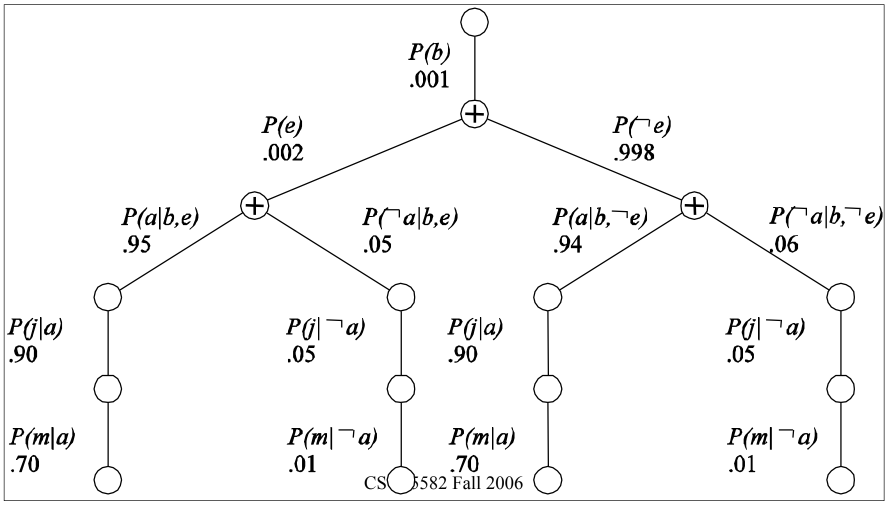
- Probability that there's a burglary given that John and Mary are calling
- P(B|J,M) = alpha P(B,J,M)
  = alpha *
  P(B,J,M,A,E) +
  P(B,J,M,~A,E)+
  P(B,J,M,A,~E)+
  P(B,J,M, ~A,~E)

From the Network

$$\alpha \sum_e \sum_a P(B)P(E)P(A \mid B,E)P(J \mid A)P(M \mid A)$$

$$\alpha P(B)\sum_e P(E)\sum_a P(A \mid B,E)P(J \mid A)P(M \mid A)$$

Expression Tree



Speedups

- Don't recompute things.
  - Dynamic programming
- Don't compute somethings at all
  - Ignore variables that can't effect the outcome.

Example

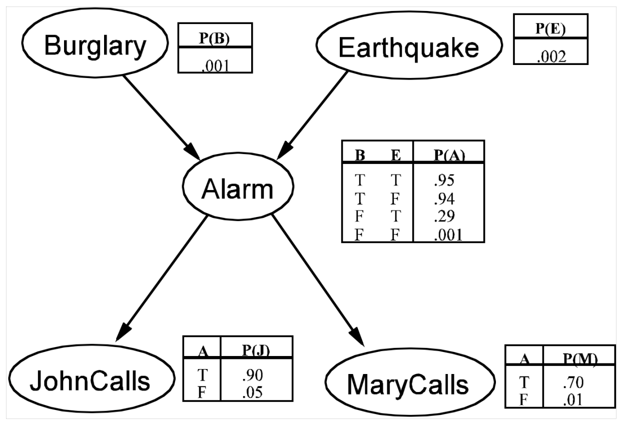
- John calls given burglary
- P(J|B)

$$\alpha P(B)\sum_e P(E)\sum_a P(A \mid B,E)P(J \mid a)\sum_m P(M \mid A)$$

Variable Elimination

- Every variable that is not an ancestor of a query variable or an evidence variable is irrelevant to the query
  - Operationally…
    - You can eliminate leaf node that isn't a query or evidence variable
    - That may produce new leaves. Keep going.

Alarm Example



Break

- Questions?

Chain Rule Basis

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

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

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

P(A|B,E)P(B,E)

P(B|E)P(E)

Chain Rule

P(E1,E2,E3,E4,E5)

P(E5|E1,E2,E3,E4)P(E1,E2,E3,E4)

P(E4|E1,E2,E3)P(E1,E2,E3)

P(E3|E1,E2)P(E1,E2)

P(E2|E1)P(E1)

Chain Rule

Rewriting that's just

P(E1)P(E2|E1)P(E3|E1,E2)P(E4|E1,E2,E3)P(E5|E1,E2,E3,E4)

The probability of a sequence of events is just the product of the conditional probability of each event given it's predecessors (parents/causes in belief net terms).

Markov Assumption

- This is just a sequence based independence assumption just like with belief nets.
  - Not all the parents matter
    - Remember P(toothache|catch, cavity)= P(toothache|cavity)
    - Now P(Event_N|Event1 to Event_N-1)= P(Event_N|Event_N-1+K to Event_N-1)

First Order Markov Assumption

P(E1)P(E2|E1)P(E3|E1,E2)P(E4|E1,E2,E3)P(E5|E1,E2,E3,E4)

P(E1)P(E2|E1)P(E3|E2)P(E4|E3)P(E5|E4)

Markov Models

- As with all our models, let's assume some fixed inventory of possible events that can occur in time
- Let's assume for now that any given point in time, all events are possible, although not equally likely

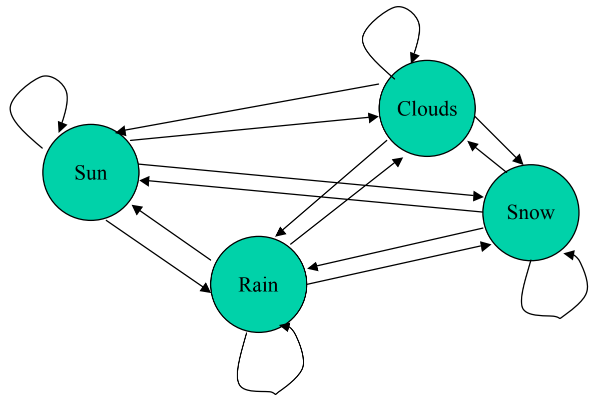
Markov Models

- You can view simple Markov assumptions as arising from underlying probabilistic state machines.
- In the simplest case (first order), events correspond to states and the probabilities are governed by probabilities on the transitions in the machine.

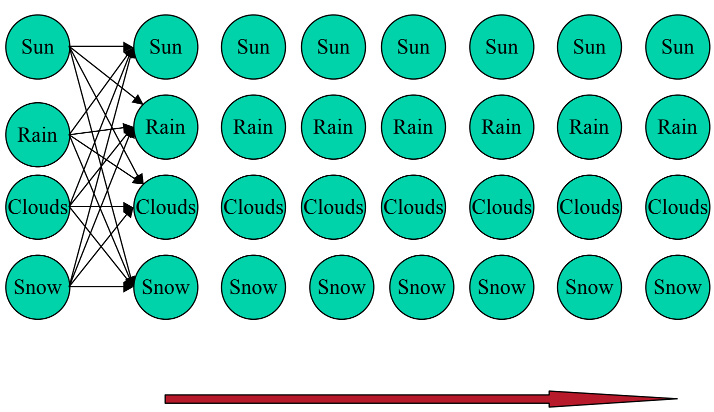
Weather

- Let's say we're tracking the weather and there are 4 possible events (each day, only one per day)
  - Sun, clouds, rain, snow

Example



Belief Net Version



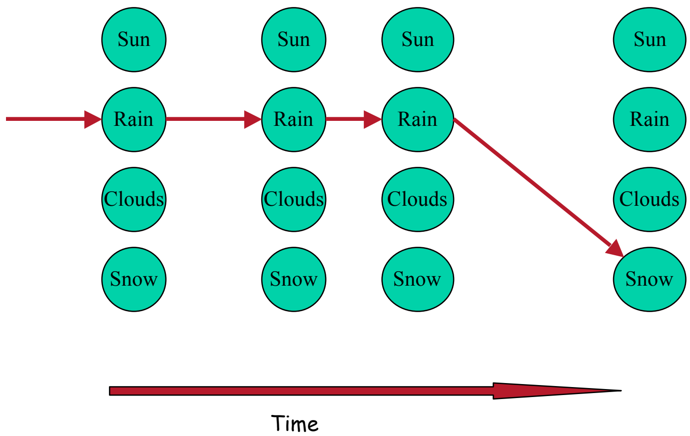
Example

- In this case we need a 4x4 matrix of transition probabilities.
  - For example P(Rain|Cloudy) or P(Sunny|Sunny) etc
- And we need a set of initial probabilities P(Rain). That's just an array of 4 numbers.

Example

- So to get the probability of a sequence like
  - Rain rain rain snow
  - You just march through the state machine
  - P(Rain)P(rain|rain)P(rain|rain)P(snow|rain)

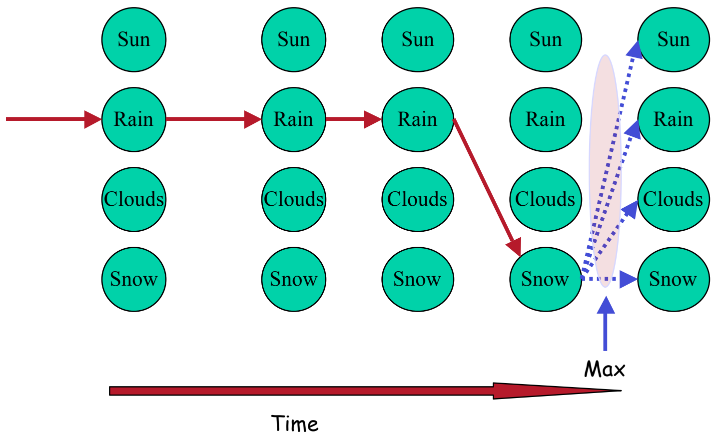
Belief Net Version



Example

- Say that I tell you that
  - Rain rain rain snow has happened
  - How would you answer
    - What's the most likely thing to happen next?

Belief Net Version



Weird Example

- What if you couldn't actually see the weather?
  - You're a security guard who lives and works in a secure facility underground.
  - You watch people coming and going with various things (snow boots, umbrellas, ice cream cones)
  - Can you figure out the weather?

Hidden Markov Models

- Add an output to the states. I.e. when a state is entered it outputs a symbol.
- You can view the outputs, but not the states directly.
  - States can output different symbols at different times
  - Same symbol can come from many states.

Hidden Markov Models

- The point
  - The observable sequence of symbols does not uniquely determine a sequence of states.
- Can we nevertheless reason about the underlying model, given the observations?

Hidden Markov Model Assumptions

- Now we're going to make two independence assumptions
  - The state we're in depends probabilistically only on the state we were last in (first order Markov assumption)
  - The symbol we're seeing only depends probabilistically on the state we're in

Hidden Markov Models

- Now the model needs
  - The initial state priors
    - $P(State_i)$
  - The transition probabilities (as before)
    - $P(State_j|State_k)$
  - The output probabilities
    - $P(Observation_i|State_k)$

HMMs

- The joint probability of a state sequence and an observation sequence is…

$$P(X_0, X_1, \ldots X_t, E_1, \ldots E_t) = P(X_0)\prod_{i=1}^{t} P(X_i \mid X_{i-1})P(E_i \mid X_i)$$

Noisy Channel Applications

- The hidden model represents an original signal (sequence of words, letters, etc)
- This signal is corrupted probabilistically. Use an HMM to recover the original signal
- Speech, OCR, language translation, spelling correction,…

Three Problems

- The probability of an observation sequence given a model
  - Forward algorithm
  - Prediction falls out from this
- The most likely path through a model given an observed sequence
  - Viterbi algorithm
  - Sometimes called decoding
- Finding the most likely model (parameters) given an observed sequence
  - EM Algorithm